HP Prime グラフ電卓

© 2015, 2016 HP Development Company, L.P.

本書の内容は、将来予告なしに変更されることがあります。HP 製品およびサービスに関する保証は、当該製品およびサービスに付属の保証規定に明示的に記載されているものに限られます。本書のいかなる内容も、当該保証に新たに保証を追加するものではありません。本書に記載されている製品情報は、日本国内で販売されていないものも含まれている場合があります。本情報の内容につきましては万全を期しておりますが、本情報の技術的あるいは校正上の誤り、省略に対して責任を負いかねますのでご了承ください。

当社は、本書の誤りや本書の提供、内容の実施、また本書の利用によって偶発的あるいは必然的に生じる損害については責任を負わないものとします。

機密情報コンピューター ソフトウェア。このソフトウェアの所有、使用、複製には HP からの有効な使用許諾が必要です。米国政府の連邦調達規則である FAR 12.211 および 12.212 の規定に従い、市販コンピューター ソフトウェア、コンピューター ソフトウェア文書および市販品の技術データは、ベンダーが提供する標準使用許諾規定に基づき米国政府に使用許諾が付与されます。

本ソフトウェアには FreeType プロジェクトの著作物が含まれます。copyright 2013 The FreeType Project (www.freetype.org). All rights reserved. HP は FreeType を FreeType License に基づいて配布しています。HP は google-droid-fonts を Apache Software License v2.0 に基づいて配布しています。HP は HIDAPI を BSD ライセンスのみに基づいて配布しています。HP は Qt を LGPLv2.1 ライセンスに基づいて配布しています。HP は Qt ソースの完全なコピーを提供しています。HP は QuaZIP を LGPLv2 ライセンスおよび zlib/libpng ライセンスに基づいて配布しています。HP は QuaZIP ソースの完全なコピーを提供しています。

『製品規制、および環境情報』は、この製品に付属している CD で提供されています。

改訂第 1 版：2016 年 9 月

初版：2015 年 7 月

製品番号：813269-292

目次

1 前書き 1
マニュアルの表記法 1

2 お使いになる前に 2
準備 2
オン/オフおよびキャンセル操作 3
電源オン 3
キャンセル 3
電源オフ 3
ホーム表示 3
CAS 表示 4
保護カバー 4
ディスプレイ 4
輝度の調整 4
ディスプレイのクリア 4
ディスプレイの各部 4
クイック設定メニュー 6
ナビゲーション 6
タッチ ジェスチャ 7
キーボード 7
コンテキスト依存メニュー 8
入力キーおよび編集キー 9
シフト キー 10
テキストの追加 11
計算キー 12
数学テンプレート 12
数学ショートカット 13
分数 14
60 進数 15
EEX キー（10 のべき乗） 15
メニュー 16
メニュー項目の選択 17
ショートカット 17
メニューの閉じ方 17
ツールボックス メニュー 17

入力フォーム 17
入力フォームのフィールドのリセット 18
システム全体の設定 18
ホーム設定 19
ページ 1 19
ページ 2 20
ページ 3 20
ページ 4 21
ホーム設定の指定 21
数学計算 22
作業場所 23
入力形式の選択 23
式の入力 23
例 24
かっこ 24
代数の優先順位 25
負数 25
明示的な乗算と暗黙的な乗算 25
大きい結果 26
式および結果の再利用 26
クリップボードの使用 26
最後の結果の再利用 26
CAS の式または結果の再利用 28
変数への値の保存 28
複素数 29
コピーおよび貼り付け 30
データの共有 32
基本手順 32
メモリ管理者の使用 33
バックアップ カタログ 33
オンライン ヘルプ 34

3 逆ポーランド記法（RPN） 37

RPN モードの履歴 38
結果の再利用 39
計算例 39
スタックの操作 41
PICK（ピック） 41
ROLL（ロール） 42

Swap（スワップ） 42
Stack（スタック） 42
DROPN（以下をドロップ） 42
DUPN（以下を複製） 42
Echo（エコー） 42
→LIST（→リスト） 42
項目の表示 43
特定の項目の削除 43
全項目の削除 44

4 数式処理システム（CAS） 45
CAS 表示 45
CAS 計算 46
例 1 46
例 2 47
設定 48
ページ 1 48
ページ 2 49
メニュー項目の形式の設定 49
ホーム表示の式または結果の使用 50
CAS でのホーム変数の使用 50

5 試験モード 51
基本モードの使用 51
初期設定構成の変更 52
新しい構成の作成 54
試験モードの有効化 55
試験モードの解除 56
構成の変更 56
構成の変更 57
初期設定の構成に戻す 57
構成の削除 57

6 HP アプリケーションの基本 58
アプリケーション ライブラリ 59
アプリケーションを開く 59
アプリケーションのリセット 60
アプリケーションの並べ替え 60
アプリケーションの削除 60

その他のオプション 61
アプリケーション表示 61
[記号表示] 61
[記号のセットアップ]表示 62
[プロット表示] 62
[プロットのセットアップ]表示 63
[数値表示] 64
[数値のセットアップ]表示 64
簡単な例 65
アプリケーションを開く 65
[記号表示] 65
[記号のセットアップ]表示 66
[プロット表示] 66
[プロットのセットアップ]表示 67
[数値表示] 68
[数値のセットアップ]表示 68
[記号表示]の共通操作 69
定義の追加 69
定義の変更 69
定義の構成要素 69
従属定義の評価 71
操作対象の定義の選択および選択解除 71
プロットの色の選択 71
定義の削除 72
[記号表示] : メニュー ボタンの概要 72
[記号のセットアップ]表示の共通操作 73
システム全体の設定の変更 74
初期設定の復元 74
[プロット表示]の共通操作 74
ズーム 75
ズーム倍率 75
ズーム オプション 75
ズーム ジェスチャ 76
ズーム キー 76
[ズーム]メニュー 76
ボックス ズーム 77
[表示]メニュー 77
分割画面表示を使用したズームのテスト 78
ズーム例 79

拡大 79
縮小 80
X 拡大 80
X 縮小 81
Y 拡大 81
Y 縮小 82
2 乗 82
自動スケール 83
小数 83
整数 84
三角法 84
トレース 85
プロットの選択 85
関数の評価 86
トレース機能のオン/オフ 86
[プロット表示] : メニュー ボタンの概要 87
[プロット表示]でのコピーおよび貼り付けの操作 87
[プロットのセットアップ]表示の共通操作 87
[プロット表示]の設定 87
ページ 1 88
ページ 2 89
ページ 3 89
グラフ化の方式 90
初期設定の復元 92
[数値表示]の共通操作 92
ズーム 92
ズーム オプション 93
ズーム ジェスチャ 94
ズーム キー 94
[ズーム]メニュー 94
評価 95
カスタム表 95
データの削除 96
[数値表示]でのコピーおよび貼り付け 96
セルのコピーおよび貼り付け 96
行のコピーおよび貼り付け 97
複数のセルのコピーおよび貼り付け 97
[数値表示] : メニュー ボタンの概要 98
[その他]メニュー 98

[数値のセットアップ]表示の共通操作 99
初期設定の復元 99
[プロット表示]と[数値表示]の連携 99
アプリケーションへの注記の追加 100
アプリケーションの作成 100
例 101
アプリケーション関数およびアプリケーション変数 102
関数 102
変数 103
変数の修飾 104

7 関数アプリケーション 105
関数アプリケーションでの作業の開始 105
関数アプリケーションを開く 105
式の定義 106
プロットのセットアップ 106
関数のプロット 107
グラフのトレース 107
スケールの変更 109
[数値表示]の表示 109
[数値表示]のセットアップ 110
[数値表示]の操作 111
表内の移動 111
値への直接移動 112
ズーム オプションへのアクセス 112
その他のオプション 113
関数の分析 113
[プロット表示]メニューの表示 113
関数のスケッチ 113
関数グラフの変更 114
2 次方程式の根を求める 116
2 つの関数の交点を求める 117
2 次方程式の傾きを求める 118
2 つの関数間の符号付き面積を求める 119
2 次方程式の極値を求める 120
関数への接線の追加 121
関数の変数 121
関数変数へのアクセス 122
[関数]演算のまとめ 123

導関数または積分による関数定義 124
導関数による関数定義 124
積分による関数定義 126

8 高度なグラフ作成アプリケーション 129
高度なグラフ作成アプリケーションでの作業の開始 131
高度なグラフ作成アプリケーションを開く 131
開放文の定義 132
プロットのセットアップ 132
選択した定義のプロット 133
グラフの操作 133
[プロット表示]でのトレース 135
[数値表示] 136
[数値表示]の表示 137
[数値表示]の操作 137
[数値のセットアップ]表示 138
[数値表示]のトレース 138
端 139
PoI 140
プロット ギャラリー 141
プロット ギャラリーでのプロットの操作 141

9 幾何 142
幾何アプリケーションでの作業の開始 142
準備 142
アプリケーションを開く/グラフのプロット 142
制限される点の追加 143
接線の追加 144
微分係数の点の作成 145
計算の追加 147
[プロット表示]での計算 149
微分係数のトレース 149
[プロット表示]の詳細 150
オブジェクトの選択 151
名前の非表示 152
オブジェクトの移動 152
オブジェクトの色の設定 152
オブジェクトの塗りつぶし 152
特定のオブジェクトの消去 153

すべてのオブジェクトの消去 154
[プロット表示]のジェスチャ 154
ズーム 154
[プロット表示] : ボタンおよびキー 154
オプション メニュー 155
勾配場コマンドの使用 156
[プロットのセットアップ]表示 157
[記号表示]の詳細 157
オブジェクトの作成 158
エントリの並べ替え 159
オブジェクトの非表示 159
オブジェクトの削除 159
[記号のセットアップ]表示 160
[数値表示]の詳細 160
全オブジェクトのリスト 161
[プロット表示]での計算の表示 162
計算の編集 162
計算の削除 163
[プロット表示] : [コマンド]メニュー 163
[点] 163
[点] 163
[点オン] 164
[中点] 164
[中央] 164
[交点] 164
[交点] 164
[ランダム点] 164
[線] 165
[線分] 165
[半直線] 165
[直線] 165
[平行] 165
[垂直] 165
[正接] 166
[中線] 166
[垂線] 166
[∡二等分線] 166
[多角形] 166
[三角形] 166

[Δ三角形] 166
[⊿三角形] 167
[四辺形] 167
[平行四辺形] 167
[菱形] 167
[長方形] 167
[多角形] 168
[正多角形] 168
[正方形] 168
[曲線] 168
[円] 168
[外接円] 168
[傍接円] 169
[内接円] 169
[楕円] 170
[双曲線] 170
[放物線] 170
[円錐] 170
[軌跡] 170
[プロット] 171
[関数] 172
[パラメトリック] 172
[極座標] 172
[数列] 173
[陰] 173
[勾配場] 173
[ODE] 173
[リスト] 173
[スライダー] 174
[変換] 174
[平行移動] 174
[鏡映] 175
[回転] 176
[拡大縮小] 176
[相似] 177
[投影] 177
[反転] 178
[逆数変換] 178
[デカルト] 179

[横座標] 179
[縦座標] 179
点→複素数 179
[座標] 179
[方程式] 179
[パラメトリック] 180
[極座標] 180
[測定] 180
[距離] 180
[半径] 180
[周長] 180
[傾き] 180
[面積] 180
[角度] 180
[弧長] 180
[テスト] 181
[共線] 181
[円周上] 181
[オブジェクト上] 181
[平行] 181
[垂直] 181
[二等辺] 181
[等辺形] 182
[平行四辺形] 182
[共役] 182
幾何関数およびコマンド 182
[記号表示] : [コマンド]メニュー 182
[点] 183
[点] 183
[点オン] 183
[中点] 183
[中央] 183
[交点] 183
[交点] 184
[線] 184
[線分] 184
[半直線] 184
[直線] 184
[平行] 184

[垂直] 185
[正接] 185
[中線] 185
[垂線]） 185
[二等分線] 185
[多角形] 186
[三角形] 186
[Δ三角形] 186
[◿三角形] 186
[四辺形] 186
[平行四辺形] 186
[菱形] 187
[長方形] 187
[多角形] 187
[正多角形] 187
[正方形] 188
[曲線] 188
[円] 188
[外接円] 188
[傍接円] 188
[内接円] 188
[楕円] 189
[双曲線] 189
[放物線] 189
[円錐] 189
[軌跡] 190
[プロット] 190
[関数] 190
[パラメトリック] 190
[極座標] 190
[数列] 190
[陰] 191
[勾配場] 191
[ODE] 191
[リスト] 191
[スライダー] 191
[変換] 192
[平行移動] 192
[鏡映] 192

[回転] ... 192
[拡大縮小] ... 192
[相似] ... 192
[投影] ... 193
[反転] ... 193
[逆数変換] ... 193
[数値表示] : [コマンド]メニュー ... 193
[デカルト] ... 193
[横座標] ... 193
[縦座標] ... 193
[座標] ... 193
[方程式] ... 194
[パラメトリック] ... 194
[極座標] ... 194
[測定] ... 194
[距離] ... 194
[半径] ... 194
[周長] ... 194
[傾き] ... 195
[面積] ... 195
[角度] ... 195
[弧長] ... 195
[テスト] ... 196
[共線] ... 196
[円周上] ... 196
[オブジェクト上] ... 196
[平行] ... 196
[垂直] ... 196
[二等辺] ... 196
[等辺形] ... 197
[平行四辺形] ... 197
[共役] ... 197
その他の幾何アプリケーション関数 ... 197
affix ... 197
barycenter ... 197
convexhull ... 198
distance2 ... 198
division_point ... 198
equilateral_triangle ... 198

exbisector 199
extract_measure 199
harmonic_conjugate 199
harmonic_division 199
isobarycenter 199
is_harmonic 200
is_harmonic_circle_bundle 200
is_harmonic_line_bundle 200
is_orthogonal 200
is_rectangle 200
is_rhombus 200
is_square 201
LineHorz 201
LineVert 201
open_polygon 201
orthocenter 201
perpendicular_bisector 201
point2d 202
polar 202
pole 202
powerpc 202
radical_axis 202
vector 203
vertices 203
vertices_abca 203

10 スプレッドシート 204

スプレッドシート アプリケーションでの作業の開始 204
基本操作 209
ナビゲーション、選択、およびジェスチャ 209
セル参照 209
セルの名前設定 210
方法 1 210
方法 2 210
計算での名前の使用 210
コンテンツの入力 211
直接入力 211
データのインポート 212
外部関数 212
コピーおよび貼り付け 213

CHOOSE コマンドの使用 214
外部参照 214
変数の参照 215
スプレッドシート計算での CAS の使用 216
ボタンおよびキー 216
形式オプション 217
形式パラメーター 218
スプレッドシート関数 219

11 1 変数統計アプリケーション 220

1 変数統計アプリケーションでの作業の開始 220
[記号表示]：メニュー項目 223
統計データの入力および編集 226
[数値表示]：メニュー項目 227
[その他]メニュー 227
データ セットの編集 228
データの削除 228
データの挿入 228
データの生成 229
データ値の並べ替え 229
計算される統計 229
プロット 230
統計データのプロット 230
プロット タイプ 231
ヒストグラム 231
箱ひげ図 231
正規確率プロット 232
線プロット 232
棒グラフ 233
パレート図 233
コントロール図 234
ドット図 234
茎葉図 235
円グラフ 235
プロットのセットアップ 236
グラフの操作 236
[プロット表示]：メニュー項目 236

12 2 変数統計アプリケーション 238
2 変数統計アプリケーションでの作業の開始 238
2 変数統計アプリケーションを開く 238
データの入力 239
データ列およびフィットの選択 240
統計の操作 241
プロットのセットアップ 242
グラフのプロット 243
方程式の表示 243
値の予測 244
統計データの入力および編集 245
[数値表示]：メニュー項目 246
[その他]メニュー 246
回帰モデルの定義 247
フィットの選択 247
フィット タイプ 247
独自のフィットの定義 248
計算される統計 248
統計データのプロット 249
散布図のトレース 250
曲線のトレース 250
トレースの順序 251
[プロット表示]：メニュー項目 251
[関数]メニュー 251
スケッチ 252
[プロットのセットアップ]表示 252
値の予測 252
[プロット表示] 252
ホーム表示 252
プロットのトラブルシューティング 253

13 推論アプリケーション 254
サンプル データ 254
推論アプリケーションでの作業の開始 254
推論アプリケーションを開く 254
[記号表示]のオプション 255
推論方式の選択 256
データの入力 258
検定結果の表示 258

検定結果のプロット 259
統計のインポート 260
1 変数統計アプリケーションを開く 260
不要なデータのクリア 260
データの入力 260
統計の計算 261
推論アプリケーションを開く 261
推論の方式およびタイプの選択 262
データのインポート 263
数値での結果の表示 263
グラフでの結果の表示 264
仮説検証 264
1 標本 Z 検定 265
メニュー名 265
入力 265
結果 265
2 標本 Z 検定 265
メニュー名 265
入力 266
結果 266
1 比率の Z 検定 266
メニュー名 266
入力 267
結果 267
2 比率の Z 検定 267
メニュー名 267
入力 268
結果 268
1 標本 T 検定 268
メニュー名 268
入力 268
結果 269
2 標本 T 検定 269
メニュー名 269
入力 269
結果 270
信頼区間 270
1 標本 Z 区間 270
メニュー名 270

入力 270
結果 271
2 標本 Z 区間 271
メニュー名 271
入力 271
結果 271
1 比率の Z 区間 272
メニュー名 272
入力 272
結果 272
2 比率の Z 区間 272
メニュー名 272
入力 273
結果 273
1 標本 T 区間 273
メニュー名 273
入力 273
結果 273
2 標本 T 区間 274
メニュー名 274
入力 274
結果 274
カイ 2 乗検定 275
適合度検定 275
メニュー名 275
入力 275
結果 275
メニューキー 275
2 元表に基づく検定 276
メニュー名 276
入力 276
結果 276
メニューキー 276
回帰の推論 276
線形 t 検定 277
メニュー名 277
入力 277
結果 277
メニューキー 277

傾きについての信頼区間 278
メニュー名 278
入力 278
結果 278
メニューキー 278
切片についての信頼区間 279
メニュー名 279
入力 279
結果 279
メニューキー 279
平均応答についての信頼区間 280
メニュー名 280
入力 280
結果 280
メニューキー 280
予測区間 281
メニュー名 281
入力 281
結果 281
メニュー キー 281
ANOVA 282
メニュー名 282
入力 282
結果 282
メニュー キー 282

14 Solve（解く）アプリケーション 283
解くアプリケーションでの作業の開始 283
1 つの方程式 283
Solve（解く）アプリケーションを開く 283
アプリケーションのクリアおよび方程式の定義 284
既知の変数の入力 285
未知の変数を求める 285
方程式のプロット 286
複数の方程式 287
Solve（解く）アプリケーションを開く 287
方程式の定義 287
シード値の入力 288
未知の変数を求める 289

制限事項 289
解の情報 289

15 Linear Solver（線形ソルバー）アプリケーション 291
Linear Solver（線形ソルバー）アプリケーションでの作業の開始 291
Linear Solver（線形ソルバー）アプリケーションを開く 291
方程式を定義して解く 292
2x2 系を解く 293
メニュー項目 293

16 Parametric（パラメトリック）アプリケーション 294
Parametric（パラメトリック）アプリケーションでの作業の開始 294
Parametric（パラメトリック）アプリケーションを開く 294
関数の定義 295
角度の単位の設定 295
プロットのセットアップ 296
関数のプロット 297
グラフの操作 297
[数値表示]の表示 298

17 Polar（極座標）アプリケーション 299
Polar（極座標）アプリケーションでの作業の開始 299
Polar（極座標）アプリケーションを開く 299
関数の定義 299
角度の単位の設定 300
プロットのセットアップ 301
式のプロット 301
グラフの操作 302
[数値表示]の表示 303

18 数列アプリケーション 304
数列アプリケーションでの作業の開始 305
数列アプリケーションを開く 305
式の定義 305
プロットのセットアップ 306
数列のプロット 307
グラフの操作 308
[数値表示]の表示 308
値の表の操作 309

値の表のセットアップ 310
別の例：明示的に定義された数列 310
式の定義 310
プロットのセットアップ 311
数列のプロット 311
値の表の操作 312

19 Finance（金融）アプリケーション 313
Finance（金融）アプリケーションでの作業の開始 313
キャッシュ フロー ダイアグラム 315
貨幣の時間価値（TVM） 316
別の例：TVM 計算 317
償却 318
償却の計算 318
住宅ローンの例での償却 318
償却グラフ 320

20 Triangle Solver（三角ソルバー）アプリケーション 321
Triangle Solver（三角ソルバー）アプリケーションでの作業の開始 321
Triangle Solver（三角ソルバー）アプリケーションを開く 321
角度の単位の設定 321
既知の値の指定 322
未知の値を求める 322
三角形のタイプの選択 323
特殊な場合 323
解が 1 つに決まらない場合 323
与えられたデータで解がない 324
データが不十分 324

21 エクスプローラー アプリケーション 326
Linear Explorer（線形エクスプローラー）アプリケーション 326
アプリケーションを開く 326
グラフ モード 327
方程式モード 328
テスト モード 328
2 次エクスプローラーアプリケーション 329
アプリケーションを開く 329
グラフ モード 330
方程式モード 330

テスト モード 331
三角法エクスプローラーアプリケーション 331
アプリケーションを開く 332
グラフ モード 332
方程式モード 333
テスト モード 333

22 関数およびコマンド 335

キーボード関数 337

..... 337

..... 337

(ex) 338

..... 338

(10x) 338

..... 338

(ASIN) 338

(ACOS) 339

(ATAN) 339

..... 339

..... 339

..... 339

..... 340

..... 340

..... 340

(|x|) 340

..... 340

Shift a b/c 341
EEX 341
Shift 9 341
Shift 6 342
Shift 2 342
Shift 3 342
[数学]メニュー 342
[数] 342
[天井] 342
[床] 342
[IP] 342
[FP] 343
[丸め] 343
[切り捨て] 343
[仮数部] 343
[指数] 343
[算術計算] 344
[最大] 344
[最小] 344
[係数] 344
[根を求める] 344
[パーセント] 344
[算術計算] : [複素数] 345
[偏角] 345
[共役] 345
[実部] 345
[虚部] 345
[単位ベクトル] 345
[算術計算] : [指数] 346
[ALOG] 346
[EXPM1] 346
[LNP1] 346
[三角関数] 346
[CSC] 346
[ACSC] 346

[SEC] 346
[ASEC] 346
[COT] 346
[ACOT] 346
[双曲線] 347
[SINH] 347
[ASINH] 347
[COSH] 347
[ACOSH] 347
[TANH] 347
[ATANH] 347
[確率] 347
[階乗] 347
[組み合わせ] 347
[順列] 348
[確率] : [乱数] 348
[番号] 348
[整数] 348
[正規] 348
[シード] 348
[確率] : [密度] 349
[正規] 349
[T] 349
[χ2] 349
[F] 349
[二項] 349
[幾何] 350
[ポワソン] 350
[確率] : [累積] 350
[正規] 350
[T] 350
[X2] 350
[F] 351
[二項] 351
[幾何] 351
[ポワソン] 351
[確率] : [逆] 351
[正規] 351
[T] 352
[χ2] 352

[F] 352
[二項] 352
[幾何] 352
[ポワソン] 353
[リスト] 353
[行列] 353
[特殊] 353
[ベータ] 353
[ガンマ] 353
[Psi] 353
[Zeta] 353
[erf] 353
[erfc] 353
[Ei] 354
[Si] 354
[Ci] 354
[CAS]メニュー 354
[代数] 354
[簡略化] 354
[整理] 355
[展開] 355
[因数] 355
[置換] 355
[部分分数] 355
[代数] : [抽出] 355
[分子] 355
[分母] 356
[左辺] 356
[右辺] 356
[微積分] 356
[微分] 356
[積分] 357
[極限] 357
[数列] 357
[総和] 357
[微積分] : [微分] 358
[回転] 358
[発散] 358
[勾配] 358
[ヘッセ] 358

[微積分] : [積分] 358
[部分積分 u] 358
[部分積分 v] 359
[F(b) – F(a)] 359
[微積分] : [極限] 359
[リーマン和] 359
[テイラー] 359
[商のテイラー] 359
[微積分] : [変換] 360
[ラプラス] 360
[ラプラス逆] 360
[FFT] 360
[逆 FFT] 360
[解く] 360
[解く] 360
[ゼロ] 361
[複素数を解く] 361
[複素零点] 361
[数値的に解く] 361
[微分方程式] 362
[常微分方程式を解く] 362
[線形系] 362
[書き換え] 362
[lncollect] 362
[powexpand] 363
[テイラー展開] 363
[書き換え] : [指数関数と自然対数] 363
[ey*lnx→ xy] 363
[xy→ey*lnx] 363
[exp2trig] 363
[expexpand] 363
[書き換え] : [正弦] 364
[asinx→acosx] 364
[asinx→atanx] 364
[sinx→cosx*tanx] 364
[書き換え] : [余弦] 364
[acosx→asinx] 364
[acosx→atanx] 364
[cosx → sinx/tanx] 365
[書き換え] : [正接] 365

[atanx→asinx] 365
[atanx→acosx] 365
tanx→sinx/cosx 365
[halftan] 365
[書き換え]：[三角法] 365
[trigx→sinx] 365
[trigx→cosx] 366
[trigx→tanx] 366
[atrig2ln] 366
[tlin] 366
[tcollect] 366
[trigexpand] 367
[trig2exp] 367
[整数] 367
[除数] 367
[因数] 367
[因数リスト] 367
[GCD] 368
[LCM] 368
[整数]：[素数] 368
[素数判定] 368
[N 番目の素数] 368
[次の素数] 368
[前の素数] 368
[オイラー] 369
[整数]：[除算] 369
[商] 369
[余り] 369
[anMOD p] 369
[中国式剰余] 369
[多項式] 369
[根を求める] 369
[係数] 370
[除数] 370
[因数リスト] 370
[GCD] 370
[LCM] 370
[多項式]：Create（作成） 371
[多項式から係数へ] 371
[係数から多項式へ] 371

[根から係数へ] 371
[根から多項式へ] 371
[乱数] 371
[最小] 372
[多項式] : [代数] 372
[商] 372
[余り] 372
[次数] 372
[次数による因数分解] 372
[係数 GCD] 373
[ゼロカウント] 373
[中国式剰余] 373
[多項式] : [特殊] 373
[円分] 373
[グレブナー基底] 374
[グレブナー余り] 374
[エルミート] 374
[ラグランジュ] 374
[ラゲール] 374
[ルジャンドル] 375
[チェビシェフの Tn] 375
[チェビシェフの Un] 375
[プロット] 375
[関数] 375
[等高線] 375
[アプリ]メニュー 376
関数アプリケーション関数 376
[面積] 376
[極値] 376
[ISECT] 377
[根] 377
[傾き] 377
解くアプリケーション関数 377
[解く] 377
スプレッドシート アプリケーション関数 378
[和] 379
[平均] 379
[償却] 379
[STAT1] 380

STAT2 381
[REGRS] 382
[predY] 383
[PredX] 384
[HypZ1mean] 384
[HYPZ2mean] 385
[HypZ1prop] 385
[HypZ2prop] 386
[HypT1mean] 387
[HypT2mean] 387
[ConfZ1mean] 388
[ConfZ2mean] 388
[ConfZ1prop] 389
[ConfZ2prop] 389
[ConfT1mean] 389
[ConfT2mean] 390
1 変数統計アプリケーション関数 390
[Do1VStats] 390
[SetFreq] 391
[SetSample] 391
2 変数統計アプリケーション関数 391
[PredX] 391
[PredY] 391
[Resid] 391
[Do2VStats] 392
[SetDepend] 392
[SetIndep] 392
推論アプリケーション関数 392
[DoInference] 392
[HypZ1mean] 392
[HypZ2mean] 393
[HypZ1prop] 393
[HypZ2prop] 394
[HypT1mean] 394
[HypT2mean] 395
[ConfZ1mean] 396
[ConfZ2mean] 396
[ConfZ1prop] 396
[ConfZ2prop] 396
[ConfT1mean] 397
[ConfT2mean] 397

[Chi2GOF] ... 397
[Chi2TwoWay] ... 397
[LinRegrTConf- Slope] ... 398
[LinRegrTConfInt] ... 398
[LinRegrTMean-Resp] ... 398
[LinRegrTPredInt] ... 399
[LinRegrTTest] ... 399
金融アプリケーション関数 ... 400
[CalcFV] ... 400
[CalcIPYR] ... 401
[CalcNbPmt] ... 401
[CalcPMT] ... 401
[CalcPV] ... 401
[DoFinance] ... 401
線形ソルバー アプリケーション関数 ... 401
[Solve2x2] ... 402
[Solve3x3] ... 402
[LinSolve] ... 402
三角ソルバーアプリケーション関数 ... 402
[AAS] ... 402
[ASA] ... 403
[SAS] ... 403
[SSA] ... 403
[SSS] ... 403
[DoSolve] ... 403
1 次エクスプローラー関数 ... 403
[SolveForSlope] ... 403
[SolveForYIntercept] ... 404
2 次エクスプローラー関数 ... 404
[解く] ... 404
[DELTA] ... 404
共通アプリケーション関数 ... 404
[CHECK] ... 405
[UNCHECK] ... 405
[ISCHECK] ... 405
[カタログ]メニュー ... 405
[!] ... 406
[%] ... 406
[%TOTAL] ... 407
[(] ... 407
[*] ... 407

[+] 407
[–] 407
[.*] 407
[./] 407
[.^] 407
[/] 407
[:=] 408
[<] 408
[<=] 408
[<>] 408
[=] 408
[==] 408
[EQ] 408
[>] 408
[>=] 408
[^] 409
[a2q] 409
[abcuv] 409
[additionally] 409
[Airy Ai] 409
[Airy Bi] 409
[algvar] 409
[AND] 410
[append] 410
[apply] 410
[assume] 410
[basis] 410
[betad] 410
[betad_cdf] 411
[betad_icdf] 411
[bounded_function] 411
[breakpoint] 411
[canonical_form] 411
[cat] 411
[Cauchy] 412
[Cauchy_cdf] 412
[Cauchy_icdf] 412
[cFactor] 412
[charpoly] 412
[chrem] 413
[col] 413

[colDim] 413
[comDenom] 413
[companion] 413
[compare] 413
[complexroot] 414
[contains] 414
[CopyVar] 414
[correlation] 414
[count] 415
[covariance] 415
[covariance_correlation] 415
[cpartfrac] 415
[crationalroot] 415
[cumSum] 416
[DateAdd] 416
[Day of the week] 416
[DeltaDays] 416
[delcols] 416
[delrows] 417
[deltalist] 417
[deltalist] 417
[Dirac] 417
[e] 417
[egcd] 417
[eigenvals] 418
[eigenvects] 418
[eigVl] 418
[EVAL] 418
[evalc] 418
[evalf] 419
[even] 419
[exact] 419
[EXP] 419
[exponential] 419
[exponential_cdf] 419
[exponential_icdf] 420
[exponential_regression] 420
[EXPR] 420
[ezgcd] 420
[f2nd] 420
[factorial] 421

[float] 421
[fMax] 421
[fMin] 421
[format] 421
[Fourier an] 421
[Fourier bn] 421
[Fourier cn] 422
[fracmod] 422
[froot] 422
[fsolve] 422
[function_diff] 422
[gammad] 422
[gammad_cdf] 423
[gamma_icdf] 423
[gauss] 423
[GF] 423
[gramschmidt] 423
[hadamard] 424
[halftan2hypexp] 424
[halt] 424
[hamdist] 424
[has] 424
[head] 424
[Heaviside] 425
[horner] 425
[hyp2exp] 425
[iabcuv] 425
[ibasis] 425
[icontent] 425
[id] 426
[identity] 426
[iegcd] 426
[igcd] 426
[image] 426
[interval2center] 427
[inv] 427
[iPart] 427
[iquorem] 427
[jacobi_symbol] 427
[ker] 427
[laplacian] 428

[latex] 428
[lcoeff] 428
[legendre_symbol] 428
[length] 428
[lgcd] 429
[lin] 429
[linear_interpolate] 429
[linear_regression] 429
[LineHorz] 429
[LineTan] 430
[LineVert] 430
[list2mat] 430
[lname] 430
[lnexpand] 430
[logarithmic_regression] 430
[logb] 431
[logistic_regression] 431
[lu] 431
[lvar] 431
[map] 431
[mat2list] 432
[matpow] 432
[matrix] 432
[MAXREAL] 432
[mean] 432
[median] 433
[member] 433
[MEMORY] 433
[MINREAL] 433
[modgcd] 433
[mRow] 433
[mult_c_conjugate] 434
[mult_conjugate] 434
[nDeriv] 434
[NEG] 434
[negbinomial] 434
[negbinomial_cdf] 435
[negbinomial_icdf] 435
[newton] 435
[normal] 435
[normalize] 435

[NOT] 436
[odd] 436
[OR] 436
[order_size] 436
[pa2b2] 436
[pade] 436
[part] 436
[peval] 437
[PI] 437
[PIECEWISE] 437
[plotinequation] 437
[polar_point] 437
[pole] 438
[POLYCOEF] 438
[POLYEVAL] 438
[polygon] 438
[polygonplot] 438
[polygonscatterplot] 438
[polynomial_regression] 439
[POLYROOT] 439
[potential] 439
[power_regression] 439
[powerpc] 439
[prepend] 440
[primpart] 440
[product] 440
[propfrac] 440
[ptayl] 440
[purge] 441
[Q2a] 441
[quantile] 441
[quartile1] 441
[quartile3] 441
[quartiles] 441
[quorem] 442
[QUOTE] 442
[randbinomial] 442
[randchisquare] 442
[randexp] 442
[randfisher] 442
[randgeometric] 443

[randperm] 443
[randpoisson] 443
[randstudent] 443
[randvector] 443
[ranm] 443
[ratnormal] 444
[rectangular_coordinates] 444
[reduced_conic] 444
[ref] 444
[remove] 444
[reorder] 445
[residue] 445
[restart] 445
[resultant] 445
[revlist] 445
[romberg] 445
[row] 446
[rowAdd] 446
[rowDim] 446
[rowSwap] 446
[rsolve] 447
[select] 447
[seq] 447
[seqsolve] 447
[shift] 447
[shift_phase] 448
[signature] 448
[simult] 448
[sincos] 448
[spline] 448
[sqrfree] 449
[sqrt] 449
[srand] 449
[stddev] 449
[stddevp] 449
[sto] 449
[sturmseq] 450
[subMat] 450
[suppress] 450
[surd] 450
[sylvester] 450

[table] 451
[tail] 451
[tan2cossin2] 451
[tan2sincos2] 451
[transpose] 451
[trunc] 451
[tsimplify] 452
[type] 452
[unapply] 452
[uniform] 452
[uniform_cdf] 452
[uniform_icdf] 452
[UNION] 453
[valuation] 453
[variance] 453
[vpotential] 453
[VERSION] 453
[weibull] 454
[weibull_cdf] 454
[weibull_icdf] 454
[when] 454
[XOR] 454
[zip] 455
[ztrans] 455
[|] 455
[2] 455
[π] 455
[∂] 455
[Σ] 455
[−] 456
[√] 456
[∫] 456
[≠] 456
[≤] 456
[≥] 456
[►] 456
[i] 456
[-1] 456
ユーザー独自の関数の作成 457

23 変数 459
変数の操作 459
ホーム変数の操作 459
ユーザー変数の操作 460
アプリケーション変数の操作 461
[変数]メニューに関する追加情報 461
変数の修飾 462
ホーム変数 463
アプリケーション変数 464
Function（関数）アプリケーション変数 464
[結果]の変数 465
[極値] 465
[Isect] 465
[根] 465
[符号付き面積] 466
[傾き] 466
Geometry（幾何）アプリケーション変数 466
Spreadsheet（スプレッドシート）アプリケーション変数 466
Solve（解く）アプリケーション変数 467
Advanced Graphing（高度なグラフ作成）アプリケーション変数 467
Statistics 1Var（1 変数統計）アプリケーション変数 468
[結果] 469
[NbItem] 469
[MinVal] 469
[Q1] 469
[MedVal] 469
[Q3] 469
[MaxVal] 469
[ΣX] 469
[ΣX2] 469
[MeanX] 470
[sX] 470
[σX] 470
[serrX] 470
[ssX] 470
Statistics 2Var（2 変数統計）アプリケーション変数 470
[結果] 471
[NbItem] 471
[相関] 471
[係数決定] 471

[sCov] 471
[σCov] 471
[ΣXY] 471
[MeanX] 471
[ΣX] 471
[ΣX2] 471
[sX] 472
[σX] 472
[serrX] 472
[ssX] 472
[MeanY] 472
[ΣY] 472
[ΣY2] 472
[sY] 472
[σY] 472
[serrY] 472
[ssY] 472
Inference（推論）アプリケーション変数 472
[結果] 473
[係数決定] 473
[ContribList] 473
[ContribMat] 473
[相関] 473
[CritScore] 473
[CritVal1] 474
[CritVal2] 474
[DF] 474
[ExpList] 474
[ExpMat] 474
[Inter] 474
[確率] 474
[結果] 474
[serrInter] 474
[serrLine] 474
[serrSlope] 474
[serrY] 474
[傾き] 474
[TestScore] 474
[TestValue] 475
[Yval] 475
Parametric（パラメトリック）アプリケーション変数 475

Polar（極座標）アプリケーション変数 476
Finance（金融）アプリケーション変数 476
Linear Solver（線形ソルバー）アプリケーション変数 477
Triangle Solver（三角ソルバー）アプリケーション変数 477
Linear Explorer（1 次エクスプローラー）アプリケーション変数 477
Quadratic Explorer（2 次エクスプローラー）アプリケーション変数 478
Trig Explorer（三角法エクスプローラー）アプリケーション変数 478
Sequence（数列）アプリケーション変数 478

24 単位および定数 480
単位 480
単位のカテゴリ 480
プレフィックス 481
単位の計算 481
単位ツール 485
[変換] 485
[MKSA] 486
[UFACTOR] 486
[USIMPLIFY] 486
物理定数 486
定数の一覧 489

25 リスト 491
リスト カタログでのリストの作成 491
List（リスト）エディター 493
List（リスト）エディター：ボタンおよびキー 493
[リスト]エディター：[その他]メニュー 493
リストの編集 494
リストへの要素の挿入 495
リストの削除 496
リストの削除 496
すべてのリストの削除 497
ホーム表示でのリストの操作 497
リストの作成 497
リストの格納 497
リストの表示 498
1 つの要素の表示 498
1 つの要素の格納 498
リストの参照 498

リストの送信 498
リスト関数 498
メニュー形式 499
[差異] 499
[交差] 499
[リストの作成] 500
[並べ替え] 500
[反転] 500
[結合] 501
[位置] 501
[サイズ] 501
[ΔLIST] 501
[ΣLIST] 502
[πLIST] 502
リストの統計値の計算 502

26 行列 506
行列の作成および格納 506
Matrix Catalog（行列カタログ）：ボタンおよびキー 507
行列の操作 507
Matrix（行列）エディターを開く 507
Matrix（行列）エディター：ボタンおよびキー 507
[行列]エディター：[その他]メニュー 508
[行列]エディターでの行列の作成 509
ホーム表示での行列の操作 509
行列の格納 511
行列の表示 512
1 つの要素の表示 512
1 つの要素の格納 513
行列の参照 513
行列の送信 513
行列の算術計算 513
スカラーによる乗算および除算 515
2 つの行列の乗算 515
行列のべき乗の計算 516
正方行列による除算 517
行列の逆行列の計算 517
各要素の符号の反転 518
連立一次方程式の計算 518

行列の関数およびコマンド 521
引数の表記法 522
行列関数 522
[行列] 522
[転置] 522
[行列式] 522
[RREF] 522
[作成] 523
[作成] 523
[恒等] 523
[乱数] 523
[ジョルダン] 523
[ヒルベルト] 523
[等長] 524
[ファンデルモンド] 524
[基本] 524
[ノルム] 524
[行ノルム] 524
[列ノルム] 525
[スペクトル ノルム] 525
[スペクトル半径] 525
[条件] 525
[階数] 525
[ピボット] 526
[トレース] 526
[高度] 526
[固有値] 526
[固有ベクトル] 526
[ジョルダン] 526
[対角] 527
[コレスキー] 527
[エルミート] 527
[ヘッセンベルグ] 527
[スミス] 528
[因数分解] 528
[LQ] 528
[LSQ] 528
[LU] 528
[QR] 529

[シュール] 529
[SVD] 529
[SVL] 529
[ベクトル] 529
[クロス積] 529
[ドット積] 530
[L2Norm] 530
[L1Norm] 530
[最大ノルム] 530
例 530
恒等行列 530
行列の転置 530
行既約階段形 531

27 注記および情報 533

Note Catalog（注記カタログ） 533
Note Catalog（注記カタログ）：ボタンおよびキー 533
Note（注記）エディター 534
Note（注記）カタログから注記を作成するには、以下の操作を行います。 534
アプリケーションの注記の作成 536
Note（注記）エディター：ボタンおよびキー 536
大文字および小文字の入力 537
テキスト形式 538
形式オプション 538
数式の挿入 539
注記をインポートするには、以下の操作を行います。 539

28 HP PPL でのプログラミング 541

Program Catalog（プログラム カタログ） 542
Program Catalog（プログラム カタログ）を開く 542
Program Catalog（プログラム カタログ）：ボタンおよびキー 542
新しいプログラムの作成 543
プログラム エディター 545
プログラム エディター：ボタンおよびキー 545
プログラムの実行 550
複数の関数を持つプログラム 551
プログラムのデバッグ 552
プログラムの編集 554
プログラムまたはプログラムの一部のコピー 555

プログラムの削除 555
すべてのプログラムの削除 555
プログラムの内容の削除 556
プログラムの共有 556
HP Prime プログラミング言語 556
変数と可視性 556
変数名の修飾 558
関数、その引数、およびパラメーター 558
プログラム ROLLDIE 558
プログラム ROLLMANY 558
ユーザー キーボード：キー押下のカスタマイズ 560
ユーザー モード 560
キー割り当ての変更 561
キーの名前 562
アプリケーション プログラム 565
専用プログラム関数の使用 565
[表示]メニューの再定義 566
アプリケーションのカスタマイズ 566
例 567
プログラム コマンド 572
[テンプレート]メニューの下にあるコマンド 573
ブロック 573
BEGIN END 573
RETURN 573
KILL 573
分岐 573
IF THEN 573
IF THEN ELSE 573
CASE 574
IFERR 574
IFERR ELSE 574
ループ 574
FOR 574
FOR STEP 575
FOR DOWN 576
FOR STEP DOWN 576
WHILE 577
REPEAT 577
BREAK 578
CONTINUE 578

変数 578
LOCAL 578
EXPORT 578
関数 578
EXPORT 578
VIEW 579
KEY 579
[コマンド]メニューの下にあるコマンド 579
文字列 579
ASC 579
LOWER 579
UPPER 580
CHAR 580
DIM 580
STRING 580
INSTRING 581
LEFT 581
RIGHT 581
MID 582
ROTATE 582
STRINGFROMID 582
REPLACE 582
描画 582
C→PX 583
DRAWMENU 583
FREEZE 583
PX→C 583
RGB 583
ピクセルおよびデカルト 583
ARC_P、ARC 583
BLIT_P、BLIT 584
DIMGROB_P、DIMGROB 584
FILLPOLY_P、FILLPOLY 585
GETPIX_P、GETPIX 585
GROBH_P、GROBH 585
GROBW_P、GROB 585
INVERT_P、INVERT 585
LINE_P、LINE 586
PIXOFF_P、PIXOFF 587
PIXON_P、PIXON 587

RECT_P、RECT 587
SUBGROB_P、SUBGROB 589
TEXTOUT_P、TEXTOUT 589
TRIANGLE_P、TRIANGLE 590
行列 592
ADDCOL 592
ADDROW 592
DELCOL 592
DELROW 592
EDITMAT 592
REDIM 593
REPLACE 593
SCALE 593
SCALEADD 593
SUB 593
SWAPCOL 593
SWAPROW 593
アプリケーション関数 593
STARTAPP 593
STARTVIEW 594
VIEW 595
整数 595
BITAND 595
BITNOT 595
BITOR 595
BITSL 595
BITSR 596
BITXOR 596
B→R 596
GETBASE 596
GETBITS 596
R→B 596
SETBITS 597
SETBASE 597
I/O 597
CHOOSE 597
EDITLIST 598
EDITMAT 598
GETKEY 598
INPUT 599
ISKEYDOWN 600

MOUSE 600
MSGBOX 600
PRINT 601
WAIT 602
その他 602
%CHANGE 602
%TOTAL 602
CAS 603
EVALLIST 603
EXECON 603
→HMS 603
HMS→ 603
ITERATE 604
TICKS 604
TEVAL 604
TYPE 604
変数とプログラム 604
アプリケーション変数 605

29 基本整数算術計算 631
初期設定基数 632
初期設定基数の変更 632
整数算術計算の例 633
基数が混在する算術計算 633
整数操作 634
基底関数 635

30 付録 A：用語集 637

31 付録 B：トラブルシューティング 639
電卓が応答しない 639
リセット方法 639
電卓の電源がオンにならない場合 639
動作限界 639
ステータス メッセージ 639

索引 641

1 前書き

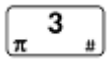
マニュアルの表記法

このガイドでは、操作を実行するために押すキーや選択するメニュー オプションが以下の表記法で示されています。

- シフトを押さずにアクセスできる機能を起動するキーは、そのキーの画像によって示されます。

  例： 、 、

- シフトを押してアクセスできる機能を起動する（または文字を挿入する）キーの組み合わせは、該当するシフト キー（ または ）と、その機能または文字のキーが続けて示されます。

  例： を押すと自然指数関数が起動され、  を押すとシャープ記号（#）が挿入されます。

  シフトを押してアクセスできる機能の名前がキーの組み合わせの後にかっこ付きで示されることもあります。

  例： （クリア）、 （セットアップ）

- 数字を挿入するために押すキーは、その数字によって表されます。

  例：5、7、8

- 画面上の固定テキスト（画面名、フィールド名など）はすべて、太字で表示されます。

  例：**[CAS 設定]**、**[x ステップ]**、**[小数点記号]**

- 画面をタッチして選択するメニュー項目は、その項目の画像によって示されます。

  例： 、 、

> **注記：**メニュー項目を選択するには、指を使用する必要があります。タッチペンなど、指以外ではタッチしても選択できません。

- 入力行に表示される文字は、次のような非プロポーショナル フォントで示されます。

  例：`Function`、`Polar`、`Parametric`、`Ans`

- カーソル キーは、 、 、 、および によって示されます。これらのキーは、画面上のフィールド間を移動するとき、またはリストでオプション間を移動するときに使用します。

- エラー メッセージは、角かっこで囲んで示されます。

  例：[構文エラー]

2 お使いになる前に

HP Prime グラフ電卓は、中学校以降の数学教育のため設計された、簡単に使用できる強力なグラフ電卓です。数百の関数およびコマンドが用意され、記号計算のための数式処理システム（CAS）が内蔵されています。

さらに、幅広い種類の関数およびコマンドに加えて、一連の HP アプリケーションも備えています。HP アプリケーションは、特定の数学分野を掘り下げたり特定のタイプの問題を解いたりするために役立つ特殊なアプリケーションです。たとえば、幾何専用のアプリケーションやパラメトリック方程式専用のアプリケーションがあります。また、連立一次方程式を解くためのアプリケーションや貨幣の時間価値問題を解くためのアプリケーションもあります。

HP Prime 電卓は独自のプログラミング言語も備えており、これを使用して数学的問題を掘り下げたり解いたりすることもできます。

これらの関数、コマンド、アプリケーション、およびプログラミングについては、以降の章で詳しく説明します。この章では、電卓の一般機能、共通操作、および基本的な数学計算について説明します。

準備

初めてこの電卓を使用するときは、バッテリを完全に充電してください。バッテリを充電するには、以下のいずれかの操作を行います。

- HP Prime 電卓のパッケージに含まれている USB ケーブルを使用して、電卓をコンピューターに接続します（充電するには、コンピューターが起動されている必要があります）。
- HP が提供している電源アダプターを使用して、電卓を電源コンセントに接続します。

電卓が起動されているときは、バッテリの記号が画面のタイトル バーに表示されます。記号の表示によってバッテリの充電レベルが示されます。空になったバッテリをフル充電するには、およそ 4 時間かかります。

⚠ **警告！**

バッテリに関する警告

- 火傷や発火のおそれがありますので、バッテリを分解したり、壊したり、穴をあけたりしないでください。また、バッテリの接点をショートさせたり、火や水の中に捨てたりしないでください。
- けがや事故、および機器の故障などの安全に関する問題の発生を防ぐため、この電卓を使用する場合は、電卓に付属しているバッテリ、HP が提供する交換用バッテリ、または HP が推奨する対応するバッテリを使用してください。
- 子供たちがバッテリに触れないようにしてください。
- 電卓を充電しているときに問題が発生した場合は、充電を中止してすぐに HP に連絡してください。

アダプターに関する警告

- 感電、または装置の損傷を防ぐため、AC アダプターは製品の近くの手が届きやすい場所にある AC 電源コンセントにのみ差し込んでください。
- けがや事故、および機器の故障などの安全に関する問題の発生を防ぐため、この製品を使用する際は、電卓に付属している AC アダプター、HP が提供する交換用 AC アダプター、または HP からオプション製品として購入した AC アダプターだけを使用してください。

オン/オフおよびキャンセル操作

電源オン

電卓の電源をオンにするには、 [On Off] を押します。

キャンセル

電卓がオンのときに現在の操作をキャンセルするには、 [Esc Clear] キーを押します。たとえば、入力行に入力した内容をクリアできます。また、メニューや画面を閉じることもできます。

電源オフ

電卓の電源をオフにするには、 [Shift] [On Off] （オフ）を押します。

バッテリ節約のため、数分間操作しないと自動的に電源が切れます。保存された情報および表示されている情報はすべて保持されます。

ホーム表示

ホーム表示は、多くの計算を開始する場所です。ほとんどの数学関数はホーム表示から利用できます。その他のいくつかの関数は数式処理システム（CAS）から利用できます。行った計算は履歴に保持されるため、以前の計算やその結果を再利用できます。

ホーム表示を表示するには、 [Settings] を押します。

CAS 表示

CAS 表示では、記号計算を実行できます。機能はホーム表示とほぼ同じで、独自の計算履歴も備えていますが、CAS 表示にはいくつかの追加機能があります。

CAS 表示を表示するには、 CAS Settings を押します。

保護カバー

電卓には、ディスプレイおよびキーボードを保護するためのスライド式カバーが付いています。両端を握って下に引くとカバーが開きます。

カバーは裏返して電卓の背面にしまうことができます。これにより、電卓の使用時に外したカバーを紛失してしまうのを防ぐことができます。

電卓をより長くお使いいただくために、使用していないときは常にディスプレイおよびキーボードをカバーで保護してください。

ディスプレイ

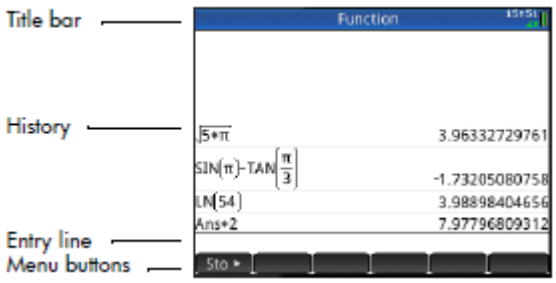
輝度の調整

ディスプレイの輝度を調整するには、 On Off を押しながら + Ans ; キー（輝度を上げる場合）または − Base ; キー（輝度を下げる場合）を押します。 + Ans ; キーまたは On Off キーを押すたびに輝度が 1 段ずつ変化します。

ディスプレイのクリア

- 入力行をクリアするには、 Esc Clear または On Off を押します。
- 入力行および履歴をクリアするには、 Shift Esc Clear （クリア）を押します。

ディスプレイの各部



ホーム表示には 4 つのセクションがあります（上の図を参照）。タイトル バーには、画面名、または現在使用しているアプリケーション名が表示されます（上の例では**[関数]**）。時間、バッテリの残量、およびさまざまな電卓の設定を示す多数の記号も表示されます。これらについては下の表を参照してください。履歴には、過去の計算の記録が表示されます。入力行には、現在入力または変更しているオブジェクトが表示されます。メニュー ボタンには、現在の表示内容に応じたオプションが表示さ

れます。オプションを選択するには、該当するメニュー ボタンをタップします。選択を行わずにメニューを閉じるには、 Esc Clear を押します。

タイトル バーには、記号または文字によるインジケータが表示されます。これらは現在の設定、時間、およびバッテリ残量を示します。

インジケータ	意味
∠° [黄緑色]	現在、角度モードが角度に設定されています
∠π [黄緑色]	現在、角度モードがラジアンに設定されています
⬆S [水色]	[Shift]キーが有効です。キーを押すと、キーに青色で刻印されている機能が有効になります。シフト モードをキャンセルするには、 Shift を押します
[CAS] [白色]	ホーム表示ではなく、CAS 表示を表示しています
A...Z [オレンジ色]	ホーム表示では、[Alpha]キーが有効であることを示します。キーを押すと、キーにオレンジ色で刻印されている文字が大文字で入力されます。詳しくは、10 ページのシフト キーを参照してください CAS 表示では、[Alpha] + [Shift]キーが有効であることを示します。キーを押すと、キーにオレンジ色で刻印されている文字が大文字で入力されます。詳しくは、10 ページのシフト キーを参照してください
a...z [オレンジ色]	ホーム表示では、[Alpha] + [Shift]キーが有効であることを示します。キーを押すと、キーにオレンジ色で刻印されている文字が小文字で入力されます。詳しくは、10 ページのシフト キーを参照してください CAS 表示では、[Alpha]キーが有効であることを示します。キーを押すと、キーにオレンジ色で刻印されている文字が小文字で入力されます。詳しくは、10 ページのシフト キーを参照してください
⬆U [黄色]	ユーザー キーボードが有効です。以降はキーを押すと、カスタマイズによってキーに関連付けられたオブジェクトが入力されます。ユーザー キーボードによる入力内容はユーザーがカスタマイズできます
1U [黄色]	ユーザー キーボードが有効です。次回キーを押したときは、カスタマイズによってキーに関連付けられたオブジェクトが入力されます。ユーザー キーボードによる入力内容はユーザーがカスタマイズできます
[時間]	現在の時刻を示します。初期設定では 24 時間形式ですが、午前/午後形式にすることもできます。詳しくは、19 ページのホーム設定を参照してください
[緑色に灰色の縁]	バッテリの残量を示します

クイック設定メニュー

タイトル バーの右側（時間、バッテリ残量、角度単位モードが表示されている場所）をタップすると、クイック設定メニューが開きます。このメニューでは次の操作を実行できます。

- 角度アイコンの 1 つをタップして角度単位モード（ラジアンまたは度）を変更します。
- 日付と時刻をタップして月別のカレンダーを開きます。各月をナビゲートして目的の日付を見つけることができます。
- 無線アイコンをタップして、HP Classroom Network に接続するか、HP Classroom Network との接続を解除します。

ナビゲーション

HP Prime には、2 つのナビゲーション モードがあります。1 つがタッチ モード、もう 1 つがキー モードです。多くの場合、アイコン、フィールド、メニュー、またはオブジェクトをタップして選択（または選択解除）できます。たとえば、[関数]アプリケーションを開くには、アプリケーション ライブラリでそのアイコンを 1 回タップします。ただし、アプリケーション ライブラリを開くときは、次のキーを押す必要があります： Apps Info

アプリケーション ライブラリでアイコンをタップする代わりに、カーソル キー（▲、▼、◀、▶）を押して目的のアプリケーションまで移動してから、Enter ≈ を押すこともできます。さらに、アプリケーション ライブラリでアプリケーション名の最初の数文字を入力して目的のアプリケーションに移動することもできます。その後、アイコンをタップするか Enter ≈ を押してアプリケーションを開きます。

操作によっては、タッチのみまたはキーとタッチの組み合わせのどちらでも実行できることがあります。たとえば、切り替えオプションの選択を解除するには、そのオプションを 2 回タップするか、または矢印キーを使用してそのフィールドに移動し、画面の下部にあるタッチ ボタン（この場合は √ ）をタップします。

注記：項目を選択するには、指または静電容量式タッチペンを使用してタッチする必要があります。

タッチ ジェスチャ

HP Prime 電卓では、以下のタッチ ジェスチャを認識します。

- タップ：画面上の項目をポイントしてから、1 本の指で軽く叩いてその項目を選択します。
- タップして保持：画面上に指を置き、しばらくそのままにします。
- スクロール：画面上に指を置き、上下左右または斜めに動かすと、ページまたは画像が上下左右または斜めに動きます。
- 1 本指スライド：画面をスクロールするには、画面上で移動したい方向に 1 本の指を軽く滑らせます。[幾何]アプリケーションの[プロット表示]では、オブジェクトを押したままドラッグして移動できます。[スプレッドシート]アプリケーションと[1 変数統計]アプリケーションと[2 変数統計]アプリケーションの[数値表示]、[リスト]エディター、および[行列]エディターで複数のセルを選択するには、1 つのセルをタップしたままドラッグして、以降のセルを選択します。この選択した要素は、1 つの値と同様にコピーして貼り付けることができます。
- 2 本指ピンチ ズーム：画面上で 2 本の指を互いに離した状態にして置き、その 2 本の指の間隔を狭めるとズームアウト（縮小）できます。画面上で 2 本の指を近づけた状態にして置き、その 2 本の指の間隔を拡げるとズームイン（拡大）できます。[スプレッドシート]アプリケーションでは、このジェスチャを使って列の幅および行の高さを調整できます。

アプリケーション、エディター、および入力フォームによっては、一部のタッチ ジェスチャに対応していなかったり、機能が異なっていたりする場合があります。以下のガイドラインを参考にしてください。

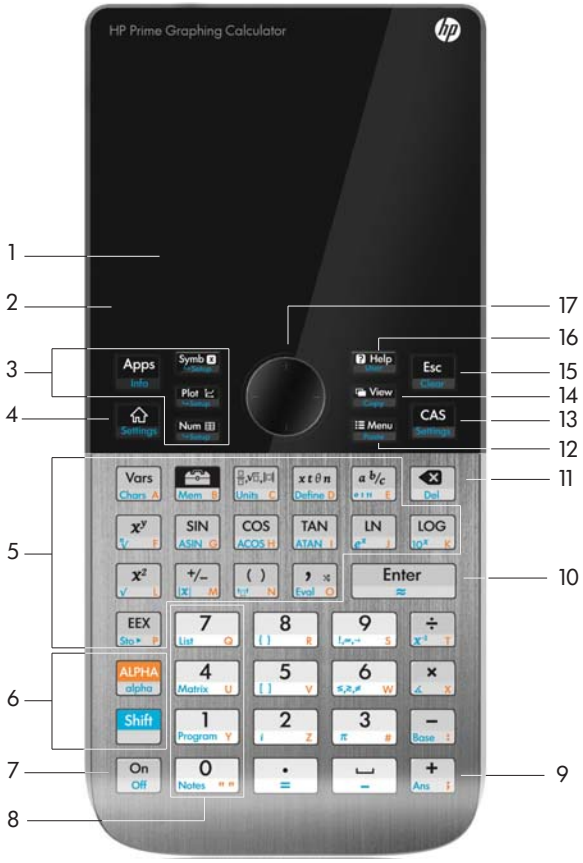
- [プロット表示]では、2 本指ピンチ ズーム ジェスチャを左右方向に実行すると、x 軸のみが拡大または縮小されます。2 本指ピンチ ズームを上下方向に実行すると、y 軸のみが拡大または縮小されます。2 本指ピンチ ズームを斜め方向に実行すると、正方形に拡大または縮小されます（つまり、両方の軸が拡大または縮小されます）。[幾何]アプリケーションでは、斜め方向の拡大または縮小のみがサポートされています。
- [数値表示]では、2 本指ピンチ ズームを上下方向に実行すると、表で現在選択されている行が拡大または縮小されます。拡大すると x 値の公差が小さくなり、縮小すると大きくなります。2 本指ピンチ ズームを左右方向に実行すると、列の幅が変更されます。

キーボード

次の表に、キーボード各部の機能を示します。番号は、次のページの図に示された番号に対応します。

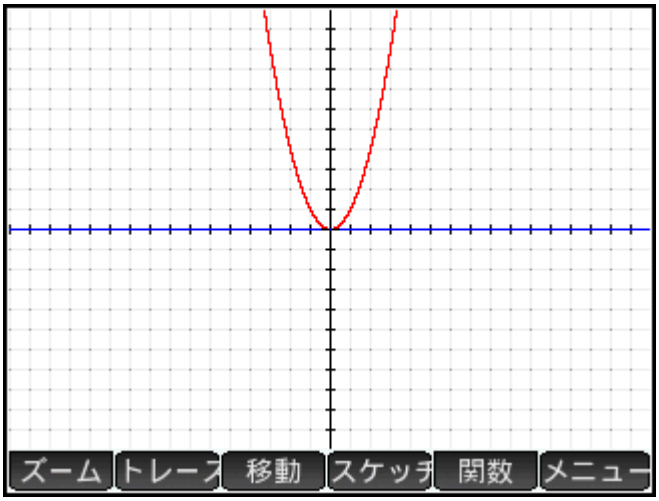
番号	機能
1	LCD およびタッチスクリーン: 320×240 ピクセル
2	コンテキスト依存タッチ ボタン メニュー
3	HP アプリケーションのキー
4	ホーム表示およびホーム基本設定
5	一般的な数学および科学の関数
6	[Alpha]キーおよび[Shift]キー
7	[On]、[Cancel]、および[Off]キー
8	リスト、行列、プログラム、および注記のカタログ

番号	機能
9	最後の解答キー（[Ans]）
10	[Enter]キー
11	バックスペースおよび[Delete]キー
12	[Menu]（および[Paste]）キー
13	CAS（および CAS 基本設定）キー
14	[View]（および[Copy]）キー
15	[Escape]（および[Clear]）キー
16	[Help]キー
17	ロッカー ホイール（カーソル移動用）



コンテキスト依存メニュー

コンテキスト依存メニューは、画面の下部に 1 列に表示されます。



使用できるオプションはコンテキスト、つまり表示している表示によって異なります。メニュー項目を選択するには、タッチ操作を使用する必要があります。

コンテキスト依存メニューのボタンには 2 種類あります。

- メニュー ボタン：タップするとポップアップ メニューが表示されます。これらのボタンは上部の角がとがっています（上の図の ズーム など）。
- コマンド ボタン:タップするとコマンドが実行されます。これらのボタンは角が丸くなっています（上の図の 移動 など）。

入力キーおよび編集キー

キー	用途
0 ~ 9	数字を入力します
On または Esc	現在の操作をキャンセルするか、入力行をクリアします
Enter	入力を確定するか、操作を実行します。計算式では、Enter は「=」のように機能します。メニュー キーとして OK または 開始 が表示されているときは、Enter を押すと OK または 開始 を押したことと同じになります
+/−	負数を入力します。たとえば、 –25 を入力するときは、+/− 25 と押します **注記：**この操作は、減算キー（ − ）で実行される操作とは異なります
	一般的な算術式を表す書式設定済みテンプレートのパレットを表示します
$x\,t\,\theta\,n$	独立変数（現在アクティブなアプリケーションに応じて X、T、θ、または N のいずれか）を入力します

キー	用途
Shift 6	比較演算子およびブール演算子のパレットを表示します
Shift 9	一般的な数学記号およびギリシャ文字のパレットを表示します
Shift a b/c	コンテキストに応じて度、分、または秒記号を自動的に挿入します
Del	カーソルの左にある文字を削除します。フィールドが選択されている場合は、初期設定値に戻します（初期設定値がある場合）
Shift Del	カーソルの右にある文字を削除します
Shift Esc Clear（クリア）	画面に表示されているすべてのデータ（履歴を含む）をクリアします。設定画面（[プロットのセットアップ]など）では、すべての設定値を初期設定値に戻します
▲ ▼ ◀ ▶	ディスプレイ内でカーソルを移動します。Shift ▼ を押すとメニューまたは画面の末尾まで移動し、Shift ▲ を押すと先頭まで移動します。これらのキーは、ロッカー ホイールの方向を表します。ロッカー ホイールでは斜め方向の移動もサポートされます
Shift Vars Chars A	使用可能なすべての文字を表示します。文字を入力するには、カーソル キーを使用して目的の文字まで移動し、OK をタップします。複数の文字を選択するには、1 つの文字を選択してエコーをタップする操作を繰り返し、最後に OK を押します。文字のページは多数あります。その他をタップしてから Unicode ブロックを選択することで、そのブロックにジャンプできます。フリックしてページをすばやく切り替えることもできます

シフト キー

キーの下部に刻印されている操作や文字を使用するためのシフト キーには、Shift と ALPHA alpha の 2 種類があります。

キー	用途
Shift	Shift を押すと、キーに水色で刻印されている操作を使用できます。たとえば、ホーム表示の設定にアクセスするには、Shift Settings を押します
ALPHA alpha	ALPHA alpha を押すと、キーにオレンジ色で刻印されている文字を入力できます。たとえば、ホーム表示で Z を入力するには、ALPHA alpha を押してか

キー	用途
	ら [2 i z] を押します。小文字を入力する場合は、[ALPHA alpha] [Shift] を押してから文字キーを押します。CAS 表示では、[ALPHA alpha] を押して文字キーを押すと小文字が入力され、[ALPHA alpha] [Shift] を押して文字キーを押すと大文字が入力されます

テキストの追加

キーにオレンジ色で刻印されている文字は直接入力できます。これらの文字を入力するには、[ALPHA alpha] キーおよび [Shift] キーと組み合わせる必要があります。大文字と小文字のどちらでも入力できますが、入力方法は CAS 表示とホーム表示で逆になります。

キー	ホーム表示での用途	CAS 表示での用途
[ALPHA alpha]	次の文字を大文字にします	次の文字を子文字にします
[ALPHA alpha] [ALPHA alpha]	ロック モード:モードがリセットされるまで、すべての文字を大文字にします	ロック モード:モードがリセットされるまで、すべての文字を小文字にします
[Shift]	大文字にロックされているときに、次の文字を小文字にします	子文字にロックされているときに、次の文字を大文字にします
[ALPHA alpha] [Shift]	次の文字を子文字にします	次の文字を大文字にします
[ALPHA alpha] [Shift] [ALPHA alpha]	ロック モード:モードがリセットされるまで、すべての文字を小文字にします	ロック モード:モードがリセットされるまで、すべての文字を小文字にします
[Shift]	子文字にロックされているときに、次の文字を大文字にします	大文字にロックされているときに、次の文字を小文字にします
[Shift] [ALPHA alpha]	小文字にロックされているときに、モードがリセットされるまで、すべての文字を大文字にします	大文字にロックされているときに、モードがリセットされるまで、すべての文字を小文字にします
[ALPHA alpha]	大文字ロック モードをリセットします	小文字ロック モードをリセットします
[ALPHA alpha] [ALPHA alpha] [ALPHA alpha] [ALPHA alpha]	小文字ロック モードをリセットします	大文字ロック モードをリセットします

次のキーを押して文字パレットを表示し、テキスト（およびその他の文字）を入力することもできます：[Shift] [Vars Chars A]

計算キー

最も一般的な関数は、キーボードのキー（または Shift キーと組み合わせたキー）に割り当てられています。

例 1：SIN(10)を計算するには、 SIN ASIN G 10 と押してから、 Enter ≈ を押します。結果は – 0.544… になります（角度単位モードがラジアンの場合）。

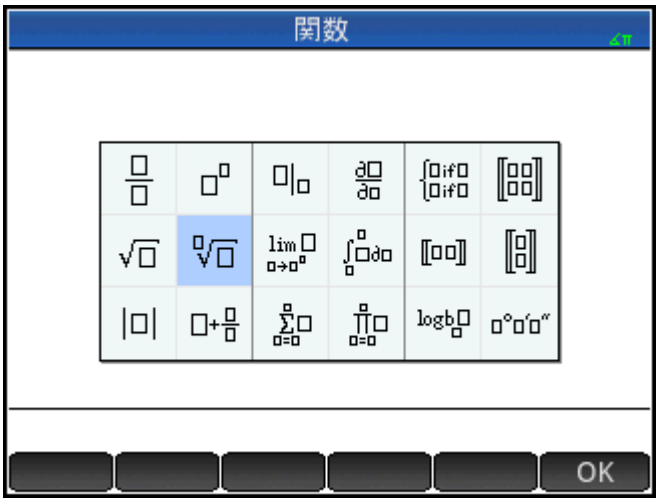
例 2：256 の平方根を求めるには、 Shift x^2 256 と押してから、 Enter ≈ を押します。結果は 16 になります。 Shift キーを押すことによって、次に押すキーに青色で刻印された演算子（この場合は x^2 キーの√）が有効になります。

キーボードに刻印されていない数学関数は、**[数学]**メニュー、**[CAS]**メニュー、および**[カタログ]**メニューから使用できます。

注記：オペランドおよび演算子の入力順序は入力モードによって決まります。初期設定の入力モードは*テキストブック*です。この場合は、紙に式を書くときと同じ順序でオペランドおよび演算子を入力します。入力モードを逆ポーランド記法にした場合は、入力順序が異なります。

数学テンプレート

数学テンプレート キー（ Units C ）を使用すると、基本的な計算、ベクトル、行列、および 60 進数のフレームワークを簡単に挿入できます。このキーを押すと、定数や変数を追加できる書式設定済みフレームワークのパレットが表示されます。テンプレートを使用するには、そのテンプレートをタップします（または矢印キーで目的のテンプレートまで移動して Enter ≈ を押します）。その後、計算を完成させるために必要なコンポーネントを入力します。



例：945 の立方根を求めるには、以下のように操作します。

1. ホーム表示で [Units] を押します。
2. [√□] を選択します。

   計算のフレームワークが入力行に表示されます：□√□
3. テンプレートの中で、塗りつぶしのボックスは入力が必須です。枠だけのボックスはオプションです。

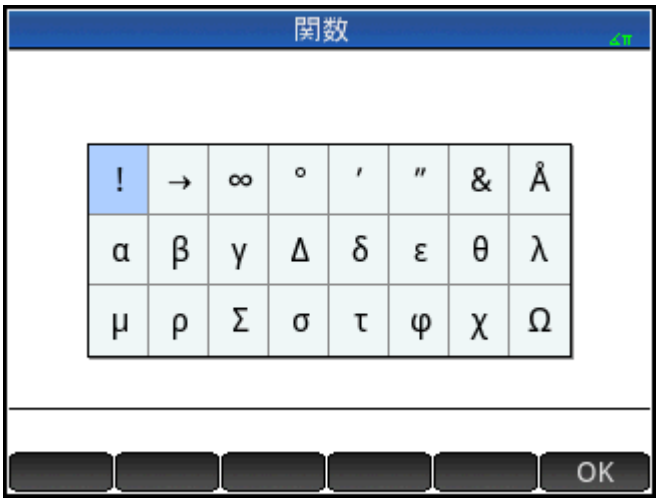
   3 ▶ 945
4. [Enter] を押して結果を表示します。結果は 9.813… になります。

テンプレート パレットを使用すると時間を大幅に節約できます。特に微積分の計算では大いに役立ちます。

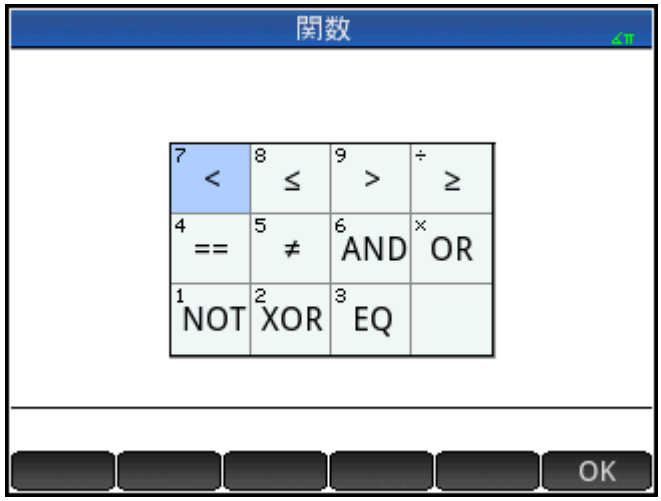
パレットは式の作成中にいつでも表示できます。必ずしもテンプレートから始める必要はありません。式の作成中に必要に応じて必要なだけテンプレートを追加できます。

数学ショートカット

数学テンプレートのほかに、特殊文字を入力するためのパレットも用意されています。たとえば、[Shift] [9] を押すと、下の図に示すような特殊記号パレットが表示されます。目的の文字をタップして選択します（または目的の文字まで移動して [Enter] を押します）。



また、[Shift] [6] を押すと、関係パレットが表示されます。このパレットには、数学やプログラミングでよく使用する演算子が表示されます。このパレットでも目的の文字をタップするだけです。



数学ショートカットには [xtθn] もあります。このキーを押すと、使用しているアプリケーションに応じて X、T、θ、または N が挿入されます（詳しくは、各アプリケーションに関する章を参照してください）。

また、[Shift] [a b/c] を押すと、度、分、または秒記号が入力されます。式に度の記号が含まれていない場合は「°」が入力され、直前に度単位の値がある場合は「′ 」が入力され、直前に分単位の値がある場合は「″ 」が入力されます。

つまり、36 [Shift] [a b/c] 40 [Shift] [a b/c] 20 [Shift] [a b/c] と入力すると、36°40′ 20″ と表示されます。詳しくは、15 ページの 60 進数を参照してください。

分数

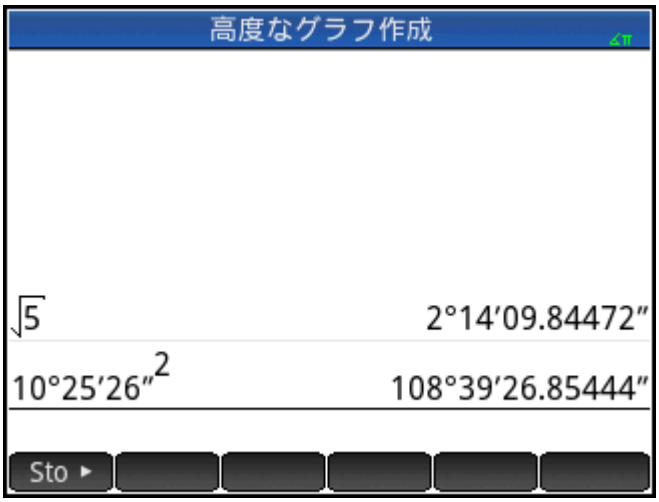
分数キー（ [a b/c] ）を押すと、3 種類の表記が順番に表示されます。たとえば、結果が 5.25 の場合、 [a b/c] を押すと最初に一般的な分数表記 21/4 に変換されます。 [a b/c] をもう一度押すと、帯分数表記 5 + 1/4 に変換されます。さらにもう一度押すと、小数表記 5.25 に戻ります。

HP Prime では、正確な小数および帯分数表現を求められない場合、近似値が用いられます。たとえば、√5 を小数で表すと近似値は 2.236… になります。 [a b/c] を押すと $\frac{219602}{98209}$ に変換され、もう一度押すと $2+\frac{23184}{98209}$ に変換されます。 [a b/c] をさらにもう一度押すと元の小数表記に戻ります。

60 進数

10 進数の結果を 60 進数で表示できます。つまり、結果を 60 で割った値が得られます。これにより、度、分、秒や時、分、秒単位の値を求めることができます。たとえば、$\frac{11}{8}$ と入力すると、10 進数の結果は 1.375 になります。このとき [Shift] [a b/c] を押すと 1°22′ 30 に変換されます。 [Shift] [a b/c] をもう一度押すと 10 進表記に戻ります。

HP Prime 電卓では、正確な結果を求められない場合、近似値が用いられます。たとえば、√5 を小数で表すと近似値は 2.236… になります。 [Shift] [a b/c] を押すと 2°14′ 9.84472 に変換されます。



注記：度および分の値は整数、分および秒の値は正の数でなければなりません。秒の値以外、小数は許容されません。

HP Prime 電卓では、60 進形式の値がひとまとまりとして扱われます。そのため、60 進値に対する演算は値全体に行われます。たとえば、10°25′ 26″ ^2 と入力すると、秒の値だけでなく値全体が 2 乗されます。この場合、結果は 108°39′ 26.8544″ になります。

EEX キー（10 のべき乗）

5×10^4 や 3.21×10^{-7} のような数は、*科学的記数法*、つまり 10 のべき乗の形で表されます。こうすることで、50 000 や 0.000 000 321 のような形式よりも値として扱いやすくなります。この形式で入力するときは、 [EEX] 機能を使用します。 [×] 10 [x^y] を使用するよりも簡単に数を入力できます。

例：次の式を計算します。

$$\frac{(4\times10^{-13})(6\times10^{23})}{3\times10^{-5}}$$

1. **[ホーム設定]**ウィンドウを開きます。

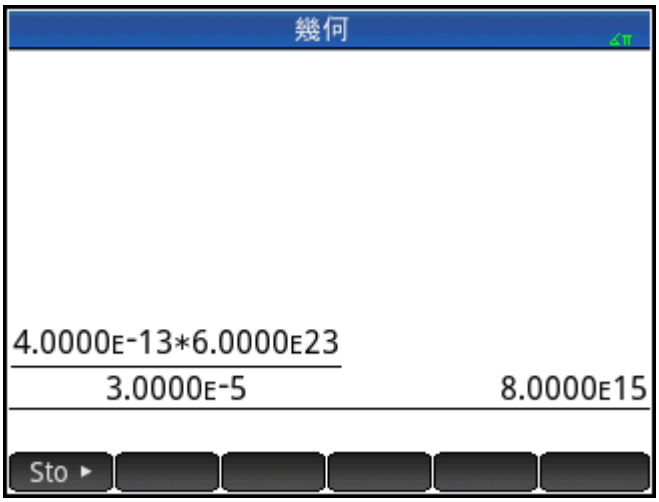
   Shift Settings

2. **[数値形式]**メニューから[関数]を選択します。

3. Settings を押してホーム表示に戻ります。

4. 4 EEX +/− 13 × 6 EEX 23 ÷ 3 EEX +/− 5 と入力します。

5. Enter を押します。



結果は 8.0000E15 になります。これは、8×10^{15} に相当します。

メニュー

メニューにはさまざまなオプションが表示されます。次の図のように、メニューによってはサブメニューやさらにそのサブメニューが表示されることもあります。

メニュー項目の選択

メニュー項目を選択する方法は 2 つあります。

- 項目を直接タップします。
- 矢印キーを使用して目的の項目まで移動してから、 OK をタップするか Enter を押します。

注記：画面の下部に並んでいるメニュー ボタンを開くには、タップする必要があります。

ショートカット

- メニューの一番上の項目から一番下の項目に直接移動するには、 ▲ を押します。
- メニューの一番下の項目から一番上の項目に直接移動するには、 ▼ を押します。
- メニューの一番下の項目までジャンプするには、 Shift ▼ を押します。
- メニューの一番上の項目までジャンプするには、 Shift ▲ を押します。
- 特定の項目にジャンプするには、項目名の最初の数文字を入力します。
- 特定の項目にジャンプするには、メニュー内での項目の番号を入力します。

メニューの閉じ方

メニュー項目を選択すると、メニューは自動的に閉じます。何も選択せずにメニューを閉じるときは、 On Off または Esc Clear を押します。

ツールボックス メニュー

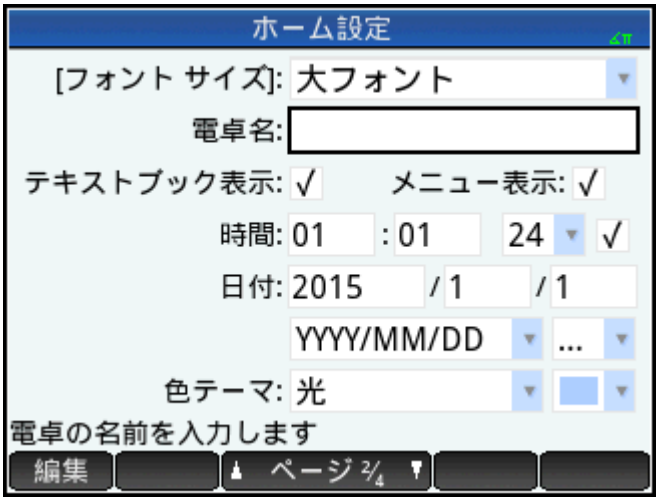
ツールボックス メニュー（ Mem ）は、数学やプログラミングで使用する関数およびコマンドがまとめられた一連のメニューです。[数学]メニュー、[CAS]メニュー、および[カタログ]メニューから 400 を超える関数およびコマンドを利用できます。

入力フォーム

入力フォームは、データの入力またはオプションの選択を行うためのフィールドが 1 つ以上ある画面です。ダイアログ ボックスとも呼ばれます。

- 任意のデータを入力できるフィールドの場合は、フィールドを選択し、データを入力して、OK をタップします（最初に 編集 をタップする必要はありません）。
- メニューから項目を選択できるフィールドの場合は、フィールド（またはそのラベル）をタップし、フィールドをもう一度タップしてオプションを表示し、目的の項目をタップします（開いたメニューから項目を選択するときは、カーソル キーを押して目的の項目まで移動し、Enter を押すこともできます）。
- フィールドが切り替えフィールド(選択するかしないかを指定するフィールド)の場合は、フィールドをタップして選択し、もう一度タップして選択を切り替えます（フィールドを選択してから √ をタップすることもできます）。

次の図は、3 種類すべてのフィールドがある入力フォームの例です。



[電卓名]は任意のデータを入力できるフィールド、**[フォント サイズ]**はメニュー オプションを備えたフィールド、**[テキストブック表示]**は切り替えフィールドです。

入力フォームのフィールドのリセット

フィールドを初期設定値にリセットするには、フィールドを選択して Del を押します。すべてのフィールドを初期設定値にリセットするには、Shift Esc Clear（クリア）を押します。

システム全体の設定

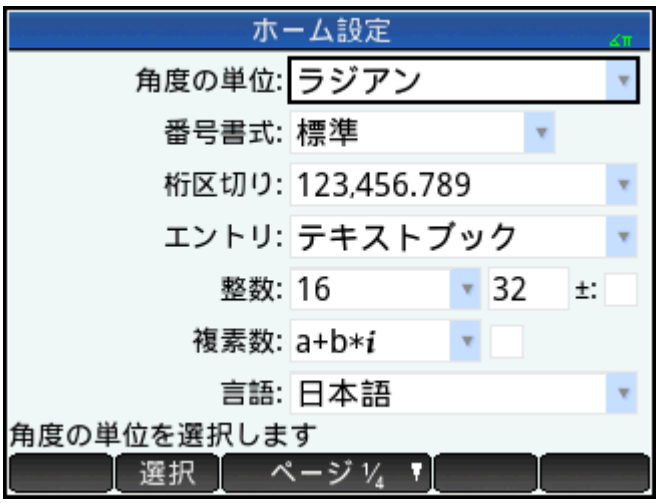
システム全体の設定では、ウィンドウの外観、数値の形式、プロットのスケール、初期設定で計算に使用する単位など、さまざまな設定を指定できます。

システム全体の設定には 2 種類あります。1 つがホーム設定、もう 1 つが CAS 設定です。ホーム設定は、ホーム表示およびアプリケーションに適用されます。CAS 設定は、数式処理システムでの計算方法に影響します。CAS 設定については、第 3 章を参照してください。

ホーム設定はアプリケーションに適用されますが、いくつかのホーム設定はアプリケーション内で変更できます。たとえば、角度の単位をホーム設定でラジアンに設定した後、[極座標]アプリケーション内で度を選択できます。この場合、別の角度単位を選択したアプリケーションを開くまで度が使用されます。

ホーム設定

ホーム表示の設定（およびアプリケーションの初期設定）を指定するには、[ホーム設定]入力フォームを使用します。 Shift Settings （設定）を押して[ホーム設定]入力フォームを開きます。設定画面は 4 ページあります。



ページ 1

設定	オプション
[角度の単位]	**[度]**：円を 360 度として表します **[ラジアン]**：円を 2π ラジアンとして表します ここで設定した角度モードはホーム表示と現在のアプリケーションの両方で使用されます。これにより、現在のアプリケーションで三角関数の計算を行ったときに、ホーム表示で同じ結果が表示されます
[番号書式]	ホーム表示で行う計算で常に使用する数値形式を設定します • **[標準]**：結果を完全な精度で表示します • **[固定]**：結果を指定の小数位までに丸めて表示します。このオプションを選択すると、小数位を指定するための新しいフィールドが表示されます。たとえば 123.456789 は、**[固定 2]**形式で小数点以下 2 桁で丸めると 123.46 になります • **[関数]**：結果を指数および 1 桁の整数部と指定の小数位までの小数部で表示します。たとえば 123.456789 は、**[Scientific]**形式で小数点以下 2 桁で丸めると 1.23E2 になります • **[工学]**：結果を 3 の倍数の指数および 1 桁の整数部と指定の有効桁数の小数部で表示します。例：123.456E7 は、**[工学]**形式で有効数字 2 桁とすると 1.23E9 になります
[エントリ]	• **[テキストブック]**：引数を上下に置くなど、紙に書くときとほぼ同じ方法で式を入力します。つまり、入力内容を 2 次元で表現できます • **[代数]**：式を 1 行で入力します。入力内容は常に 1 次元になります。

設定	オプション
	• **[RPN]**：逆ポーランド記法で入力します。先に式の引数を入力してから、演算子を入力します。演算子を入力すると、これまでに入力した内容が自動的に評価されます。
[整数]	整数計算の初期設定基数を選択します。2 進数、8 進数、10 進数、または 16 進数を選択できます。整数の表現に使用するビット数、および整数を符号付きにするかどうかも設定できます
[複素数]	複素数の表示形式を選択します。**[(a,b)]**または**[a+b*i]**を選択できます フィールドの右側にはラベルのないチェック ボックスがあります。実数の入力から複素数を出力できるようにする場合にこれを選択します
[言語]	メニュー、入力フォーム、およびオンライン ヘルプの表示に使用する言語を選択します
[小数点記号]	**[ドット]**または**[コンマ]**を選択します。たとえば、ドット モードの場合は 12456.98、コンマ モードの場合は 12456,98 のように表示されます。ドット モードでは、リストや行列の要素の区切りおよび関数の引数の区切りにコンマが使用されます。コンマ モードでは、これらの区切りにセミコロンが使用されます

ページ 2

設定	オプション
[フォントサイズ]	基本的な表示のフォント サイズを小、中、大から選択します
[電卓名]	電卓の名前を入力します
[テキストブック表示]	これを選択すると、式および結果がテキストブック形式（紙に書くときと同じ形式）で表示されます。選択しない場合は、式および結果が代数形式（1 次元の形式）で表示されます。たとえば は、代数形式では`[[4,5],[6,2]]`と表示されます
[メニュー表示]	**[数学]**メニューおよび**[CAS]**メニューのコマンドを説明的な名前で表示するか一般的な数学略記で表示するかを選択します。初期設定では、関数が説明的な名前で表示されます。数学略記で表示したい場合は、このオプションの選択を解除します
[時間]	時間を設定し、形式を選択します。24 時間形式または午前/午後形式を選択できます。右側にあるチェック ボックスでは、画面のタイトル バーに時間を表示するかどうかを選択できます
[日付]	日付を設定し、形式を選択します。**[YYYY/MM/DD]**、**[DD/MM/YYYY]**、または**[MM/DD/YYYY]**を選択できます
[色テーマ]	**[淡色]**：明るい背景に黒いテキストを表示します **[濃色]**：暗い背景に白いテキストを表示します 右側にあるオプションでは、影付け（強調表示など）の色を選択できます

ページ 3

[ホーム設定]入力フォームの 3 ページ目では、試験モードの設定ができます。試験モードを使用すると、電卓の特定の機能を一定期間だけ無効にして、パスワードを入力しないと使用できないようにす

ることができます。この機能は、試験中に学生が電卓を適切に使用していることを確認する必要がある試験の監督者を主な対象としています。

ページ 4

HP Prime 電卓で無線接続がサポートされている場合は、[ホーム設定]の 4 番目のページが表示されます。**[ホーム設定]**入力フォームの 4 ページ目では、HP Prime Wireless Kit を介して HP Prime 電卓を HP Wireless Classroom Network に接続するための設定ができます。詳しくは、http://www.hp.com/support を参照してください。

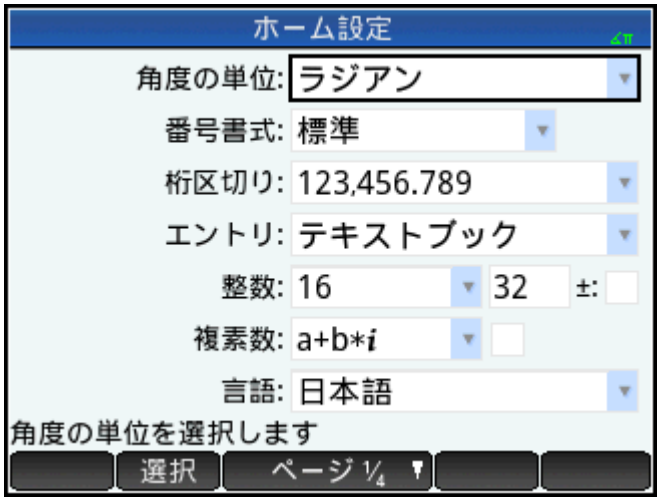
オプション	設定
[ネットワーク名]	• ネットワークが利用できません • ネットワーク 1 • ネットワーク 2（など）
[状態]	• アダプターが見つかりません • 切断済み • 接続済み
[RF バージョン]	• アダプターが見つかりません • アダプター ファームウェアのバージョン

ホーム設定の指定

この例では、数値形式を初期設定の[標準]から[関数]で小数点以下 2 桁に変更する方法について説明します。

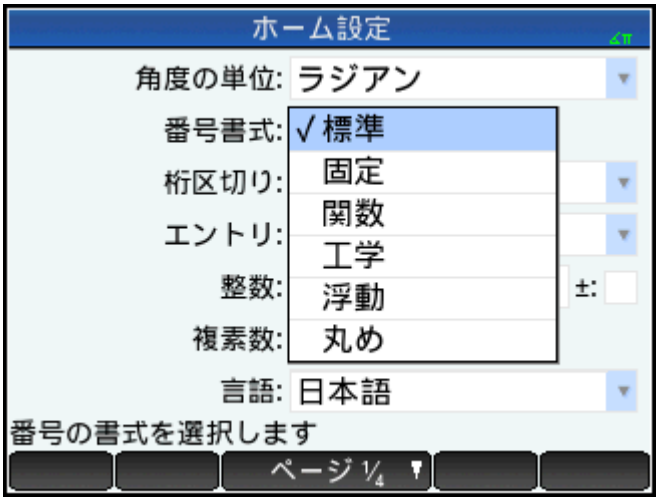
1. Shift Settings（設定）を押して[ホーム設定]入力フォームを開きます。

   [角度の単位]フィールドが選択されています。



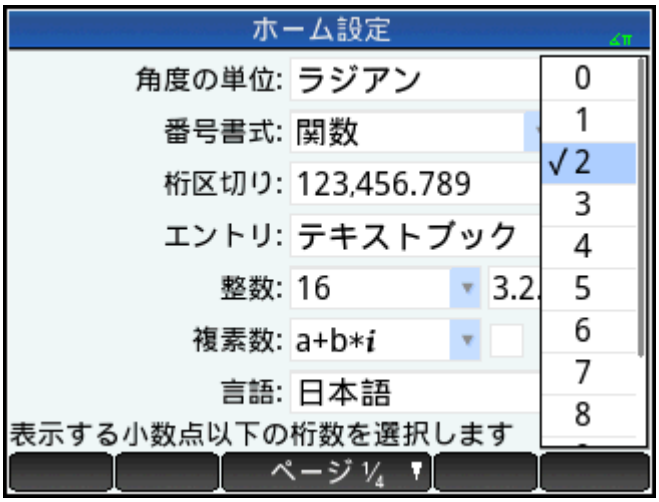
2. **[番号書式]**フィールドまたはそのラベルをタップします。これによってフィールドが選択されます（ ▼ を押して選択することもできます）。

3. **[番号書式]**をもう一度タップします。数値形式オプションのメニューが表示されます。



4. **[関数]**をタップします。オプションが選択され、メニューが閉じます（オプションを選択するときは、カーソル キーを押して目的の項目まで移動し、 Enter を押すこともできます）。

5. **[番号書式]**フィールドの右側に数字が表示されます。これは、現在設定されている小数位桁数を示します。桁数を**[2]**に変更するには、現在の数字を 2 回タップしてから、表示されるメニューで**[2]**をタップします。



6. Settings を押してホーム表示に戻ります。

数学計算

一般的な数学演算はキーボードから入力できます（12 ページの計算キーを参照）。それ以外の数学関数は各種のメニューから利用できます（16 ページのメニューを参照）。

HP Prime では、1×10^{-499} より小さい数が 0 として表示されます。表示可能な最大値は $9.99999999999 \times 10^{499}$ です。これよりも大きい数はこの値で表示されます。

作業場所

電卓の基本となる場所はホーム表示（ Settings ）です。記号を使用しない計算はすべてここで実行できます。数式処理システムを使用できる CAS 表示で計算を行うこともできます。実際には、ホーム表示で式を入力しながら**[CAS]**メニュー（ツールボックス メニューの 1 つ）の関数を使用したり、CAS 表示で式を入力しながら**[数学]**メニュー（ツールボックス メニューの 1 つ）の関数を使用したりすることができます。

入力形式の選択

最初に入力の形式を選択する必要があります。以下の 3 種類の中から選択できます。

- テキストブック

  $$\frac{LN(5)}{\pi}$$

  引数を上下に置くなど、紙に書くときとほぼ同じ方法で式を入力します。つまり、上の例のように入力内容を 2 次元で表現できます。

- 代数

  LN(5)/π

  式を 1 行で入力します。入力内容は常に 1 次元になります。

- 逆ポーランド記法（RPN）（CAS 表示では利用不可）

  先に式の引数を入力してから、演算子を入力します。演算子を入力すると、これまでに入力した内容が自動的に評価されます。そのため、演算子が 2 つ含まれる式の場合は、各演算子の式を 2 段階で入力する必要があります。上記の例の計算は次のようになります。

  ステップ 1：5 [LN] を入力します。5 の自然対数が計算され、履歴に表示されます。

  ステップ 2：[Shift] [3] [÷] を入力します。π が前の結果に除数として適用されます。

注記：[ホーム設定]画面の 2 ページ目では、式の表示方法をテキストブック形式にするかどうかを指定できます。これは、ホーム表示および CAS 表示の各履歴で式をどのように表示するかを指定する設定です。ここで説明している**[エントリ]**設定とは異なることに注意してください。

式の入力

入力モードが**[テキストブック]**の場合の入力方法を以下に示します。

- 式には数字、関数、および変数を含めることができます。
- 関数を入力するには、該当するキーを押すか、ツールボックス メニューを開いて関数を選択します。アルファベット キーで関数名を入力して関数を入力することもできます。
- 式を入力し終えたら、[Enter] を押して評価します。

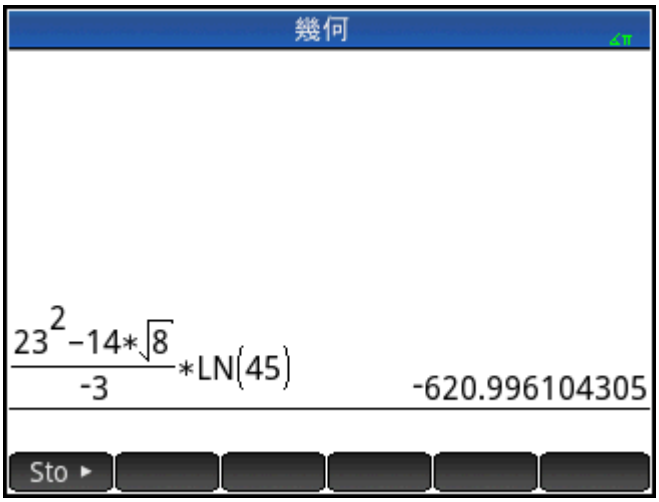
式の入力中に入力ミスをした場合は、以下のいずれかの操作を行います。

- カーソルの左にある文字を削除するには、 Del を押します。
- カーソルの右にある文字を削除するには、 Shift Del を押します。
- 行全体をクリアするには、 On Off または Esc Clear を押します。

例

$\frac{23^2-14\sqrt{8}}{-3}\ln(45)$ を計算するには、以下の操作を行います。

▲ ( ) 23 x^2 − 14 Shift x^2 8 ▶ ▶ ÷ +/− 3 ▶ LN 45 Enter ≈ と入力します。



この例には、覚えておくべき重要なポイントがいくつか含まれています。

- 区切り記号（かっこなど）の重要性
- 負数の入力方法
- 暗黙的な乗算と明示的な乗算の使用

かっこ

上の例から分かるように、LN() などの関数では引数を囲むためのかっこが自動的に追加されます。一方、一連のオブジェクトをひとまとまりの計算対象として示すためのかっこは、 ( ) を押して手動で追加する必要があります。かっこを使用することで、計算のあいまいさを避けることができます。上の例では、分子全体を −3 で割ることを示すために、分子全体をかっこで囲んでいます。かっこがないと、14√8 だけが −3 で割られることになります。

以下の例に、かっこの使い方、およびかっこを閉じるためのカーソル キーの使い方を示します。

入力	式
SIN ASIN G 45 + Ans ; Shift 3 π #	$\sin(45+\pi)$
SIN ASIN G 45 ▶ + Ans ; Shift 3 π #	$\sin(45)+\pi$
Shift √ x^2 L 85 ▶ × ∠ X 9	$\sqrt{85}\times 9$
Shift √ x^2 L 85 × ∠ X 9	$\sqrt{85\times 9}$

代数の優先順位

HP Prime 電卓では、以下の優先順位に従って式が評価されます。優先順位が同じ関数は左から右に評価されます。

1. かっこ内の式（かっこが入れ子になっている場合は内側から外側の順に評価されます）
2. !（逆数）、√（平方）
3. n 乗根
4. 10^n（10 のべき乗）
5. 否定、乗算、除算、mod（法）
6. 加算、減算
7. 関係演算子（<、>、≤、≥、==、≠、=）
8. AND、NOT
9. OR、XOR
10. |（条件）の左側の引数
11. 変数割り当て（:=）

負数

負数を入力するときまたは負符号を挿入するときは、 +/− |x| M を押すことをお勧めします。 − Base ; を押すと、場合によっては減算演算子と解釈されて、次に入力する数が結果から引かれることがあります。詳しくは、26 ページの最後の結果の再利用を参照してください。

負数を累乗するときは、かっこで囲みます。たとえば、$(-5)^2$ は 25 になります。-5^2 だと結果は −25 になります。

明示的な乗算と暗黙的な乗算

暗黙的な乗算は、2 つのオペランドが演算子をはさまずに並んでいるときに行われます。たとえば、AB と入力すると、結果は A*B になります。14 Shift x^y ∛ F 8 のような場合も、14 の後に乗算演算子を入力する必要はありません。履歴では内容を分かりやすくするために式に演算子が追加されま

すが、式を入力するときは省略できます。もちろん、演算子を入力してもかまいません。結果は同じです。

大きい結果

行数の多い行列など、結果が長すぎたり高すぎたりして一部が見えなくなっている場合は、その結果を選択して 表示 を押します。これにより、結果が全画面で表示されます。この状態で ▲ および ▼ （または ▶ および ◀ ）を押すと、見えなかった部分を表示できます。 OK をタップすると前の表示に戻ります。

式および結果の再利用

パラメーターが多少違うだけのほぼ同じ式を繰り返し入力するときは、入力した式を取得して再利用すると便利です。履歴に保存されている式はいずれも取得して再利用できます。さらに、履歴に保存されている結果も取得して再利用することができます。

式を取得して編集中の入力行に挿入するには、以下のいずれかの操作を行います。

- 目的の式を 2 回タップします。
- カーソル キーを使用して目的の式を選択してから、式または コピー をタップします。

結果を取得して入力行に挿入するには、カーソル キーを使用して目的の結果まで移動してから、 コピー をタップします。

取得したい式または結果が表示されていない場合は、目的の項目が表示されるまで ▲ を繰り返し押します。画面をスワイプして履歴をすばやくスクロールすることもできます。

ヒント： Shift ▲ を押すと履歴内の最も古いエントリに直接移動し、 Shift ▼ を押すと最も新しいエントリに直接移動できます。

クリップボードの使用

直前に入力した 4 つまでの式は常にクリップボードに保存され、 Shift Menu Paste を押すことで簡単に取得できます。キーを押すとクリップボードが開き、その中から式をすばやく選択できます。

注記： クリップボードに保存されるのは式だけです。結果は保存されません。また、直前に入力した 4 つまでの式は、履歴をクリアしてもクリップボードに残ります。

最後の結果の再利用

Shift + Ans （Ans）を押すと、最後の解答を取得して別の計算に使用できます。Ans が入力行に表示されます。これは最後の解答の省略表現であり、新しい式の一部にすることができます。計算のその他のコンポーネント（演算子、数値、変数など）を入力して新しい計算を作成できます。

ヒント: 新しい計算の先頭に置くときは Ans を選択する必要はありません。バイナリ演算子キーを押して新しい計算を開始した場合は、Ans が新しい計算の最初の要素として入力行に自動的に追加されます。たとえば、最後の解答に 13 をかける場合は、Shift + Ans × 13 Enter と入力できます。ただし、最初の 2 つのキー操作は必要ありません。必ず入力する必要があるのは × 13 Enter のみです。

履歴に保存される結果の精度は常に[番号書式]の設定（19 ページのページ 1 を参照）に従いますが、変数 Ans は常に完全精度で保存されます。つまり、Ans に割り当てられた結果を取得すると、完全精度の数値が得られます。一方、履歴から結果を取得すると、表示されている精度の数値が得られます。

直前の計算を繰り返したいときは、Enter を押します。これは、直前の結果に Ans が含まれている場合に便利です。たとえば、2 の n 乗根の計算で n を 2、4、8、16、32 と増やしていく場合は以下のように操作します。

1. 2 の平方根を計算します。

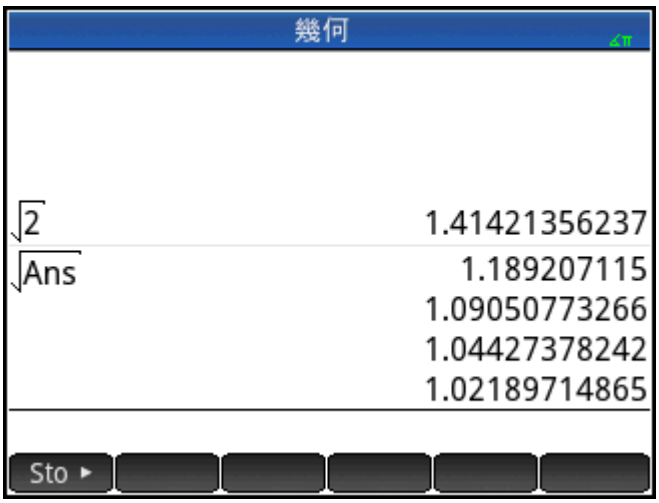
   Shift x^2 2 Enter

2. √Ans を入力します。

   Shift x^2 Shift + Ans Enter

   これは、2 の 4 乗根の計算になります。

3. Enter を繰り返し押します。押すたびに、乗根が直前の乗根の 2 倍になります。下の図で最後に表示されているのは $\sqrt[32]{2}$ の結果です。



CAS の式または結果の再利用

ホーム表示での作業中に CAS の式または結果を取得できます。取得するには、Menu Paste をタップして、**[CAS から取得]**を選択します。CAS が開きます。▲ または ▼ を押して目的の項目まで移動し、Enter を押します。選択した項目がホーム表示のカーソル位置にコピーされます。

変数への値の保存

値は変数に保存できます（つまり、変数に値を割り当てることができます）。計算でその値を使用する場合は、変数の名前でその値を参照できます。独自の変数を作成することも、表示の組み込み変数（ホーム表示では A ~ Z および θ、CAS 表示では a ~ z およびその他）を使用することもできます。CAS 変数はホーム表示での計算に使用できます。ホーム変数は CAS での計算に使用できます。組み込みのアプリケーション変数および幾何変数もあります。これらも計算に使用できます。

[例]：変数 A に π^2 を割り当てるには、次のように入力します。

Shift 3 x^2 Sto ▸ ALPHA Vars Enter

保存された値は下の図のように表示されます。保存した値に 5 をかける場合は、次のように入力します：ALPHA Vars × 5 Enter

ホーム表示で独自の変数を作成することもできます。たとえば、ME という名前の変数を作成して π² を割り当てるには、次のように入力します。

Shift 3 x^2 Sto ▸ ALPHA +/− ALPHA a b/c Enter

ME という名前の変数を作成するかどうかを尋ねるメッセージが表示されます。OK または Enter を押して操作を確定します。以降の計算ではこの変数を使用できます。たとえば、ME*3 と入力すると結果は 29.6088132033 になります。

CAS 表示でも同じ方法で変数を作成できます。組み込みの CAS 変数は小文字で入力する必要がありますが、独自に作成する変数は大文字と小文字のどちらでもかまいません。

組み込みのホーム変数および CAS 変数や独自に作成した変数に加えて、各アプリケーションに用意されている変数も計算に使用できます。

複素数

複素数を使用した算術演算を実行できます。複素数はテキストブック モードで次の形式で入力できます。*x* が実部、*y* が虚部、*i* が虚定数√-1 を示します。

- *(x, y)*
- *x* + *yi*（RPN モードを除く）
- *x* − *yi*（RPN モードを除く）
- *x* + *iy*（RPN モードを除く）
- *x* − *iy*（RPN モードを除く）

RPN モードでは、複素数は一重引用符で囲んで入力する必要があり、明示的な乗算が必要です。たとえば、'3 − 2*i' と入力します。

i を入力するには、以下のように操作します。

▲ ALPHA Shift TAN を押します。

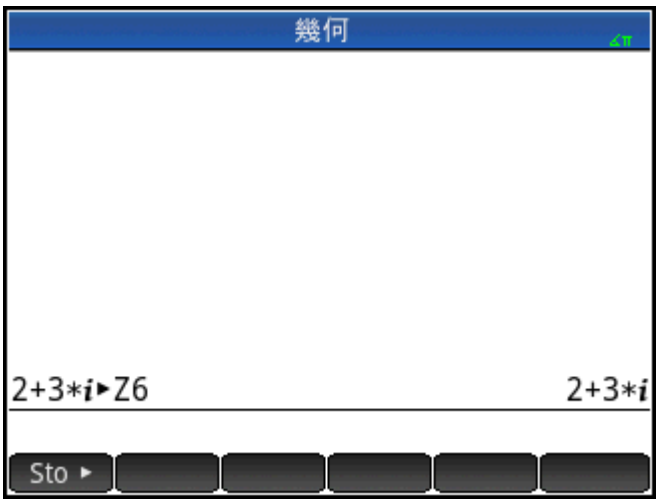
または

Shift 2 を押します。

複素数を保存できる組み込み変数は 10 個あります。それらは Z0 ~ Z9 の名前が付いています。独自に作成した変数に複素数を割り当てることもできます。

複素数を変数に保存するには、複素数を入力し、Sto ▸ を押して、保存先の変数を入力してから、Enter を押します。たとえば、2 + 3*i* を変数 Z6 に保存するには、次のように入力します。

( ) 2 Eval 3 ▶ Sto ▸ ALPHA 2 6 Enter



コピーおよび貼り付け

Shift View Copy は、選択した項目を HP Prime クリップボードにコピーします。Shift Menu Paste は、クリップボードを開き、クリップボードの項目を選択して現在のカーソル位置にその項目を貼り付けることができます。

[リスト]エディターでは、リストの一部またはリスト全体を選択したり、複数のリストの要素を矩形選択したりできます。この選択した要素をコピーして、[行列]エディターに貼り付けたり、[スプレッドシート]アプリケーション、[1 変数統計]アプリケーション、[2 変数統計]アプリケーションの[数値表示]に貼り付けたりすることができます。同様に、[行列]エディターでは、1 つ以上の行、1 つ以上の列、一部の行列、または行列全体を選択できます。この選択した要素をコピーして、[リスト]エディターで貼り付けたり、前述の 3 つのアプリケーションの[数値表示]に貼り付けたりすることができます。

たとえば次の図では、[行列]エディター内の 2×2 の配列が選択されて、クリップボードにコピーされます。

次の図では、この配列が[1 変数統計]アプリケーションの[数値表示]にグリッド データとして貼り付けられています。

次の図では、この配列が[1 変数統計]アプリケーションの[数値表示]に貼り付けられました。

通常は、コピー アンド ペースト機能を使用して、電卓内のソフトウェア間で数値や式を転送できます。

前の例を続行する場合は、 計算 をタップして、列 D1 の 2 つの標本のサマリー統計を計算します。サンプルの標準偏差をタップし、 Shift View Copy を押してこれをクリップボードにコピーします。 Settings を押してホーム表示に移動し、 Shift Menu Paste を押して、サンプルの標準偏差をコマンド行にコピーします。 $\sqrt{x^2}$ を押してこれを二乗し、 Enter ≈ を押して結果を表示します。

この同じコピー アンド ペーストの手法を使用することで、値をコピーして[プロット設定]表示の[Xmin]および[Xtick]ボックスに貼り付けるといった他の操作を実行することができます。

データの共有

HP Prime 電卓では、幅広い種類の計算機能を利用する以外に、独自のオブジェクトを作成して保存し、再利用することもできます。たとえば、アプリケーション、リスト、行列、プログラム、注記を作成できます。また、作成したオブジェクトを他の HP Prime 電卓に送信することもできます。メニュー項目として 送信 が表示される画面では、画面内の項目を選択して他の HP Prime 電卓に送信できます。

HP Prime 間でオブジェクトを送信するときは、付属の USB ケーブルを使用します。これは、micro-A – micro – B USB ケーブルです。USB ケーブルの両端のコネクタは若干異なります。micro-A コネクタの端子は長方形で、micro-B コネクタの端子は台形です。オブジェクトを別の HP Prime と共有するには、micro-A コネクタを送信側電卓の USB ポートに挿入し、micro-B コネクタを受信側電卓の USB ポートに挿入します。

基本手順

オブジェクトを共有するための基本手順は以下のとおりです。

1. 送信するオブジェクトが表示される画面に移動します。

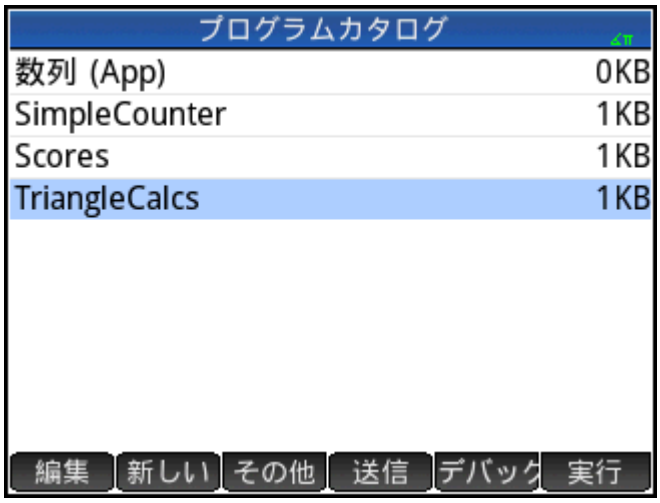
   アプリケーションを送信する場合はアプリケーション ライブラリ、リストを送信する場合は List（リスト）カタログ、行列を送信する場合は Matrix（行列）カタログ、プログラムを送信する場合は Program Catalog（プログラム カタログ）、注記を送信する場合は Note（注記）カタログに移動します。

2. 2 つの電卓を USB ケーブルで接続します。

   送信側電卓の USB ポートに micro-A コネクタ（長方形の端子）を差し込みます。

3. 送信側電卓で、送信するオブジェクトを選択して、 送信 をタップします。

次の図は、Program Catalog（プログラム カタログ）で**[TriangleCalcs]**という名前のプログラムを選択したところです。 送信 をタップすると、接続先の電卓に送信されます。



メモリ管理者の使用

メモリ管理者には、カタログ、ホームと CAS の履歴、ユーザー変数、およびバックアップのリストが含まれます。

▲ メモリ管理者を開くには、 Shift Mem を押します。

メモリ管理者を使用するには、以下の操作を行います。

▲ 以下のメニュー ボタンのどれかを選択します。

- 情報 ：利用できるメモリおよびストレージ領域が表示されます。
- 複製 ：HP Prime 電卓を接続されている HP Prime 電卓に複製します。
- 送信 ：選択したカテゴリのすべてのデータ（リストや行列など）を接続されている HP Prime 電卓に送信します。
- 表示 ：選択されているカタログを開きます。 Enter ≈ を押してカタログを開くこともできます。カタログでは、不要なオブジェクトを削除できます。

バックアップ カタログ

バックアップ カタログを使用すると、コンピューターに接続しないで、お使いの HP Prime 電卓をバックアップまたは復元できます。

バックアップ カタログを開くには、以下の操作を行います。

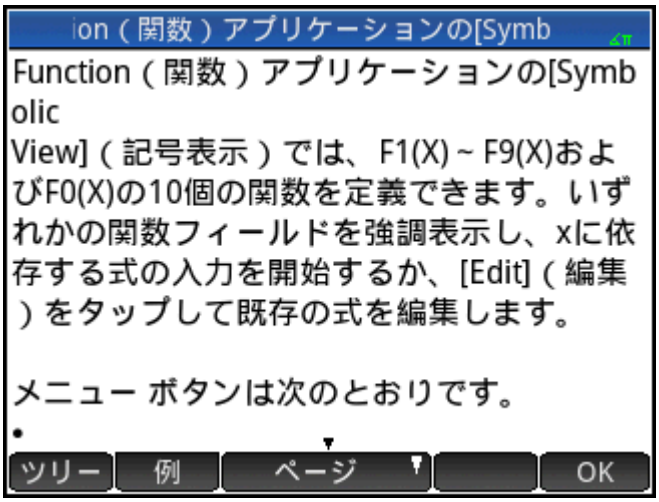
1. メモリ管理者を開きます。
2. **[バックアップ]**→ 表示 の順にタップします。

以下のオプションを使用できます。

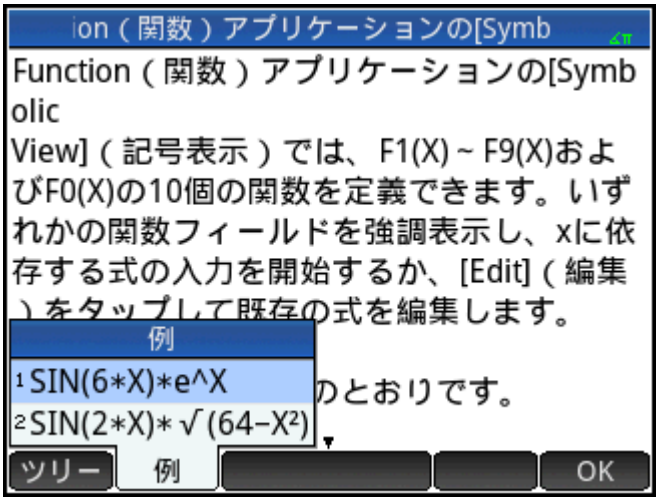
- 復元 : 選択されたバックアップ ファイルを使用して HP Prime 電卓を復元します。
- 削除 :選択されたバックアップ ファイルを削除します。 Del を押して選択されたバックアップ ファイルを削除することもできます。
- 新しい : HP Prime 電卓の現在の状態を使用して、新しいバックアップ ファイルを作成します。初期設定では、バックアップ ファイルの名前には日付が含まれます。

オンライン ヘルプ

HP Prime 電卓には、コンテキスト依存の詳細なオンライン ヘルプ システムが用意されています。通常、アプリケーション、アプリケーションの表示、専用エディター（[リスト]エディターや[行列]エディターなど）、関数、コマンドごとにコンテキスト依存ヘルプを参照できます。現在のコンテキストに関するオンライン ヘルプを開くには、 Help User を押します。たとえば、[関数]アプリケーションの[記号表示]を開いているときに Help User を押すと、次のヘルプ ページが表示されます。

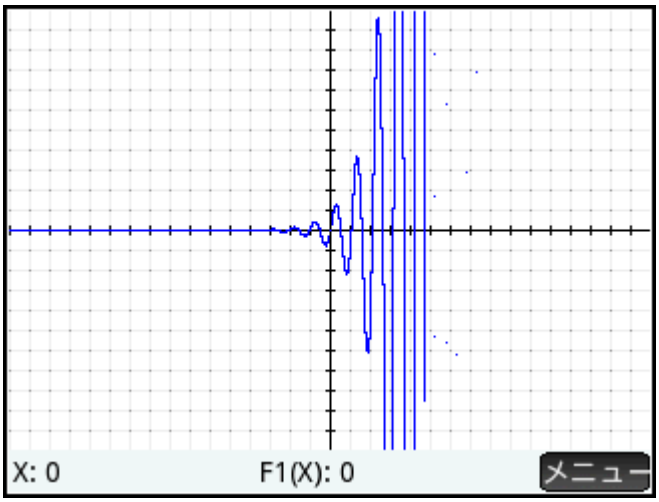


メニュー ページの多くでは、 例 メニュー キーを利用できます。このキーをタップすると、現在のカーソル位置に例を貼り付けることができます。たとえば、 例 をタップしてから、リストの先頭にある次の式をタップします：SIN(6*X)*e^X。



[関数]アプリケーションの[記号表示]のコマンド行にこの関数が貼り付けられます。この関数を F1(X) に貼り付ける場合は、 Enter ≈ を押します。

Plot Setup を押すと、そのグラフが表示されます。



ヘルプ ページを表示しているときに ツリー をタップすると、ヘルプ システム全体の階層ツリーを表示できます。項目をタップしてから OK をタップすると、そのページが表示されます。**[+]**記号をタップすると、その項目が展開されてサブ項目が表示されます。 キー をタップしてからどれかのキーを押す（またはシフト キーと組み合わせて押す）と、そのキーのヘルプが表示されます。

各コマンドの詳細なヘルプも利用できます。このヘルプでは、各コマンドの構文、そのコマンドの説明、および例を参照できます。コマンドを入力したものの、構文が必要になった場合は、 Help User を押すと構文が表示されます。たとえば、CAS ビューで `int( )`と入力した場合、 Help User を押すと、積分コマンドのヘルプが表示されます。

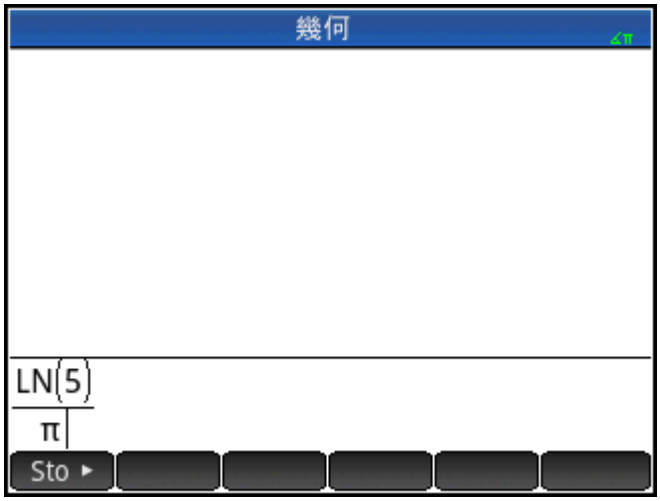
さらに、オンライン ヘルプを開いているときに、 検索 をタップしてキーワードを入力すると、ヘルプ内でそのキーワードを検索できます。

3 逆ポーランド記法（RPN）

HP Prime 電卓のホーム表示では、以下の 3 つの方法でオブジェクトを入力できます。

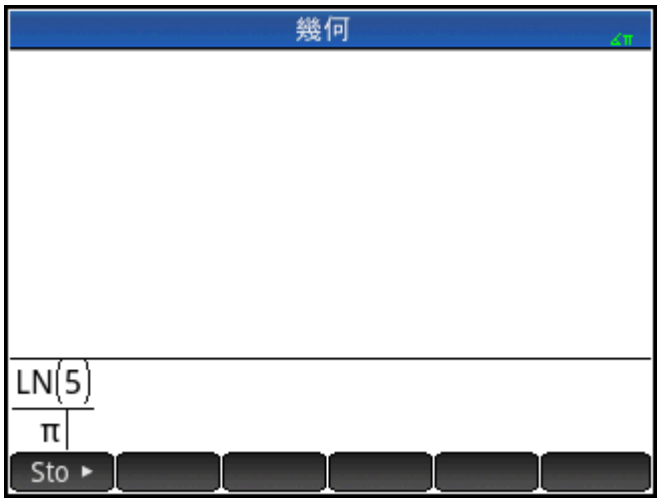
- テキストブック

  引数を上下に置くなど、紙に書くときとほぼ同じ方法で式を入力します。つまり、次の例のように入力内容を 2 次元で表現できます。



- 代数

  式を 1 行で入力します。入力内容は常に 1 次元になります。上記の例の計算は、代数入力モードでは次のようになります。



- 逆ポーランド記法（RPN）

  先に式の引数を入力してから、演算子を入力します。演算子を入力すると、これまでに入力した内容が自動的に評価されます。そのため、演算子が 2 つ含まれる式の場合は、各演算子の式を 2 段階で入力する必要があります。上記の例の計算は次のようになります。

ステップ 1：5 [LN] を入力します。5 の自然対数が計算され、履歴に表示されます。

ステップ 2：[Shift] [3] [÷] を入力します。π が前の結果に除数として適用されます。

入力モードは、**[ホーム設定]**画面（ [Shift] [Settings] ）の 1 ページ目で選択できます。通常の方法で選択できます。

RPN はホーム表示でのみ使用できます。CAS 表示では使用できません。

RPN モードでは、代数モードやテキストブック モードと同じ入力行編集ツールを利用できます。次のキーを使用して、入力行の式を編集できます。

- [Del] を押すと、カーソルの左にある文字が削除されます。
- [Shift] [Del] を押すと、カーソルの右にある文字が削除されます。
- [Shift] [Esc Clear] を押すと、入力行全体がクリアされます。

入力行に式がない場合、 [Shift] [Esc Clear] を押して履歴全体をクリアできます。

RPN モードの履歴

計算結果は履歴に保存されます。この履歴は入力行の上に表示されます（画面に入りきらなくなったときはスクロールします）。履歴には 3 種類あります。1 つは CAS 表示の履歴、あとの 2 つはホーム表示の履歴です。ホーム表示の履歴は次の 2 種類です。

- 非 RPN：入力モードとして代数またはテキストブックを選択した場合に表示されます。
- RPN：入力モードとして RPN を選択した場合にのみ表示されます。RPN 履歴はスタックとも呼ばれます。次の図のように、スタック内の各項目には番号が付けられます。これをスタック レベル番号といいます。

計算を追加するたびに、各項目のスタック レベル番号が大きくなっていきます。

RPN から代数またはテキストブックに切り替えても履歴は消去されません。表示されないだけです。RPN に戻ると、RPN 履歴が再度表示されます。同様に、RPN に切り替えても非 RPN 履歴は消去されません。

RPN モード以外では、履歴は古い順に並べられます。つまり、最も古い計算が一番上、最も新しい計算が一番下に表示されます。RPN モードでは、履歴は初期設定で古い順に並べられますが、履歴内で項目の順序を変えることができます。詳しくは、41 ページのスタックの操作を参照してください。

結果の再利用

履歴内の結果を再利用する方法は 2 つあります。方法 1 では、コピーした結果がコピー後に選択解除されます。方法 2 では、コピーした結果が選択されたままになります。

方法 1

1. コピーする結果を選択します。選択するには、▲ または ▼ を押して目的の結果まで移動するか、目的の結果をタップします。
2. Enter を押します。結果が入力行にコピーされ、選択が解除されます。

方法 2

1. コピーする結果を選択します。選択するには、上矢印または下矢印を押して目的の結果まで移動するか、目的の結果をタップします。
2. スタックをタップして、**[エコー]**を選択します。結果が入力行にコピーされ、選択されたままになります。

CAS 履歴の項目をコピーしてホーム表示で計算に使用したり、ホーム履歴の項目をコピーして CAS で計算に使用したりすることはできますが、RPN 履歴の項目はこのようなやりとりはできません。ただし、RPN モードで CAS のコマンドや関数を使用することはできます。

計算例

RPN の基本は、演算子の前に引数を置くことです。引数は入力行に入力することも（スペース区切り）、履歴から入力することもできます。たとえば、π に 3 を掛けるには、入力行に次のように入力します。

Shift 3 _ 3

その後、演算子 × を入力します。演算子を入力する前の入力行には次のように表示されます。

または、引数を個別に入力した後、何も表示されていない入力行に演算子 [×] を入力することもできます。その場合、演算子を入力する前の履歴には次の図のように表示されます。

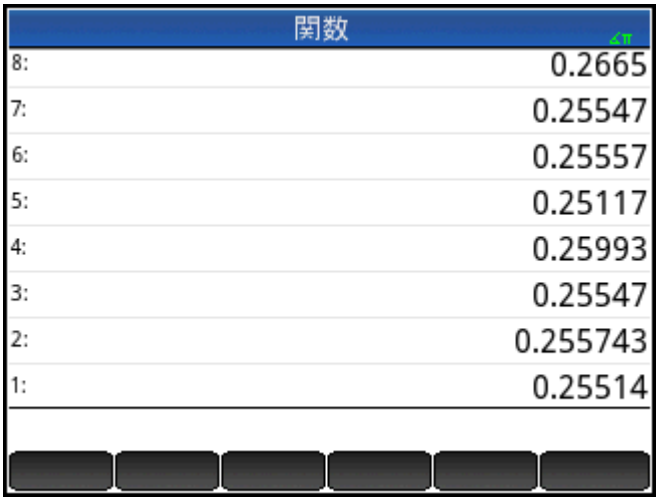
同じ結果を得るために、[Shift] [3 π] [Enter ≈] を押して π の値をスタック レベル 1 に入力し、その後で [3 π] [×] を押すこともできます。

履歴に項目がない状態で演算子や関数を入力すると、エラー メッセージが表示されます。また、演算子の対象となるスタック レベル項目がその演算子の引数として適切でない場合にもエラー メッセージが表示されます。たとえば、レベル 1 に文字列があるときに [COS ACOS H] を押すとエラー メッセージが表示されます。

演算子や関数では、結果を出すために必要最小限の引数のみが使用されます。たとえば、入力行に 2 4 6 8 と入力して [×] を押すと、スタック レベル 1 に「48」と表示されます。これは、乗算で必要な引数は 2 つだけであるため、後に入力した 2 つの引数のみが計算に使用された結果です。先に入力した 2 と 4 は無視されます。2 はスタック レベル 3 に置かれ、4 はスタック レベル 2 に置かれます。

受け付ける引数の個数が決まっていない関数の場合は、演算に含める引数の個数を指定する必要があります。個数を指定するには、関数名の直後に個数をかっこで囲んで指定します。その後

Enter ≈ を押すと、関数が評価されます。たとえば、スタックが次のようになっているとします。



このとき、スタック レベル 1、2、3 の数の中だけで最小値を調べる場合は、**[数学]**メニューから**[MIN]**を選択し、個数を入力して**[MIN(3)]**とします。 Enter ≈ を押すと、スタックの下 3 項目の中で最小の値が表示されます。

スタックの操作

いくつかのオプションを使用してスタックを操作できます。ほとんどのオプションは画面下部にメニュー項目として表示されます。これらの項目を表示するには、最初に履歴項目を選択する必要があります。

PICK（ピック）

選択した項目をスタック レベル 1 にコピーします。コピー後は 1 つ下の項目が選択されます。つまり、 PICK を 4 回タップした場合は、4 つの連続する項目が下から 4 つのスタック レベル（レベル 1 ~ 4）に移動します。

ROLL（ロール）

[ROLL]コマンドは 2 種類あります。

- ROLL↑ をタップすると、選択した項目がスタック レベル 1 に移動します。これは[PICK]と似ていますが、[PICK]では項目がコピーされ、そのコピーがスタック レベル 1 に置かれます。一方、[ROLL]では項目はコピーされません。移動されるだけです。
- ROLL↓ をタップすると、スタック レベル 1 の項目が、現在選択されているレベルに移動します。

Swap（スワップ）

スタック レベル 1 のオブジェクトとスタック レベル 2 のオブジェクトを入れ替えます。 を押すだけです。他のレベルのオブジェクトは変更されません。入力行がアクティブなときはこの操作は機能しません。代わりにカンマが入力されます。

Stack（スタック）

スタックをタップすると、その他のスタック操作ツールが表示されます。

DROPN（以下をドロップ）

選択した項目からスタック レベル 1 の項目までをすべて削除します。選択した項目より上にある項目は下の空いたレベルに移動します。

特定の項目をスタックから削除する方法については、43 ページの特定の項目の削除を参照してください。

DUPN（以下を複製）

選択した項目からスタック レベル 1 の項目までをすべて複製します。たとえば、スタック レベル 3 の項目を選択して**[DUPN]**を選択すると、その項目とその下 2 つの項目が複製されてスタック レベル 1 ~ 3 に置かれ、元の 3 項目がスタック レベル 4 ~ 6 に移動されます。

Echo（エコー）

選択した項目のコピーを入力行に置き、元の項目を選択状態のままにします。

→LIST（→リスト）

選択した項目を先頭とし、スタック レベル 1 の項目を末尾とする項目リストを作成します。

図 3-1 実行前

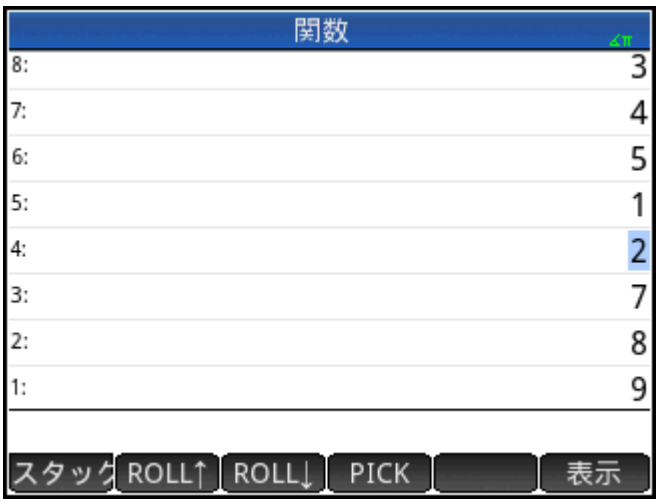
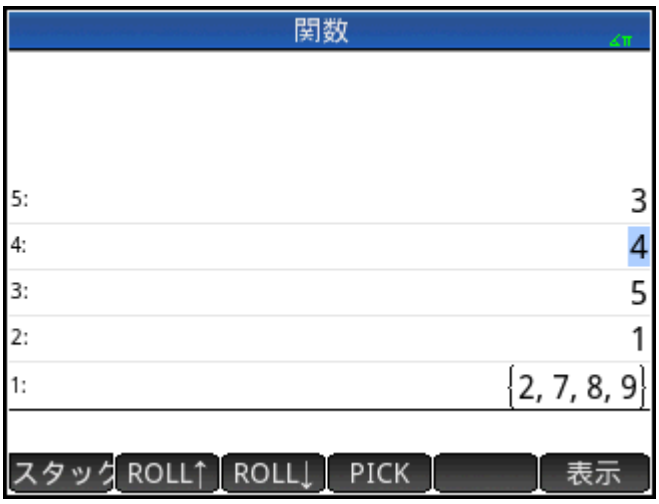


図 3-2 実行後



項目の表示

項目をテキストブック形式で全画面表示するときは、 表示 をタップします。

履歴に戻るときは OK をタップします。

特定の項目の削除

スタックから特定の項目を削除するには、以下の操作を行います。

1. 目的の項目を選択します。選択するには、 ▲ または ▼ を押して目的の項目まで移動するか、目的の項目をタップします。
2. Del を押します。

全項目の削除

すべての項目を削除して履歴を消去するには、

を押します。

4 数式処理システム（CAS）

数式処理システム（CAS）を使用すると、記号計算を実行できます。初期設定では、CAS は正確モードで動作し、無限精度で計算ができます。一方、CAS 以外の計算（ホーム表示またはアプリケーションで実行される計算など）は数値計算であり、多くの場合、電卓の精度（HP Prime の場合は 12 桁）で制限される近似値となります。たとえば、1/3+2/7 はホーム表示（標準数値形式）では近似値の解答.619047619047 となりますが、CAS では正確な解答 13/21 となります。

CAS には、数百の関数（代数、微積分、方程式の解法、多項式など）が用意されています。関数はツールボックス メニューの 1 つである**[CAS]**メニューから選択します。CAS コマンドについて詳しくは、「*関数およびコマンド*」の章の「*[CAS]メニュー*」を参照してください。

CAS 表示

CAS 計算は CAS 表示で行います。CAS 表示はホーム表示とほとんど同じです。計算の履歴が作成され、ホーム表示でできるような以前の計算の選択およびコピーを行ったり、変数内のオブジェクトを保管したりできます。

CAS 表示を開くには、 CAS Settings を押します。ホーム表示でなく CAS 表示が表示されていることを示す**[CAS]**というラベルがタイトル バーの左側に白色で表示されます。

CAS 表示のメニュー ボタンは次のとおりです。

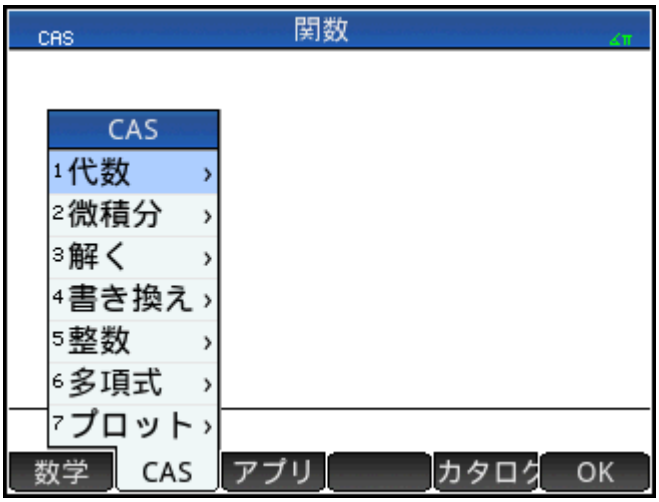
- Sto ▸ ：オブジェクトを変数に割り当てます。
- simplify ：一般的な簡約規則を適用して、式を最も簡単な形式に短縮します。たとえば、simplify($e^{a + \mathrm{LN}(b*e^{c})}$) は b*EXP(a)*EXP(c) になります。
- コピー ：履歴から選択された項目を入力行にコピーします。
- 表示 ：選択した項目を、水平方向および垂直方向のスクロールが有効な全画面表示モードで表示します。項目はテキストブック形式で表示されます。

CAS 計算

CAS 表示ではホーム表示とほぼ同じように計算ができます。CAS 表示には RPN 入力モードがなく、代数モードおよびテキストブック モードしかない点だけが異なります。演算子キーおよび関数キーもすべてホーム表示と同じように機能します（ただし、英字はすべて大文字ではなく小文字で表示されます）。主な違いは、初期設定で解答が数値ではなく記号で表示されることです。

テンプレート キー（ ）を使用して、基本的な計算およびベクトルや行列のフレームワークを簡単に挿入することもできます。

一般的な CAS 関数は、[CAS]メニューから利用できます。メニューを表示するには、 ボタンを押します。初期設定で[CAS]メニューが表示されない場合は、 CAS をタップしてください。その他の CAS コマンドは、ツールボックス メニューの[カタログ]メニューから利用できます。



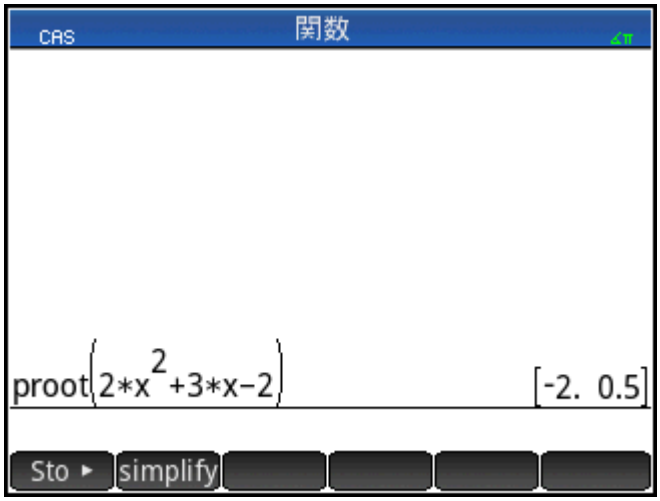
関数を選択するには、カテゴリを選択してから、コマンドを選択します。

例 1

$2x^2 + 3x - 2$ の根を求めるには、以下の操作を行います。

1. [CAS]メニューを開いて、**[多項式]**、**[根を求める]**の順に選択します。

   関数 `proot()` が入力行に表示されます。



2. かっこ内に 2 [ALPHA] [×] [x^2] [+] 3 [ALPHA] [×] [−] 2 と入力します。

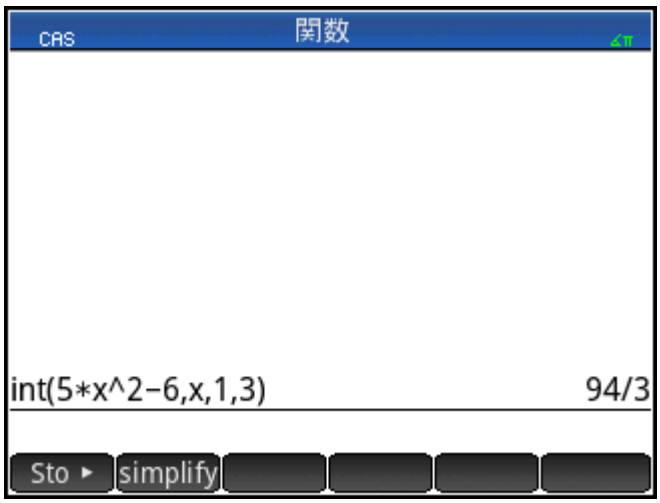
3. [Enter] を押します。

例 2

$5x^2 - 6$ のグラフの x =1 から x = 3 までの範囲と x 軸で囲まれた領域の面積を求めるには、以下の操作を行います。

1. [CAS]メニューを開いて、**[微積分]**、**[積分]**の順に選択します。

   関数 `int()` が入力行に表示されます。

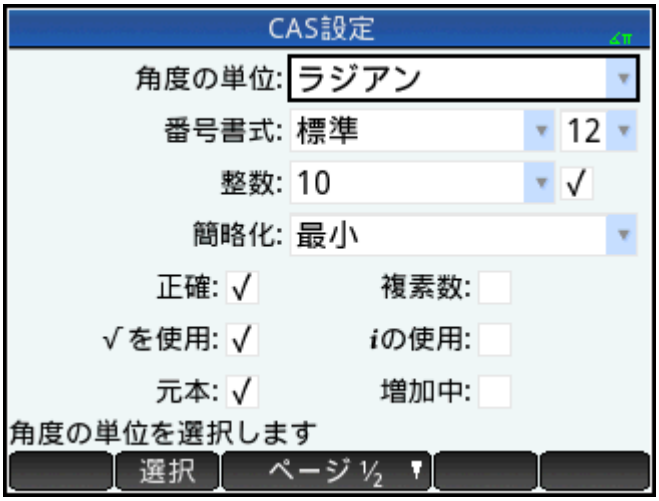


2. かっこ内に 5 [ALPHA] [×] [x^2] [−] 6 [,] [ALPHA] [×] [,] 1 [,] 3 と入力します。

3. [Enter] を押します。

設定

さまざまな設定を使用して、CAS の動作を構成できます。設定を表示するには、 Shift CAS Settings を押します。各種のモードが 2 ページにわたって表示されます。



ページ 1

設定	用途
[角度の単位]	角度の単位を選択します。**[ラジアン]**または**[度]**を選択できます
[番号書式]の 1 つ目のドロップダウン リスト	表示される解の数値形式を選択します。**[標準]**、**[関数]**、または**[工学]**を選択できます
[番号書式]の 2 つ目のドロップダウン リスト	近似モードで表示する桁数（仮数部+指数）を選択します
[整数]ドロップダウン リスト	整数の基数を選択します **[10]**：10 進数 **[16]**：16 進数 **[8]**：8 進数
[整数]チェック ボックス	チェックを入れると、CAS 以外の環境で整数に相当する実数が CAS で整数に変換されます。整数に相当しない実数は、このオプションの選択に関係なく、CAS でも実数として扱われます
[簡略化]	自動簡略化のレベルを選択します **[なし]**：自動簡略化をしません（手動で簡略化するときは simplify を使用します） **[最小]**：基本的な簡略化を行います（初期設定） **[最大]**：常に簡略化を試みます
[正確]	チェックを入れると、電卓が正確モードになり、解が記号で表現されます。チェックを外すと、電卓が近似モードになり、解が近似値で表現されます。たとえば、26 ÷ 5 の解は、正確モードでは 26/5、近似モードでは 5.2 になります

設定	用途
[複素数]	変数の結果に複素数を許可する場合に選択します
[√を使用]	チェックを入れると、複素数モードまたは実数モードで 2 次多項式の判別式が正の場合に式が因数分解されます
[i を使用]	チェックを入れると、電卓が複素数モードになり、複素数解がある場合は複素数解が表示されます。チェックを外すと、電卓が実数モードになり、実数解のみが表示されます。たとえば、factors(x^4-1)の解は、複素数モードでは$(x-1),(x+1),(x+i),(x-i)$、実数モードでは$(x-1),(x+1),(x^2+1)$になります
[主要解]	チェックを入れると、三角関数の主要解が表示されます。チェックを外すと、三角関数の一般解が表示されます
[増加中]	チェックを入れると、多項式が昇べきの順で表示されます（例：$-4+x+3x^2+x^3$）。チェックを外すと、多項式が降べきの順で表示されます（例：x^3+3x^2+x-4）

ページ 2

設定	用途
[再帰的評価]	対話型評価で許可される組み込み変数の最大数を指定します。「再帰的置換」も参照してください
[再帰的置換]	プログラムの単一評価で許可される組み込み変数の最大数を指定します。「Recursive Evaluation」も参照してください
[再帰的関数]	組み込み関数を呼び出すことができる最大回数を指定します
[イプシロン]	イプシロンに指定した値よりも小さい数値はすべてゼロとして扱われます
[確率]	非決定性アルゴリズムで解答が誤りとなる最大の確率を指定します。決定性アルゴリズムの場合はこれをゼロに設定します
[ニュートン]	ニュートン法を使用して 2 次方程式の根を求めるときの最大反復回数を指定します

メニュー項目の形式の設定

[CAS 設定]画面以外にも CAS に関する設定が 1 つあります。この設定では、[CAS]メニューのコマンドを説明的な名前で表示するかコマンド名で表示するかを選択できます。次の表に、表示モードによる同じ関数の異なる名前の例をいくつか示します。

説明的な名前	コマンド名
因数リスト	ifactors
複素零点	cZeros
グレブナー基底	gbasis
次数による因数分解	factor_xn
根を求める	proot

初期設定のメニュー表示モードでは、CAS 関数の説明的な名前が表示されます。関数をコマンド名で表示するには、**[ホーム設定]**画面の 2 ページ目で**[メニュー表示]**オプションの選択を解除します。

ホーム表示の式または結果の使用

CAS での作業中にホーム表示の式または結果を取得できます。取得するには、 Menu Paste をタップして、**[ホームから取得]**を選択します。ホーム表示が開きます。 ▲ または ▼ を押して目的の項目まで移動し、 Enter ≈ を押します。選択した項目が CAS のカーソル位置にコピーされます。コピー（ Shift View Copy ）してから貼り付ける（ Shift Menu Paste ）こともできます。

CAS でのホーム変数の使用

CAS 内からホーム変数にアクセスできます。ホーム変数には大文字が割り当てられ、CAS 変数には小文字が割り当てられます。このため、SIN(x)と SIN(X)では結果が異なります。

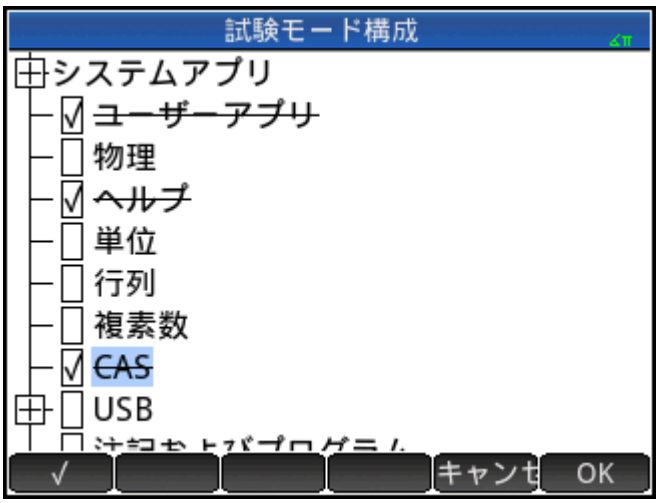
CAS でホーム変数を使用するには、計算の中にその名前を含めます。たとえば、ホーム表示で変数 Q に 100 を代入し、CAS で変数 q に 1000 を代入したとします。CAS で 5*q と入力した場合の結果は 5000 です。5*Q と入力した場合の結果は 500 です。

同様に、ホーム表示で計算に CAS 変数を使用することもできます。その場合は、ホーム表示で 5*q と入力すると、q は CAS 変数なので結果は 5000 になります。

5 試験モード

HP Prime 電卓では、特定の機能を一定期間だけ無効にして、試験での使用に適した構成にすることができます。このような試験用の HP Prime 電卓構成を試験モード構成といいます。複数の試験モード構成を作成し、異なる機能を無効にして保存することもできます。各構成には期間およびパスワードの有無を個別に設定できます。試験モード構成は、HP Prime 電卓で直接有効にすることも、別の HP Prime 電卓から USB ケーブル経由で送信することも、または HP 接続キットを使用して 1 台以上の HP Prime 電卓に送信することもできます。

試験モード構成は、試験を受ける学生に電卓を適切に使用させたい教員や試験監督を主な対象としています。次の図では、ユーザーがカスタマイズしたアプリケーション、ヘルプ システム、および CAS が無効になっています。

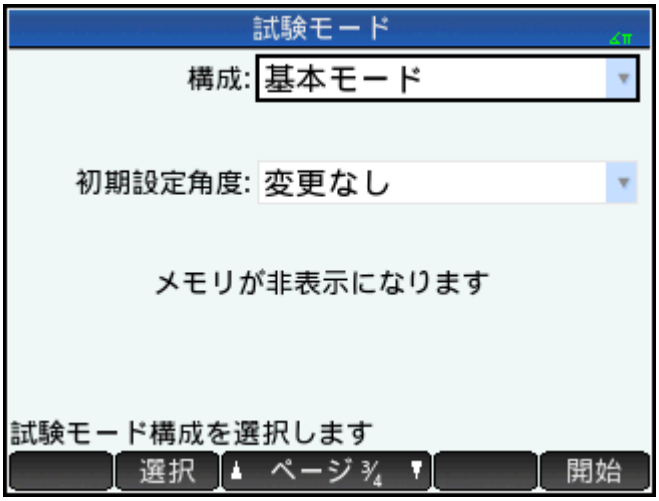


試験モード構成では、試験モード中に電卓の 3 つのランプを定期的に点滅させることができます。ランプは電卓の上端にあります。ランプを点滅させることで、試験中に試験モードが解除されていないかどうかを監督が確認できます。ランプは試験モードのすべての電卓で同期して、同じタイミング、同じパターンで点滅します。

基本モードの使用

試験モード表示に初めてアクセスした場合、初期設定では[構成]フィールドは[基本モード]と表示されます。ユーザーは基本モードを変更できません。独自の試験モード構成を定義するには、構成を**[初期設定試験]**に変更してください。独自の構成の設計について詳しくは、52 ページの初期設定構成の変更を参照してください。基本モードでは、以下の設定が構成されます。

- 試験モードが有効になっている間、HP Prime 電卓のメモリは非表示になります。
- 電卓の上部にある緑色のランプが点滅します。



電卓が基本モードを維持する時間の長さの設定に制限はありません。このモードを終了するには、付属の microUSB ケーブルを使用して、この電卓をコンピューターまたは別の HP Prime 電卓に接続します。

初期設定構成の変更

[構成]ボックスで**[初期設定試験]**を選択した後、独自の試験モード構成を定義できます。必要な構成が 1 つだけの場合は、この初期設定試験構成を変更するだけです。試験ごとに構成を変えたい場合など、複数の構成が必要な場合は、最もよく使用する設定に合わせて初期設定構成を変更し、それ以外の設定用の構成を追加で作成します。初期設定試験を構成して有効にするための画面にアクセスする方法は 2 つあります。

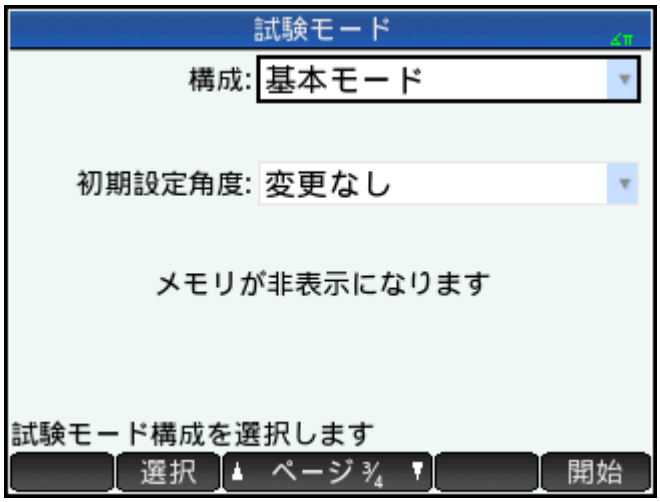
- On + a b/c または On + Esc を押します。
- **[ホーム設定]**画面の 3 ページ目を選択します。

以下の手順では 2 つ目の方法を使用しています。

1. Shift Settings を押します。**[ホーム設定]**画面が表示されます。
2. ページ ¼ の右側をタップします。

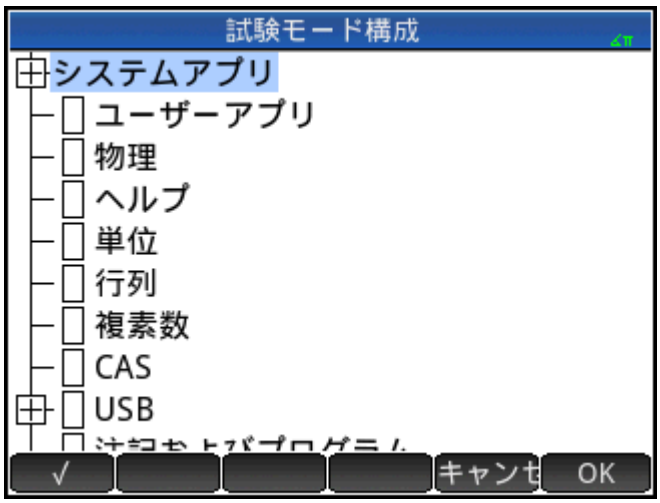
3. ▲ ページ ¾ ▼ の右側をタップします。

   [試験モード]画面が表示されます。特定の構成を有効にするときは（試験開始前など）、この画面を使用します。



4. 選択 をタップして**[初期設定試験]**を選択します。

5. 構成 をタップします。**[試験モード構成]**画面が表示されます。



6. 無効にする機能を選択し、無効にしない機能が選択されていないことを確認します。

   各機能の左側にある展開ボックスは、そのカテゴリにサブ項目が含まれ、それらを個別に無効にできることを示します。上記の図では**[システムアプリケーション]**の横に展開ボックスが表示されています。サブ項目を表示するには展開ボックスをタップします。この状態でサブ項目を個別に選択できます。すべてのサブ項目を無効にする場合はカテゴリを選択します。

   オプションを選択/選択解除するには、チェック ボックスをタップするか、カーソル キーを使用して目的のオプションまでスクロールしてから √ をタップします。

7. 無効にする機能をすべて選択したら、 OK をタップします。

   試験モードをすぐに有効にする場合は、55 ページの試験モードの有効化に進んでください。

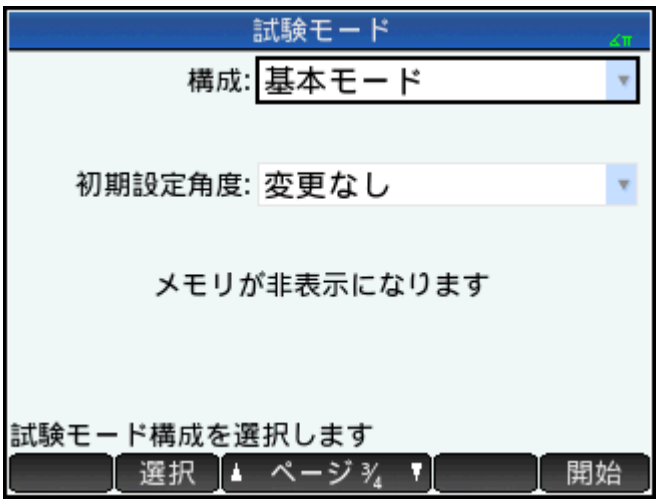
新しい構成の作成

無効にする機能を変更する必要があるときは、初期設定試験構成を変更できます。代わりに、初期設定構成はそのままにして、新しい構成を作成することもできます。新しい構成を作成するときは、そのベースとなる既存の構成を選択します。

ヒント：基本モードを変更することはできません。

1. Shift Settings を押します。**[ホーム設定]**画面が表示されます。
2. ページ 1/4 をタップします。
3. ページ 3/4 をタップします。

   [試験モード]画面が表示されます。



4. [基本モード]以外の基本構成を**[構成]**リストから選択します。新しい試験モード構成を初めて作成するときは、選択できる基本構成は[初期設定試験]のみです。
5. その他 をタップし、メニューから**[コピー]**を選択して、新しい構成の名前を入力します。
6. OK を 2 回タップします。
7. 構成 をタップします。**[試験モード構成]**画面が表示されます。
8. 無効にする機能を選択し、無効にしない機能が選択されていないことを確認します。
9. 無効にする機能をすべて選択したら、 OK をタップします。

   HP 接続キットを使用すると、HP Prime で作業するときとほぼ同じ方法で試験モード構成を作成できます。その後、作成した構成を USB 経由で送信するか無線モジュールでクラスに一斉送信して、複数の HP Prime で有効にできます。詳しくは、製品 CD に付属の HP 接続キットをインストールして起動し、[接続キット]メニューで**[ヘルプ]**をクリックして、**[HP 接続キット ユーザーガイド]**を選択してください。

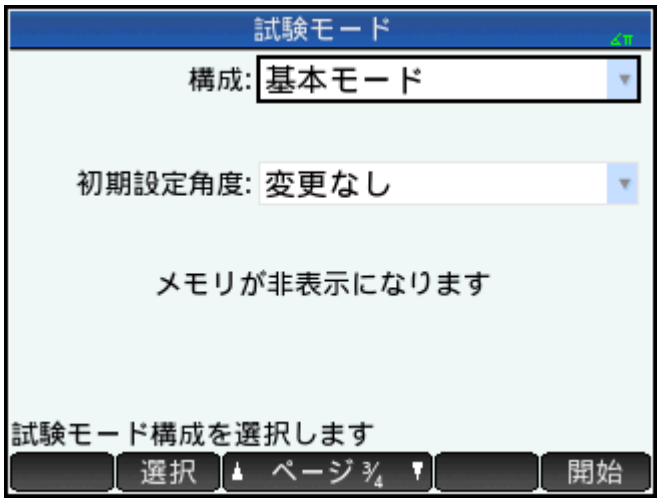
   試験モードをすぐに有効にする場合は、55 ページの試験モードの有効化に進んでください。

試験モードの有効化

試験モードを有効にすると、電卓のユーザーは、無効にした機能にアクセスできなくなります。無効にした機能は、指定した期間が過ぎるか試験モード用パスワードを入力すると再度アクセスできるようになります。

試験モードを有効にするには、以下の操作を行います。

1. **[試験モード]**画面が表示されていない場合は、Shift Settings を押してから、ページ 1/4 、ページ 2/4 の順にタップします。



2. [基本モード]以外の構成を有効にする場合は、**[構成]**リストから目的の構成を選択します。
3. [基本モード]以外の構成を使用している場合、**[タイムアウト]**リストからタイムアウト期間を選択します。

   最長期間は 8 時間です。試験監督の準備をする際は、選択したタイムアウト期間が試験時間よりも長いことを確認してください。
4. [基本モード]以外の構成を使用している場合は、初期設定角度モードを選択するか、初期設定を空のままにします（または、**[変更なし]**を選択します）。
5. [基本モード]以外の構成を使用している場合、1 文字から 10 文字のパスワードを入力します。タイムアウト期間が終了する前に試験モードを解除するときは、このパスワードを入力する必要があります。
6. 以下の電卓メモリ オプションのどれかを選択します。

ヒント：[基本モード]では、試験モードが有効になっている間、電卓メモリは自動的に非表示になります。

- **[Keep]**：プログラムおよび注記を含む、現在の電卓メモリへのフル アクセスを生徒に許可します。
- **[消去]**：電卓メモリを完全に消去します。

注記：この操作は元に戻せません。

- **[非表示]**：試験モードが有効になっている間、電卓メモリを非表示にします。
- **[維持および復元]**：試験モードが有効になっている間、電卓メモリを非表示にします。試験モードを無効にすると、電卓メモリは試験モードを有効にする前の状態に復元されます。

7. 試験モード中に試験モード インジケーターを定期的に点滅させる場合は、**[LED の点滅]**を選択します。基本モードでは、電卓の上部にある緑色のランプが自動的に点滅します。
8. 試験モードのセキュリティを強化したい場合は、**[セキュリティ コード]**を選択して、試験モードを開始するために入力するセキュリティ コードを生徒に渡します。
9. 基本モードを使用する場合、学生用の電卓の 開始 をタップします。あるいは、付属の USB ケーブルを使用して、学生用の電卓を接続します。

   micro-A コネクタ（長方形の方の端）を送信側電卓の USB ポートに接続し、もう一方のコネクタを受信側電卓の USB ポートに接続します。
10. 接続した電卓で構成を有効にするには、 送信 をタップし、以下のオプションのどちらかを選択します。
    - **[送信および開始]**：接続した電卓で試験モードが自動的に開始され、指定した機能が無効になってアクセスできなくなります。
    - **[ファイルの送信]**：電卓を切断して 開始 をタップした後に、接続した電卓で試験モードを開始します。電卓が試験モードになり、指定した機能が無効になってアクセスできなくなります。
11. 機能を制限する必要のあるすべての電卓について手順 9 を繰り返します。

試験モードの解除

指定した期間が終了する前に試験モードを解除する場合は、以下のどちらかの操作を行います。

- 適切なケーブルを使用して、電卓をコンピューターまたは別の HP Prime 電卓に接続します。
- 試験モード構成のパスワードを設定した場合は、以下の手順を使用してパスワードを入力します。

試験モードのパスワードを入力するには、以下の操作を行います。

1. **[試験モード]**画面が表示されていない場合は、 Shift Settings を押してから、 ページ 1/4 、 ページ 2/4 の順にタップします。
2. 現在の試験モードを有効にしたときに指定したパスワードを入力して、 OK を 2 回タップします。

HP 接続キットを使用して試験モードを解除することもできます。詳しくは、『HP 接続キット ユーザー ガイド』を参照してください。

構成の変更

試験モードの構成を変更できます。構成を削除したり、初期設定の構成に戻したりすることもできます。

構成の変更

1. **[試験モード]**画面が表示されていない場合は、Shift Settings を押してから、ページ 1/4 、ページ 2/4 の順にタップします。
2. **[構成]**リストから、変更する構成を選択します。
3. 構成 をタップします。
4. 必要な変更を行ってから、OK をタップします。

初期設定の構成に戻す

1. Shift Settings を押します。**[ホーム設定]**画面が表示されます。
2. ページ 1/4 をタップします。
3. ページ 2/4 をタップします。

   [試験モード]画面が表示されます。
4. **[構成]**リストから**[初期設定試験]**を選択します。
5. その他 をタップして、メニューから**[リセット]**を選択し、OK をタップして構成を初期設定に戻すことを確認します。

構成の削除

1. **[試験モード]**画面が表示されていない場合は、Shift Settings を押してから、ページ 1/4 、ページ 2/4 の順にタップします。
2. **[構成]**リストから、削除する構成を選択します。

   注記：[基本モード]または[初期設定試験]を削除することはできません。
3. その他 をタップしてから、**[削除]**を選択します。
4. 削除の確認を求められたら、OK をタップするか、Enter を押します。

6 HP アプリケーションの基本

HP Prime 電卓の多くの関数は、HP アプリケーションと呼ばれるパッケージで提供されています。HP Prime 電卓には HP アプリケーションが 18 本付属しています。特定の数学的主題または課題専用のアプリケーションが 10 本、専門ソルバー アプリケーションが 3 本、関数エクスプローラーが 3 本、スプレッドシート アプリケーションが 1 本、外部センサー装置から電卓にストリーミングされたデータを記録するためのアプリケーションが 1 本あります。アプリケーションを起動するには、最初に Apps Info を押して[アプリケーション ライブラリ]画面を表示し、目的のアプリケーションのアイコンをタップします。

次の表に、各アプリケーションの簡単な説明を示します。アプリケーションはアルファベット順に並んでいます。

アプリケーション名	説明
高度なグラフ作成	x および y の記号開放文のグラフを操作します 例：$x^2 + y^2 = 64$
データストリーマー	科学的なセンサーから実環境データを収集して、統計分析アプリケーションにエクスポートします
金融	貨幣の時間価値（TVM）問題および償却の問題を解きます
関数	x の実数値関数 y を操作します $y = 2x^2 + 3x + 5$
幾何	幾何学的構造を操作して幾何学的計算を実行します
推論	正規分布およびスチューデントの t 分布に基づく信頼区間と仮説検証を調べます
1 次エクスプローラー	一次方程式の性質を調べて自分の知識を試します
線形ソルバー	2 つまたは 3 つの一次方程式の組み合わせの解を見つけます
パラメトリック	t のパラメトリック関数 x および y を操作します。例：x = cos(t)、y = sin(t)
極座標	角度 θ の極関数 r を操作します 例：$r = 2\cos(4\theta)$
2 次エクスプローラー	二次方程式の性質を調べて自分の知識を試します
数列	n によって U を定義する数列関数、または同じ数列や別の数列の前の項によって U を定義する数列関数（U_{n-1}、U_{n-2} など）を操作します 例：$U_1 = 0$、$U_2 = 1$、$U_n = U_{n-2} + U_{n-1}$
解く	実数値の変数を 1 つ以上含む方程式および連立方程式を操作します 例：$x + 1 = x^2 - x - 2$
スプレッドシート	スプレッドシートに適した問題を解き、データを表示します
1 変数統計	1 変数統計データ（x）を計算します
2 変数統計	2 変数統計データ（x および y）を計算します

アプリケーション名	説明
三角ソルバー	三角形の長さおよび角度の未知値を求めます
三角法エクスプローラー	正弦関数の性質を調べて自分の知識を試します

アプリケーションを使用して調べごとをしたり問題を解いたりしていくと、1 つ以上の表示にデータや定義を追加することになります。これらの情報はすべてアプリケーション内に自動的に保存されます。いったんアプリケーションを離れてから戻ったときも情報は以前のまま保持されています。また、現在の状態のアプリケーションに名前を付けて保存し、初期状態のアプリケーションを別の問題や目的に使用することもできます。アプリケーションのカスタマイズおよび保存について詳しくは、100 ページのアプリケーションの作成を参照してください。

上記のアプリケーションは 1 つを除いて、このユーザー ガイドで詳しく説明します。例外は DataStreamer アプリケーションです。このアプリケーションについては、『*HP Prime グラフ電卓クイックスタート ガイド*』で簡単に紹介しているほか、HP StreamSmart 410 ユーザー ガイドで詳しく説明しています。

アプリケーション ライブラリ

アプリケーションはアプリケーション ライブラリにまとめられています。アプリケーション ライブラリを表示するには、 Apps Info を押します。

アプリケーションを開く

1. アプリケーション ライブラリを開きます。
2. 目的のアプリケーションのアイコンをタップします。

   カーソル キーを使用して目的のアプリケーションを選択し、 開始 をタップするか Enter ≈ を押すこともできます。

アプリケーションのリセット

アプリケーションを離れても、アプリケーション内のデータおよび設定はすべて保持されます。そのアプリケーションに戻ると、以前の状態から作業を再開できます。

以前のデータや設定を使用したくない場合は、アプリケーションを初期状態、つまりそのアプリケーションを初めて開いたときの状態に戻すことができます。

アプリケーションをリセットするには、以下の操作を行います。

1. アプリケーション ライブラリを開きます。
2. カーソル キーを使用してアプリケーションを選択します。
3. リセット をタップします。
4. OK をタップして操作を確定します。

アプリケーション内からリセットすることもできます。アプリケーションのメイン表示（多くの場合は[記号表示]）で、 Shift Esc Clear を押してから、 OK をタップして操作を確定します。

アプリケーションの並べ替え

アプリケーション ライブラリでは、初期設定で組み込みアプリケーションが使用順に並べられます。最後に使用したアプリケーションが先頭に表示されます。カスタマイズしたアプリケーションは常に組み込みアプリケーションの後に表示されます。

組み込みアプリケーションの並び順を以下のように変更できます。

- **[Alphabetically]**（アルファベット順）: アプリケーション名のアルファベット順に並べます。A から Z の順になります。
- **[Fixed]**（固定）: 初期設定の順序に固定します。関数、高度なグラフ作成、Geometry（幾何）… 極座標、数列の順になります。カスタマイズしたアプリケーションは常に組み込みアプリケーションの後に表示されます。これらは作成順に並べられます。古いものから順に表示されます。

並び順を変更するには、以下の操作を行います。

1. アプリケーション ライブラリを開きます。
2. 並替 をタップします。
3. **[アプリの並替]**リストで、目的のオプションを選択します。

アプリケーションの削除

HP Prime 電卓に付属するアプリケーションは組み込み型であるため削除できませんが、ユーザーが作成したアプリケーションは削除できます。

アプリケーションを削除するには、以下の操作を行います。

1. アプリケーション ライブラリを開きます。
2. カーソル キーを使用してアプリケーションを選択します。
3. 削除 をタップします。
4. OK をタップして操作を確定します。

その他のオプション

上記のほかに、アプリケーション ライブラリでは以下のオプションを使用できます。

- 保存：アプリケーションのコピーを新しい名前で保存します。100 ページのアプリケーションの作成を参照してください。
- 送信：アプリケーションを他の HP Prime 電卓に送信します。

アプリケーション表示

ほとんどのアプリケーションには 3 種類の主要表示が用意されています。[記号表示]、[プロット表示]、[数値表示]の 3 つです。これらの表示はそれぞれ、計算オブジェクトの記号表現、グラフ表現、数値表現に対応します。これらの表示に移動するには、キーボードの左上にある Symb Setup 、 Plot Setup 、または Num Setup キーを押します。これらの表示では通常、計算オブジェクト（式や開放文など）の定義、プロット、値表示ができます。

各表示にはセットアップのための表示が用意され、そこでデータの表示方法を設定できます。これらのセットアップ表示は[記号のセットアップ]、[プロットのセットアップ]、[数値のセットアップ]と呼ばれます。表示するには、 Shift Symb Setup 、 Shift Plot Setup 、または Shift Num Setup を押します。

すべてのアプリケーションに 6 種類すべての表示があるわけではありません。どの表示があるかはアプリケーションの用途および複雑さによって異なります。たとえば、スプレッドシート アプリケーションには[プロット表示]と[プロットのセットアップ]の両方がなく、2 次エクスプローラー アプリケーションには[プロット表示]だけしかありません。各アプリケーションに用意されている表示については、以降のセクションで説明します。

データストリーマー アプリケーションは、この章では説明していません。このアプリケーションについて詳しくは、HP StreamSmart 410 ユーザー ガイドを参照してください。

[記号表示]

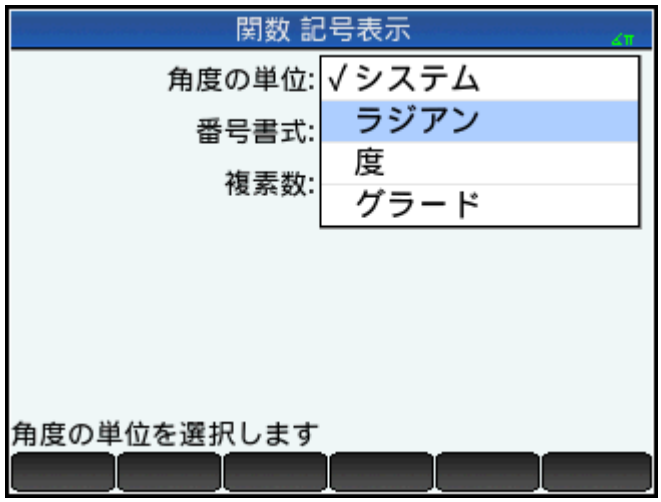
次の表に、各アプリケーションの[記号表示]でできる操作の概要を示します。

アプリケーション	[記号表示]でできる操作
高度なグラフ作成	開放文を最大 10 個まで指定できます
金融	該当なし
関数	x の実数値関数 y を最大 10 個まで指定できます
幾何	幾何学的作図の記号定義を表示できます
推論	仮説検証を行うか信頼水準をテストするかを選択し、テストのタイプを選択できます
1 次エクスプローラー	該当なし
線形ソルバー	該当なし
パラメトリック	t のパラメトリック関数 x および y を最大 10 個まで指定できます
極座標	角度 θ の極関数 r を最大 10 個まで指定できます
2 次エクスプローラー	該当なし

アプリケーション	[記号表示]でできる操作
数列	数列関数を最大 10 個まで指定できます
解く	方程式を最大 10 個まで指定できます
スプレッドシート	該当なし
1 変数統計	単変量分析を最大 5 個まで指定できます
2 変数統計	多変量分析を最大 5 個まで指定できます
三角ソルバー	該当なし
三角法エクスプローラー	該当なし

[記号のセットアップ]表示

[記号のセットアップ]表示はどのアプリケーションでも同じです。角度の単位、数値の形式、および複素数の形式についてシステム全体の設定を変更できます。ここでの変更は現在のアプリケーションにのみ適用されます。



すべてのアプリケーションに適用される設定はホーム設定および CAS 設定で変更できます。

[プロット表示]

次の表に、各アプリケーションの[プロット表示]でできる操作の概要を示します。

アプリケーション	[プロット表示]でできる操作
高度なグラフ作成	[記号表示]で選択した開放文のグラフをプロットおよび操作できます
金融	償却グラフを表示できます
関数	[記号表示]で選択した関数をプロットおよび操作できます
幾何	幾何学的作図を作成および操作できます
推論	テスト結果のプロットを表示できます
1 次エクスプローラー	一次方程式を操作したり、自分の知識をテストしたりできます
線形ソルバー	該当なし

アプリケーション	[プロット表示]でできる操作
パラメトリック	[記号表示]で選択した関数をプロットおよび操作できます
極座標	[記号表示]で選択した関数をプロットおよび操作できます
2 次エクスプローラー	二次方程式を操作したり、自分の知識をテストしたりできます
数列	[記号表示]で選択した数列をプロットおよび操作できます
解く	[記号表示]で選択した単一関数をプロットおよび操作できます
スプレッドシート	該当なし
1 変数統計	[記号表示]で選択した分析をプロットおよび操作できます
2 変数統計	[記号表示]で選択した分析をプロットおよび操作できます
三角ソルバー	該当なし
三角法エクスプローラー	正弦関数を操作したり、自分の知識をテストしたりできます

[プロットのセットアップ]表示

次の表に、各アプリケーションの[プロットのセットアップ]表示でできる操作の概要を示します。

アプリケーション	[プロットのセットアップ]表示でできる操作
高度なグラフ作成	プロットの外観およびプロット環境を変更できます
金融	該当なし
関数	プロットの外観およびプロット環境を変更できます
幾何	描画環境の外観を変更できます
推論	該当なし
1 次エクスプローラー	該当なし
線形ソルバー	該当なし
パラメトリック	プロットの外観およびプロット環境を変更できます
極座標	プロットの外観およびプロット環境を変更できます
2 次エクスプローラー	該当なし
数列	プロットの外観およびプロット環境を変更できます
解く	プロットの外観およびプロット環境を変更できます
スプレッドシート	該当なし
1 変数統計	プロットの外観およびプロット環境を変更できます
2 変数統計	プロットの外観およびプロット環境を変更できます
三角ソルバー	該当なし
三角法エクスプローラー	該当なし

[数値表示]

次の表に、各アプリケーションの[数値表示]でできる操作の概要を示します。

アプリケーション	[数値表示]でできる操作
高度なグラフ作成	[記号表示]で選択した開放文によって生成される数値の表を表示できます
金融	貨幣の時間価値計算の値を入力できます
関数	[記号表示]で選択した関数によって生成される数値の表を表示できます
幾何	[プロット表示]で描画した幾何オブジェクトに対する計算を実行できます
推論	[記号表示]で選択したテストを実行するために必要な統計を指定できます
1 次エクスプローラー	該当なし
線形ソルバー	計算対象の一次方程式の係数を指定できます
パラメトリック	[記号表示]で選択した関数によって生成される数値の表を表示できます
極座標	[記号表示]で選択した関数によって生成される数値の表を表示できます
2 次エクスプローラー	該当なし
数列	[記号表示]で選択した数列によって生成される数値の表を表示できます
解く	既知値を入力して未知値を求めることができます
スプレッドシート	数値、テキスト、式などを入力できます。このアプリケーションでは[数値表示]がメイン表示になります
1 変数統計	分析のためのデータを入力できます
2 変数統計	分析のためのデータを入力できます
三角ソルバー	三角形の既知データを入力して未知データを求めることができます
三角法エクスプローラー	該当なし

[数値のセットアップ]表示

次の表に、各アプリケーションの[数値のセットアップ]表示でできる操作の概要を示します。

アプリケーション	[数値のセットアップ]表示でできる操作
高度なグラフ作成	[記号表示]で選択した開放文を使用して計算する数値を指定したり、ズーム倍率を設定したりできます
金融	該当なし
関数	[記号表示]で選択した関数を使用して計算する数値を指定したり、ズーム倍率を設定したりできます
幾何	該当なし
推論	該当なし
1 次エクスプローラー	該当なし

アプリケーション	[数値のセットアップ]表示でできる操作
線形ソルバー	該当なし
パラメトリック	[記号表示]で選択した関数を使用して計算する数値を指定したり、ズーム倍率を設定したりできます
極座標	[記号表示]で選択した関数を使用して計算する数値を指定したり、ズーム倍率を設定したりできます
2 次エクスプローラー	該当なし
数列	[記号表示]で選択した関数を使用して計算する数値を指定したり、ズーム倍率を設定したりできます
解く	該当なし
スプレッドシート	該当なし
1 変数統計	該当なし
2 変数統計	該当なし
三角ソルバー	該当なし
三角法エクスプローラー	該当なし

簡単な例

アプリケーションの一般的な操作手順を知るために役立つ、6 種類のアプリケーション表示の使用例を以下に示します。この例では極座標アプリケーションを使用します。

アプリケーションを開く

1. Apps Info を押してアプリケーション ライブラリを開きます。
2. 極座標アプリケーションのアイコンをタップします。

Polar アプリケーションが開き、最初に[記号表示]が表示されます。

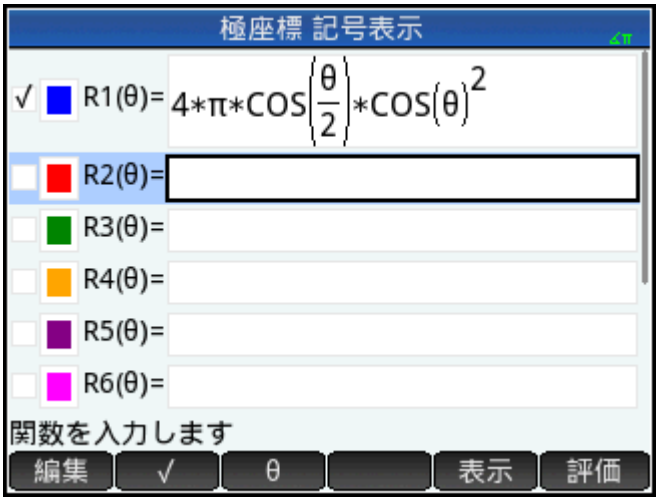
[記号表示]

極座標アプリケーションの[記号表示]では、プロットする極方程式を定義または指定します。この例では、方程式 $r = 5\pi\cos(\theta/2)\cos(\theta)^2$ をプロットします。

▲ 次のようにして、方程式 $r = 5\pi\cos(\theta/2)\cos(\theta)^2$ を定義します。

5 Shift π 3 # COS ACOS H θ ÷ x⁻¹ T 2 ▶ ▶ COS ACOS H θ ▶ √ x² L Enter ≈

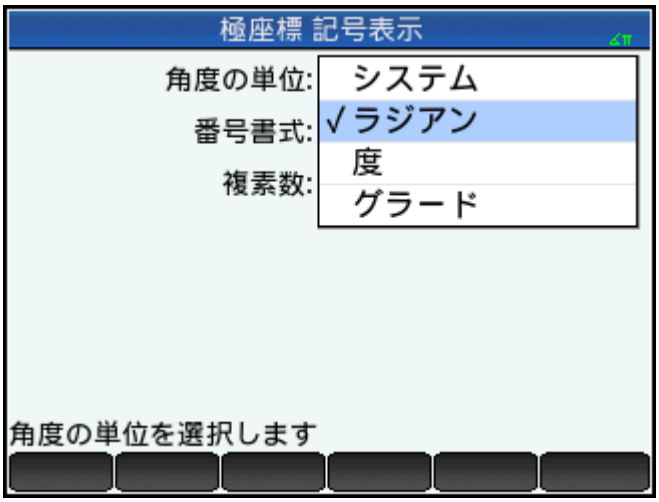
(代数入力モードを使用している場合は、5 Shift π 3 # COS ACOS H θ ÷ x⁻¹ T 2 ▶ COS ACOS H θ ▶ √ x² L Enter ≈ と入力します。)



角度単位をラジアンに設定すると、この方程式では左右対称の花びらのような形が描かれます。このアプリケーションの角度単位は[記号のセットアップ]表示で設定します。

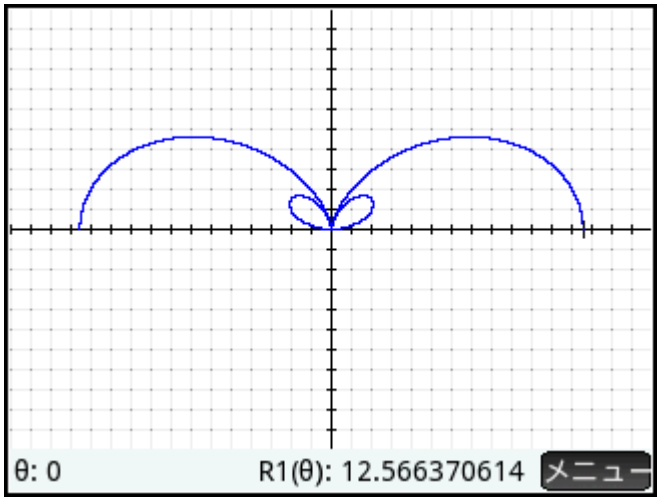
[記号のセットアップ]表示

1. Shift Symb Setup を押します。
2. [角度の単位]メニューから**[ラジアン]**を選択します。



[プロット表示]

▲ Plot Setup を押します。



方程式のグラフがプロットされます。ただしこの状態では、上の図のように花びらの半分しか見えません。残りの部分を表示するには、プロットのセットアップ パラメーターを変更する必要があります。

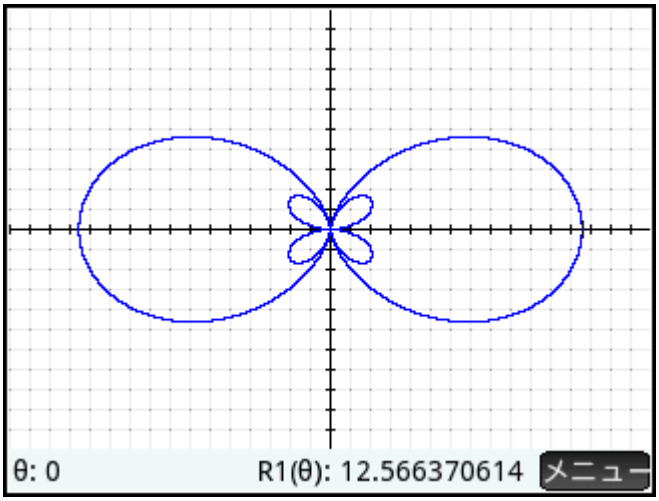
[プロットのセットアップ]表示

1. Shift Plot を押します。

2. 次のように入力して、**[θ 範囲]**の 2 つ目のフィールドを「4π」に設定します。

   ▶ 4 Shift 3 (π) OK

3. Plot Setup を押して[プロット表示]に戻ると、プロット全体が表示されます。



[数値表示]

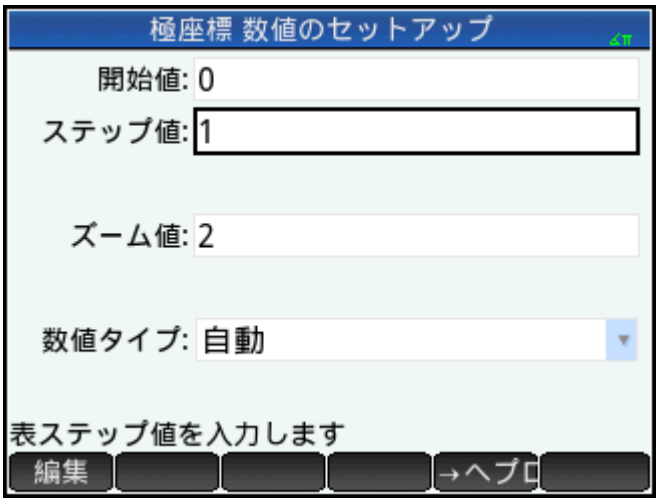
この方程式によって生成される値を[数値表示]で確認できます。

▲ Num Setup を押します。

ここではθが整数の場合だけを表示してみます。つまり、θ列の値が1ずつ増えるようにします。この設定は[数値のセットアップ]表示で行います。

[数値のセットアップ]表示

1. Shift Num Setup を押します。
2. **[ステップ値]**フィールドを「1」に設定します。



3. Num Setup を押して[数値表示]に戻ります。

これで、θ 列に 0 から 1 ずつ増える整数が表示され、R1 列には、[記号表示]で指定した方程式から求められる各値が表示されます。

[記号表示]の共通操作

このセクションの説明は、高度なグラフ作成、関数、パラメトリック、極座標、数列、および解くアプリケーションに適用されます。その他のアプリケーションについては、各アプリケーションの章を参照してください。

[記号表示]では通常、プロットや評価の対象となる関数または開放文を定義します。このセクションでは、関数と開放文の両方について「定義」という言葉を使用します。

[記号表示]を開くには、 Symb Setup を押します。

定義の追加

パラメトリック以外のアプリケーションでは、定義を入力するためのフィールドが 10 個あります。Parametric アプリケーションでは、定義が 2 つ 1 組で合計 20 個のフィールドがあります。

1. 空のフィールドをタップするかそこまでスクロールで移動することで、フィールドを選択します。
2. 定義を入力します。

   注記 : 定義に使用する変数は大文字で入力する必要があります。変数を小文字で入力すると、エラー メッセージが表示されます。

   操作方法がわからない場合は、69 ページの定義の構成要素を参照してください。
3. 入力し終えたら、 OK をタップするか、 Enter ≈ を押します。

新しい定義が定義リストに追加されます。

定義の変更

1. 変更する定義をタップするかそこまでスクロールで移動することで、定義を選択します。
2. 編集 をタップします。

   定義が入力行にコピーされます。
3. 定義を変更します。
4. 入力し終えたら、 OK をタップするか、 Enter ≈ を押します。

定義の構成要素

記号定義には、さまざまなソースのコンポーネントを使用できます。

- キーボード

キーボードでコンポーネントを直接入力できます。$2X^2 - 3$ を入力するには、2 ALPHA X x^2 − 3 と押します。

- ユーザー変数

  COST という名前の変数を作成した場合は、その変数名を入力するか、[変数]メニューのサブメニューである**[ユーザー]**メニューからその変数を選択することによって、定義に含めることができます。たとえば、$F1(X) = X^2 + COST$ のような定義を作成できます。

  ユーザー変数を選択するには、 Vars を押し、 ユーザ をタップして、**[ユーザー変数]**を選択してから、目的の変数を選択します。

- ホーム変数

  一部のホーム変数を記号定義に含めることができます。ホーム変数を選択するには、 Vars を押し、 ホーム をタップして、変数のカテゴリを選択してから、目的の変数を選択します。たとえば、$F1(X) = X^2 + Q$ のような定義を作成できます（Q は**[ホーム]**メニューの**[実数]**サブメニューにあります）。

- アプリケーション変数

  アプリケーションの設定、定義、および結果はすべて変数として保存されます。これらの変数の多くを記号定義に含めることができます。アプリケーション変数を選択するには、 Vars を押し、 アプリ をタップして、アプリケーションを選択してから、変数のカテゴリを選択して、目的の変数を選択します。たとえば、$F2(X) = X^2 + X -$ Root のような定義を作成できます。この定義の評価時には、関数アプリケーションで最後に計算された根の値が Root に代入されます。

- 数学関数

  [数学]メニューの一部の関数を定義に含めることができます。**[数学]**メニューはツールボックス メニュー（ ）の 1 つです。次の定義では、数学関数**[サイズ]**とホーム関数 L1 を組み合わせて使用しています：$F4(X) = X^2 - SIZE(L1)$。これは、$x^2 - n$ の計算で、n に L1 という名前のリストの要素数が入ることを示します。**[Size]**は、**[数学]**メニューのサブメニューである**[リスト]**メニュー内にあります。

- CAS 関数

  [CAS]メニューの一部の関数を定義に含めることができます。**[CAS]**メニューはツールボックス メニュー（ ）の 1 つです。次の定義では、CAS 関数 irem を使用しています：$F5(X) = X^2+$ CAS.irem(45,7)。irem を入力するには、**[整数]**メニューのサブメニューである**[商]**メニューで**[余り]**オプションを選択します。CAS のコマンドまたは関数を CAS 以外の場所での計算に使用するときは、プレフィックス CAS.を付ける必要があることに注意してください。

- アプリケーション関数

  [アプリ]メニューの一部の関数を定義に含めることができます。**[アプリ]**メニューはツールボックス メニュー（ ）の 1 つです。次の定義では、アプリケーション関数 PredY を使用しています：

  $F9(X) = X^2 +$ Statistics_2Var.PredY(6)

- **[カタログ]**メニュー

  [カタログ]メニューの一部の関数を定義に含めることができます。**[カタログ]**メニューはツールボックス メニュー（ ）の 1 つです。次の定義では、このメニューのコマンドとアプリケーション変数を組み合わせて使用しています：F6(X) = X^2 + INT(Root)。この定義の評価時には、関数アプリケーションで最後に計算された根の整数値が INT(Root)に代入されます。

- その他の定義

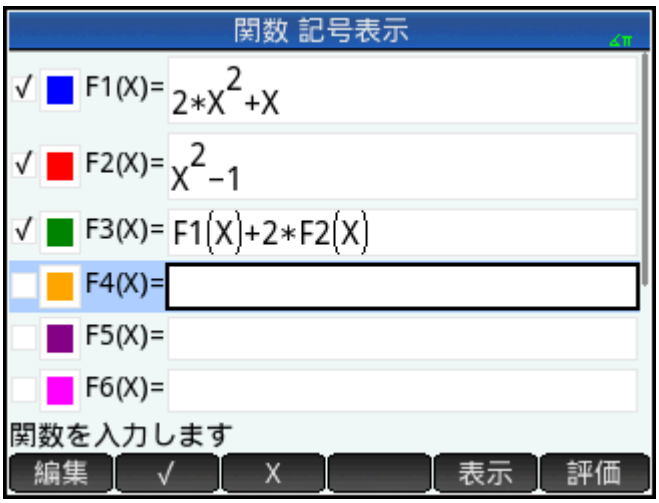
  たとえば、F3(X)を F1(X) * F2(X)のように定義できます。

従属定義の評価

従属定義とは、他の定義を使用する定義を指します。従属定義を評価することによって、すべての定義を 1 つにまとめることができます。

1. 従属式を選択します。
2. 評価 をタップします。

次の例について考えます。ここでは、F3(X)で他の 2 つの定義が使用されています。これが従属定義であり、この定義を評価できます。F3(X)を選択して 評価 をタップすると、F3(X)は 2 * X^2 + X+ 2 * (X^2 − 1)になります。



操作対象の定義の選択および選択解除

高度なグラフ作成、関数、パラメトリック、極座標、数列、および解くアプリケーションでは最大 10 個の定義を入力できます。ただし、[プロット表示]でのプロットや[数値表示]での評価の対象となるのは、[記号表示]で選択した定義のみです。

定義が選択されているかどうかは、定義の横にあるチェックマークで示されます。定義を作成すると、初期設定でチェックマークが付きます。定義をプロットや評価に含めたくない場合は、その定義を選択して √ をタップします。選択を解除した関数を再度選択する場合も同様に操作します。

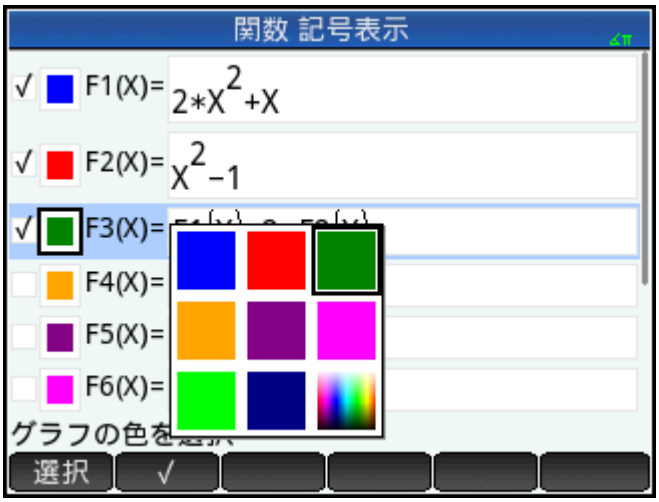
プロットの色の選択

関数や開放文はすべて異なる色でプロットされます。プロットの初期設定の色を変更するには、以下の操作を行います。

1. 関数定義の左側にあるカラー ボックスをタップします。

   定義を選択して Enter をタップすることでボックスを選択することもできます。

   Enter を押すたびに定義とカラー ボックス間で選択が移動します。



2. 選択 をタップします。
3. カラー ピッカーで色を選択します。

定義の削除

特定の定義を削除するには、以下の操作を行います。

1. 目的の定義を 1 回タップします（またはカーソル キーを使用して選択します）。
2. Del を押します。

すべての定義を削除するには、以下の操作を行います。

1. Shift Esc Clear を押します。
2. OK をタップするか Enter を押して操作を確定します。

[記号表示]：メニュー ボタンの概要

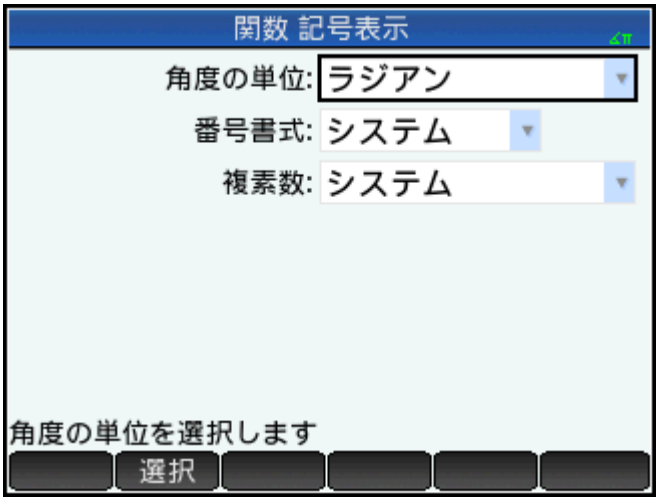
ボタン	用途
編集	選択した定義を編集するために入力行にコピーします。完了したら OK をタップします 新しい定義を追加するか、既存の定義を置き換えるときは、単にフィールドを選択して新しい定義を入力します
√	定義を選択（または選択解除）します

ボタン	用途
X （関数のみ）	関数アプリケーションで独立変数を入力します。 xtθn Define D を押すこともできます
X （高度なグラフ作成のみ）	高度なグラフ作成アプリケーションで X を入力します。 xtθn Define D を押すこともできます
Y （高度なグラフ作成のみ）	高度なグラフ作成 アプリケーションで Y を入力します
T （パラメトリックのみ）	パラメトリック アプリケーションで独立変数を入力します。 xtθn Define D を押すこともできます
θ （極座標のみ）	極座標アプリケーションで独立変数を入力します。 xtθn Define D を押すこともできます
N （数列のみ）	数列アプリケーションで独立変数を入力します。 xtθn Define D を押すこともできます
= （解くのみ）	解くアプリケーションで等号を入力します。 Shift • = のショートカットです
表示	選択した定義を全画面モードで表示します
評価	従属定義を評価します。71 ページの従属定義の評価を参照してください

[記号のセットアップ]表示の共通操作

[記号のセットアップ]表示はどのアプリケーションでも同じです。主に、**[ホーム設定]**ウィンドウで指定したシステム全体の設定のうちの 3 つをアプリケーション内で変更するために使用します。

[記号のセットアップ]表示を開くには、 Shift Symb Setup を押します。



システム全体の設定の変更

1. 変更する設定をタップします。

   フィールド名またはフィールドのどちらをタップしてもかまいません。

2. 設定をもう一度タップします。

   オプションのメニューが表示されます。

3. 新しい設定を選択します。

注記：[番号書式]メニューで**[固定]**、**[関数]**、または**[工学]**オプションを選択した場合は、有効数字の必須桁数を入力するための 2 つ目のフィールドが表示されます。

フィールドを選択して 選択 をタップしてから、新しい設定を選択することもできます。

初期設定の復元

初期設定に戻すと、**[ホーム設定]**画面での設定に戻ります。

特定のフィールドを初期設定に戻すには、以下の操作を行います。

1. 目的のフィールドを選択します。
2. Del を押します。

すべてのフィールドを初期設定に戻すには、 Shift Esc Clear を押します。

[プロット表示]の共通操作

このセクションでは、多くのアプリケーションに共通する[プロット表示]の機能について詳しく説明します。特定のアプリケーションでのみ使用できる機能については、各アプリケーションの章を参照してください。

[プロット表示]を開くには、 Plot Setup を押します。

ズーム

[プロット表示]で簡単にズームを行うには、2 本指ピンチ ズームを使用します。2 本指ピンチ ズーム ジェスチャを左右方向に実行すると、x 軸のみが拡大または縮小されます。2 本指ピンチ ズームを上下方向に実行すると、y 軸のみが拡大または縮小されます。2 本指ピンチ ズームを斜め方向に実行すると、正方形に拡大または縮小されます（つまり、両方の軸が拡大または縮小されます）。

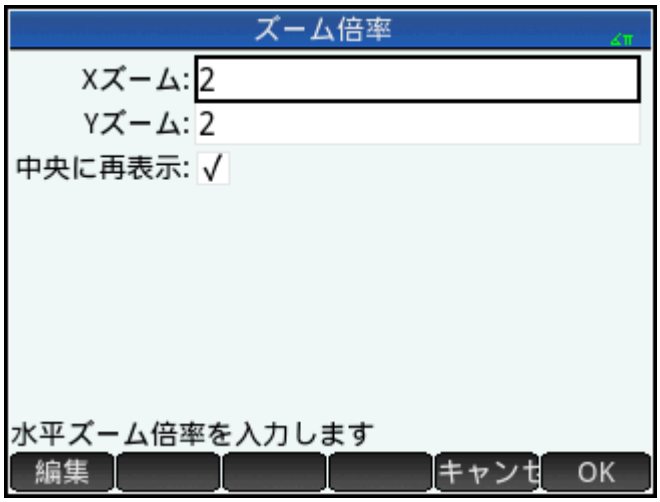
さらに正確な制御を行うには、[ズーム]メニューのオプションを使用します。これらのオプションでは、水平または垂直の倍率あるいは両方の倍率を使用します。初期設定では、これらの倍率はどちらも 2 に設定されています。縮小すると、スケールが倍率で乗算され、画面に表示される範囲が大きくなります。拡大すると、スケールが倍率で除算され、画面に表示される範囲が小さくなります。

ズーム倍率

初期設定のズーム倍率を変更するには、以下の操作を行います。

1. アプリケーションの[プロット表示]を開きます（ Plot Setup ）。
2. メニュー をタップして、[プロット表示]メニューを開きます。
3. ズーム をタップして、[ズーム]メニューを開きます。
4. スクロールして**[倍率の設定]**を選択します。

   [ズーム倍率]画面が表示されます。



5. 一方または両方のズーム倍率を変更します。
6. [プロット表示]で常にプロットの中心を現在のカーソル位置に合わせたい場合は、[中央に再表示]を選択します。
7. OK をタップするか Enter を押します。

ズーム オプション

ズーム オプションは以下の方法で操作できます。

- タッチスクリーン
- キーボード

- [プロット表示]の ズーム メニュー
- **[表示]**メニュー（ View Copy ）

ズーム ジェスチャ

[プロット表示]では、2 本指ピンチ ズームを斜め方向に実行すると、垂直および水平の両方の方向に同じスケール倍率で拡大または縮小されます。2 本指ピンチ ズームを上下方向に実行すると、y 軸のみが拡大または縮小されます。2 本指ピンチ ズームを左右方向に実行すると、x 軸のみが拡大または縮小されます。

[数値表示]では、2 本指ピンチ ズームを上下方向に実行すると、選択された行が拡大または縮小されます。拡大すると x 値の公差が小さくなり、縮小すると大きくなります。

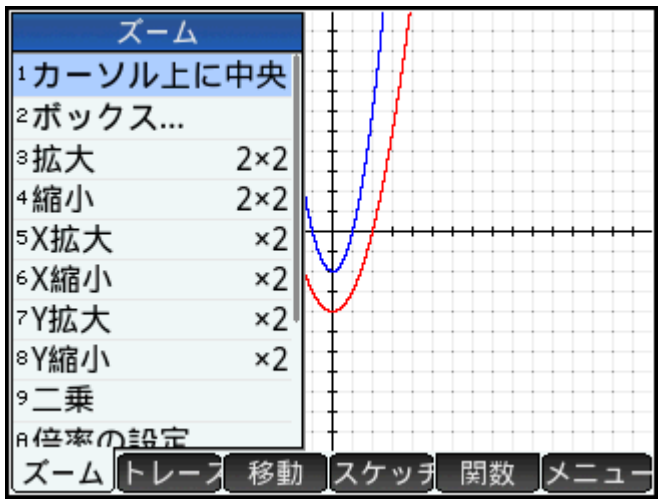
ズーム キー

ズーム キーは 2 つあります。 + Ans ; を押すと拡大され、 − Base ; を押すと縮小されます。拡大や縮小の度合いは**[ズーム倍率]**設定によって決まります。

[ズーム]メニュー

[プロット表示]で ズーム をタップして、オプションをタップします。 ズーム が表示されていない場合は、 メニュー をタップしてください。

ズーム オプションの説明を次の表に示します。79 ページのズーム例に示す操作例も参考にしてください。



オプション	結果
カーソル上に中央	カーソルが画面の中央に位置するようにプロットを再描画します。拡大や縮小はしません
ボックス	77 ページのボックス ズームを参照してください
拡大	横軸および縦軸のスケールを**[X ズーム]**および**[Y ズーム]**で除算します（ズーム倍率値は**[倍率の設定]**オプションで設定します）。たとえば、ズーム倍率を両方とも 4 に設定

オプション	結果
	した場合は、1 ピクセルあたりに描画される単位数が 1/4 になります（ショートカット： + Ans ; を押します）
縮小	横軸および縦軸のスケールを**[X ズーム]**および**[Y ズーム]**で乗算します（ショートカット： − Base ; を押します）
X 拡大	横軸のスケールのみを**[X ズーム]**で除算します
X 縮小	横軸のスケールのみを**[X ズーム]**で乗算します
Y 拡大	縦軸のスケールのみを**[Y ズーム]**で除算します
Y 縮小	縦軸のスケールのみを**[Y ズーム]**で除算します
2 乗	縦軸のスケールを横軸のスケールと一致させます。これは、ボックス ズーム、X のみのズーム、Y のみのズームを元に戻すときに便利です
自動スケール	x 軸設定を所定としてプロット部分が表現されるように縦軸のスケールを調整します（数列、極座標、パラメトリック、1 変数統計、および 2 変数統計アプリケーションでは両方の軸のスケールが調整されます）。自動スケール処理では、最初に選択されている関数によって最適なスケールが判断されます
小数	両方の軸のスケールを、1 ピクセルが 0.1 単位になるように調整します。これは、**[X 範囲]**および**[Y 範囲]**を初期設定値にリセットするのと同じことです
整数	横軸のスケールのみを、1 ピクセルが 1 単位になるように調整します
三角法	横軸のスケールを、1 ピクセルが π/24 ラジアン（7.5 度）になるように調整します。また、縦軸のスケールを、1 ピクセルが 0.1 単位になるように調整します
ズーム解除	表示を直前のズームに戻します。 **注記：**このオプションは、ズーム操作を実行した後でのみ使用できます

ボックス ズーム

ボックス ズーム機能を使用すると、画面内の指定した領域を拡大できます。

1. [プロット表示]メニューを開き、ズーム をタップして、**[ボックス]**を選択します。
2. 拡大する領域のいずれかの隅になる位置をタップして、OK をタップします。
3. その対角の隅の位置をタップして、OK をタップします。

   指定した領域が画面いっぱいに表示されます。初期設定の表示に戻すには、ズーム をタップして**[少数]**を選択します。

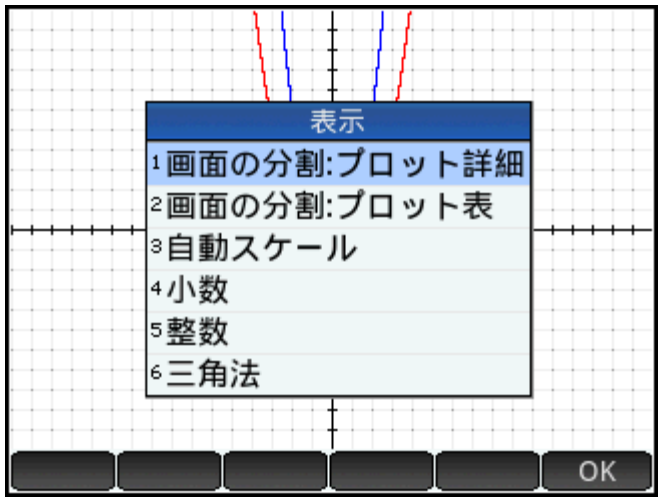
カーソル キーを使用して領域を指定することもできます。

[表示]メニュー

よく使用されるズーム オプションは、[表示]メニューからも選択できます。以下のオプションがあります。

- 自動スケール
- 小数

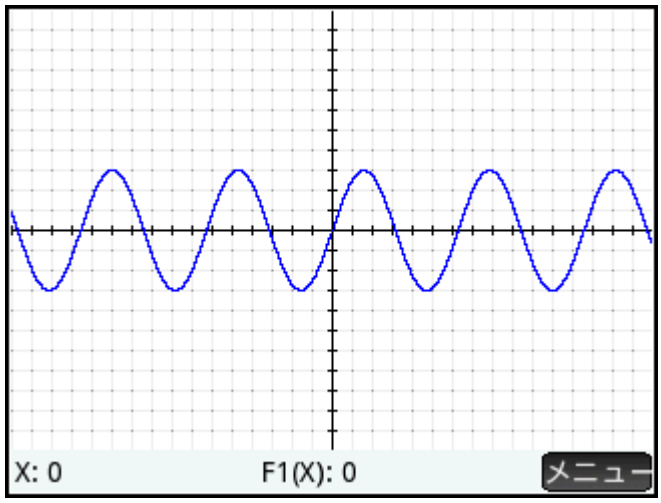
- 整数
- 三角法



これらのオプションは、現在作業している表示に適用されます。

分割画面表示を使用したズームのテスト

ズームを試してみるときは、画面を 2 つに分割して同じプロットを 2 つ表示し、一方のみにズームを適用すると便利です。次の図では、y = 3sinx のプロットが表示されています。



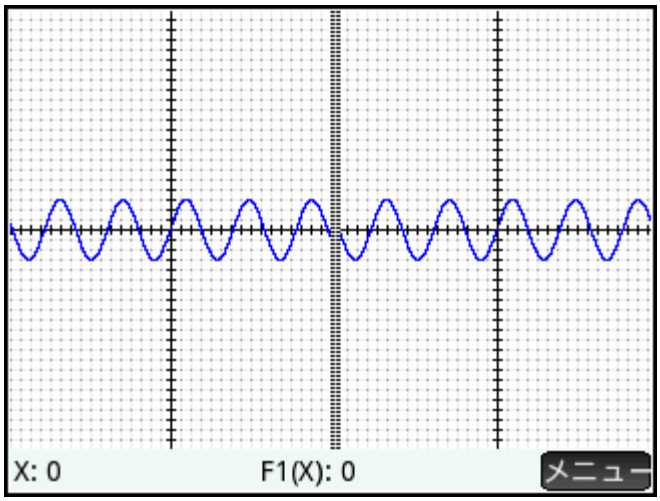
画面を 2 つに分割するには、以下の操作を行います。

1. [表示]メニューを開きます。

   を押します。

2. **[画面の分割：プロット詳細]**を選択します。

結果は次の図のようになります。実行したズーム操作は、画面右半分のプロットにのみ適用されます。これにより、ズーム オプションを選択してズームを試してみることができます。



注記：左側の元のプロットを右側のズームしたプロットに置き換えるときは、←ヘプロ をタップします。

画面の分割を解除するには、 Plot Setup を押します。

ズーム例

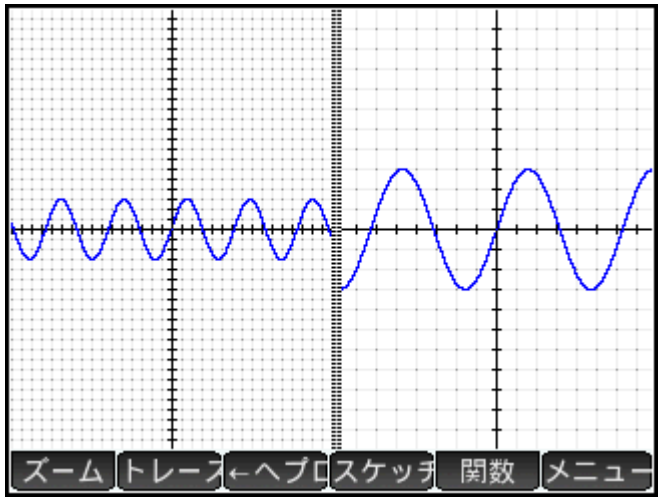
3sinx のプロットに対するズーム オプションの適用例を以下に示します。ズーム倍率は初期設定の 2 × 2 とします。ズームの効果がわかりやすいように、前述の分割画面モードを使用しています。

注記：[ズーム]メニューには**[ズーム解除]**オプションがあります。このオプションを使用すると、ズームを直前の状態に戻すことができます。**[ズーム]**メニューが表示されない場合は、メニュー をタップしてください。

拡大

ズーム **[拡大]**

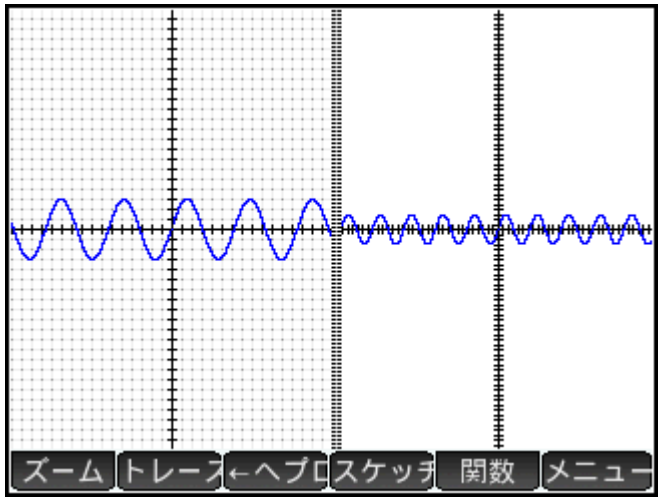
ショートカット： + Ans ; を押します



縮小

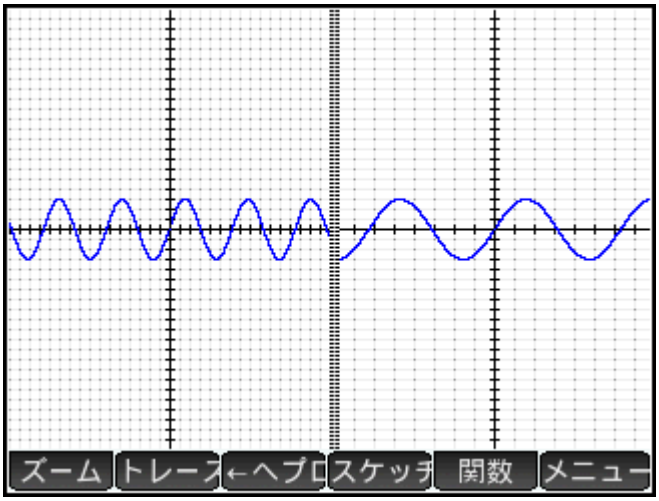
メニュー ズーム **[縮小]**

ショートカット： ー を押します



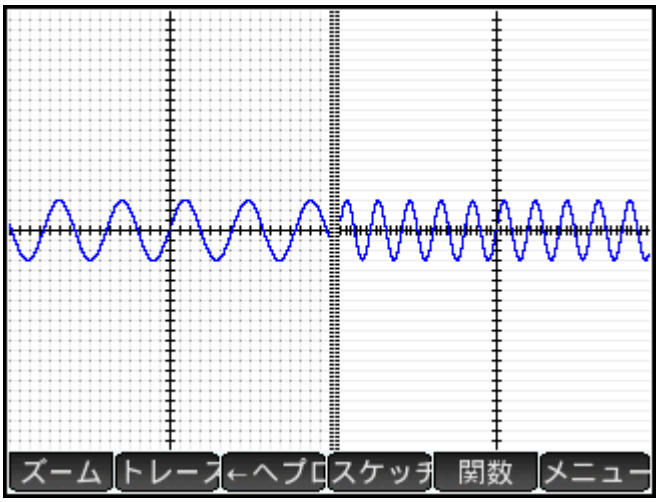
X 拡大

ズーム **[X 拡大]**



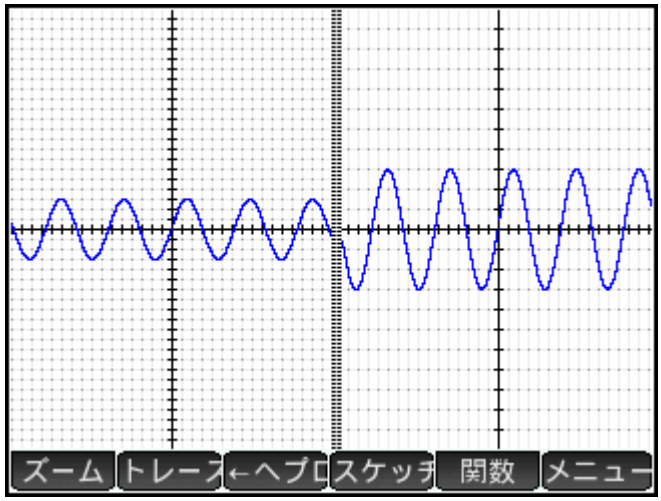
X 縮小

ズーム **[X 縮小]**



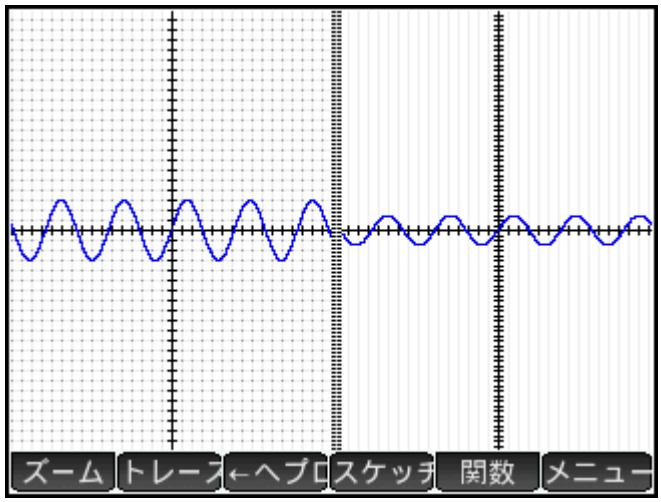
Y 拡大

ズーム **[Y 拡大]**



Y 縮小

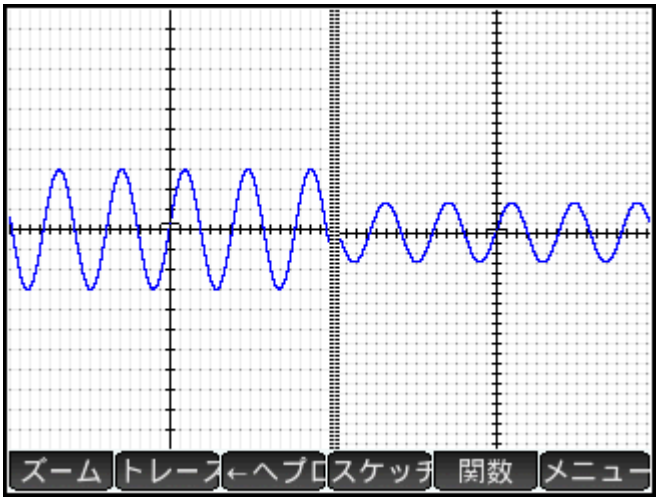
ズーム **[Y 縮小]**



2 乗

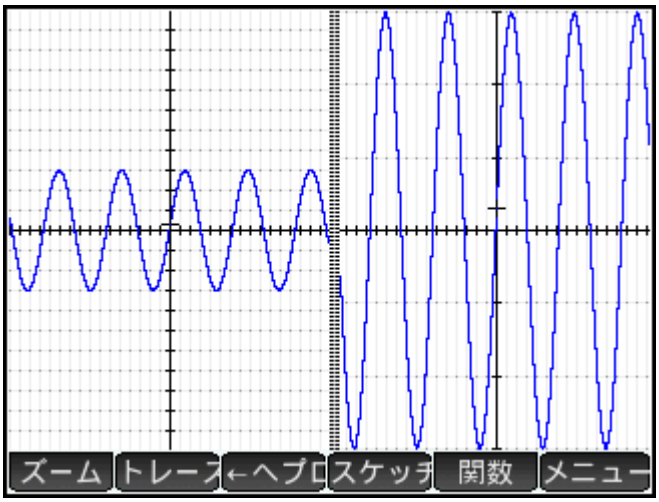
ズーム **[2 乗]**

注記：この例では、左側のプロットに**[Y 拡大]**ズームが適用されています。X と Y のスケールが等しいときは、**[2 乗]**ズームを適用するとプロットが初期状態に戻ります。



自動スケール

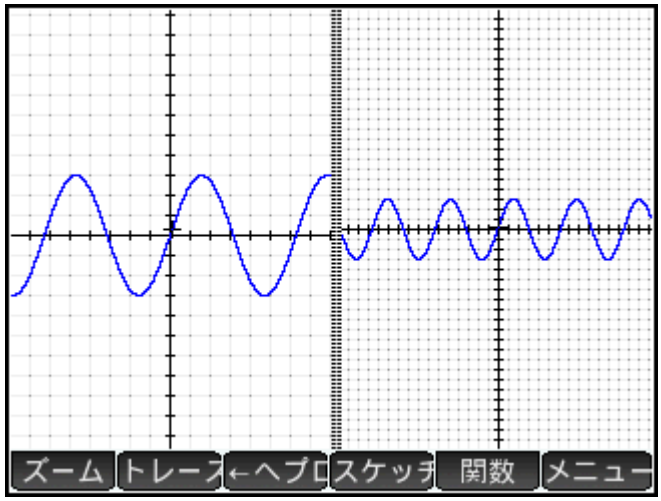
ズーム **[自動スケール]**



小数

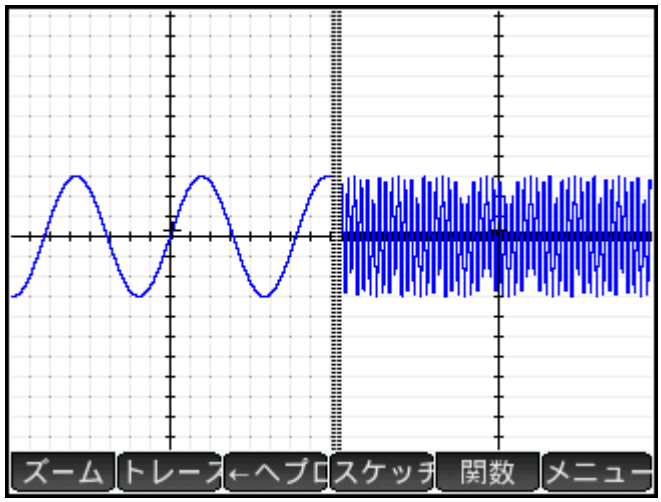
ズーム **[少数]**

注記 : この例では、左側のプロットに**[X 拡大]**ズームが適用されています。X と Y のスケールが等しいときは、**[Decimal]**ズームを適用するとプロットが初期状態に戻ります。



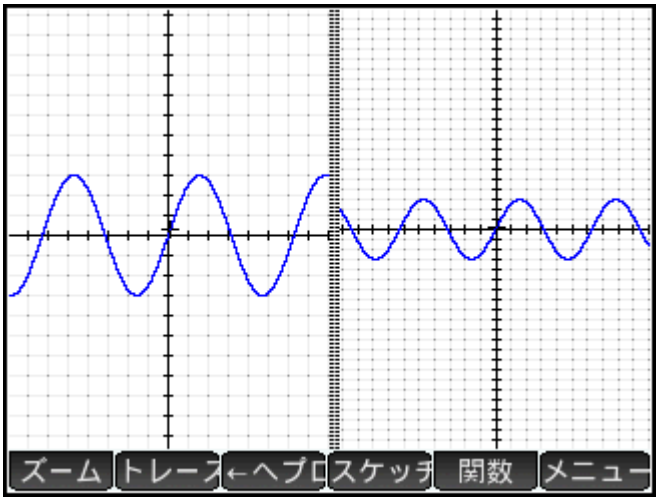
整数

ズーム **[整数]**



三角法

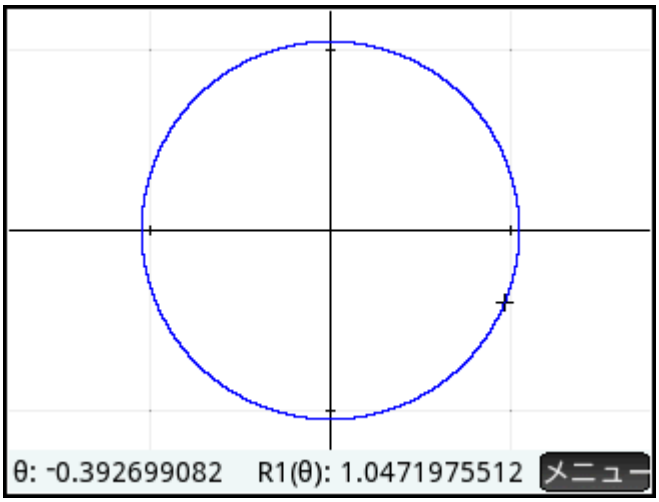
ズーム **[三角法]**



トレース

このセクションの説明は、高度なグラフ作成、関数、パラメトリック、極座標、数列、解く、1 変数統計、および 2 変数統計アプリケーションに適用されます。

トレース機能を使用すると、カーソル（トレース カーソル）を現在のグラフに沿って動かすことができます。トレース カーソルを動かすには、◀ または ▶ を押します。また、現在のプロット上またはプロット付近をタップして動かすこともできます。この場合、グラフ上で、タップした位置に最も近いポイントにトレース カーソルが移動します。



カーソルの現在の座標は画面の下部に表示されます。メニュー ボタンによって座標が隠れている場合は、メニュー をタップしてボタンを非表示にしてください。

トレース モードおよび座標表示は、プロットの描画時に自動的に有効になります。

プロットの選択

高度なグラフ作成以外のアプリケーションでは、複数のプロットが表示されている場合、▲ または ▼ を押してトレース カーソルを目的のプロットまで移動します。

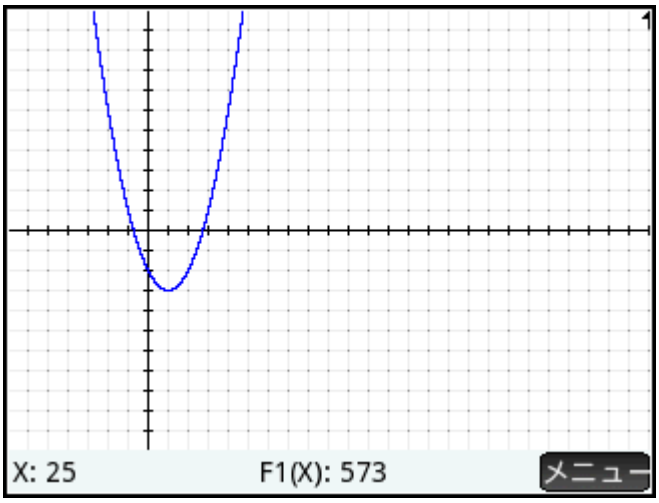
高度なグラフ作成アプリケーションでは、目的のプロットをタップしたまま保持します。プロットが選択されるか、表示されるプロット メニューでプロットを選択できます。

関数の評価

トレース機能の主な用途の 1 つは、プロットされた定義を評価することです。たとえば、[記号表示]で F1(X) = (X − 1)² − 3 と定義した場合、X が 25 のときの値を求めるには以下の操作を行います。

1. [プロット表示]を開きます（ Plot Setup ）。
2. 画面の下部にメニューが表示されていない場合は、 メニュー をタップします。
3. 移動 をタップします。
4. 「25」と入力して OK をタップします。
5. メニュー をタップします。

   X が 25 のときの F1(X)の値が画面の下部に表示されます。



HP Prime 電卓では、独立変数が特定の値をとるときの関数の値をさまざまな方法で求めることができます。[数値表示]で関数を評価することもできます（92 ページの[数値表示]の共通操作を参照してください）。さらに、[記号表示]で定義した式をホーム表示で評価することもできます。たとえば、F1(X) = (X − 1)² − 3 と定義した場合、ホーム表示で F1(4)と入力して Enter ≈ を押すと、(4 − 1)² − 3 = 6 で結果が 6 と表示されます。

トレース機能のオン/オフ

- トレース機能をオンにするには、 トレース をタップします。
- トレース機能をオフにするには、 トレース をタップします。

これらのオプションが表示されていない場合は、 メニュー をタップしてください。

トレース機能をオフにすると、カーソル キーを押したときに、カーソルの動きはプロット上に制限されなくなります。

[プロット表示]：メニュー ボタンの概要

ボタン	用途
ズーム	ズーム オプションのメニューを表示します。75 ページのズーム オプションを参照してください
トレース/トレース	トレース機能のオン/オフを切り替えます。85 ページのトレースを参照してください
移動	カーソルのジャンプ先の値を指定するための入力フォームを表示します。独立変数の値を入力できます
関数	プロットの分析オプションのメニューを表示します
([関数]と[変数統計 2]のみ)	
定義	現在の関数の記号定義を表示します。[関数]アプリと[変数統計 2]アプリでは、このエントリーは 関数 メニューにあります
メニュー	画面の下部に表示されるその他のボタンの表示/非表示を切り替えます

[プロット表示]でのコピーおよび貼り付けの操作

多くのアプリで、[プロット表示]で Shift View Copy を押すとコピーするためのオプションのリストが表示されます。現在の表示を任意のグラフィックス変数（G1 ~ G9）にコピーしたり、選択した x 値または y 値をクリップボードにコピーしたりできます。

[プロットのセットアップ]表示の共通操作

このセクションでは、以下に挙げるアプリケーションに共通の操作についてのみ説明します。[プロットのセットアップ]表示で実行できるアプリケーション固有の操作については、各アプリケーションの章を参照してください。

[プロットのセットアップ]表示を開くには、 Shift Plot Setup を押します。

[プロット表示]の設定

このセクションの説明は、高度なグラフ作成、関数、パラメトリック、極座標、数列、1 変数統計、および 2 変数統計アプリケーションに適用されます。

[プロットのセットアップ]表示では、[プロット表示]の外観およびグラフのプロット方法を設定できます。設定オプションは 3 ページにわたって表示されます。ページ間を移動するには、上下にスワイプするか、メニュー キーを使用します。

ヒント : [記号表示]で選択したグラフを表示するために[プロット表示]に移動しても、グラフが表示されないことがあります。ほとんどの場合これは、[プロットのセットアップ]表示で設定した範囲外にプロット値があることが原因です。この場合、すばやくグラフを表示するには、 View Copy を押して**[自動スケール]**を選択します。これによって、[プロットのセットアップ]表示での範囲設定も変更されます。

ページ 1

設定フィールド	用途
T 範囲 （パラメトリックのみ）	プロットする T 値の範囲を設定します。フィールドは 2 つあります。一方に最小値を指定し、もう一方に最大値を指定します
[T ステップ] （パラメトリックのみ）	連続する T 値の増分間隔を設定します
[θ 範囲] （極座標のみ）	プロットする角度値の範囲を設定します。フィールドは 2 つあります。一方に最小値を指定し、もう一方に最大値を指定します
[θ ステップ] （極座標のみ）	連続する角度値の増分間隔を設定します
[シーケンスプロット] （数列のみ）	プロットのタイプを設定します。[階段状]または[クモの巣]を選択できます
[N 範囲] （数列のみ）	プロットする N 値の範囲を設定します。フィールドは 2 つあります。一方に最小値を指定し、もう一方に最大値を指定します
[H 幅] （1 変数統計のみ）	ヒストグラムの棒の幅を設定します
[H 範囲] （1 変数統計のみ）	ヒストグラムに含める値の範囲を設定します。フィールドは 2 つあります。一方に最小値を指定し、もう一方に最大値を指定します
[X 範囲]	x 軸の初期範囲を設定します。フィールドは 2 つあります。一方に最小値を指定し、もう一方に最大値を指定します。[プロット表示]内ではパン操作やズーム操作によって範囲を変更できます

設定フィールド	用途
[Y 範囲]	y 軸の初期範囲を設定します。フィールドは 2 つあります。一方に最小値を指定し、もう一方に最大値を指定します。[プロット表示]内ではパン操作やズーム操作によって範囲を変更できます
[X 目盛]	x 軸の目盛の間隔を設定します
[Y 目盛]	y 軸の目盛の間隔を設定します

ページ 2

設定フィールド	用途
[軸]	軸の表示/非表示を切り替えます
[ラベル]	各軸の端に現在の値範囲を示す値を表示します
[グリッドドット]	縦横のグリッド線の交点にドット マークを表示します
[グリッド線]	x と y の整数値の位置に縦横のグリッド線を表示します
[カーソル]	トレース カーソルの外観を設定します。[標準]、[反転中]、または[点滅中]を選択できます
[接続] (2 変数統計のみ)	データ ポイントを直線で結びます
[方式] (両変数統計アプリケーションを除く)	グラフ化の方式を設定します。[adaptive]、[Fixed-Step セグメント]、または[Fixed-Step ドット]を選択できます。次のセクションを参照してください

ページ 3

一部の HP Prime アプリでは、[プロット表示]に背景画像を使用できます。画像を選択し、これらのアプリの[プロット表示]の外観を設定するには、[プロット設定]メニューのページ 3 を使用します。

▲ 背景画像メニューを開くには、Shift Plot Setup を押し、ページ 1/3 を 2 回タップします。

背景画像を設定するには、以下の操作を行います。

1. 背景のサイズと位置を選択します。オプションは、以下のとおりです。
   - **[背景なし]**：初期設定では、背景画像は使用されません。
   - **[中央]**：選択された画像を[プロット表示]の垂直方向および水平方向の中央に配置します。
   - **[画面に合わせて伸縮]**：選択した画像が、[プロット表示]のディスプレイ全体に合わせて、垂直方向および水平方向の両方に伸縮されます。
   - **[ウィンドウに合わせる]**：選択した画像が、[プロット表示]の幅または高さに合わせて、水平方向または垂直方向のどちらかに伸縮されます。
   - **[XY の範囲]**：[プロット表示]に画像を配置する x と y の範囲をユーザーが入力します。
2. **[不透明度]**ボックスには 0 ~ 100 の整数を入力します。0 は透明です。100 は完全に不透明です。
3. 背景画像を選択します。アプリに関連付けられているすべての画像が表示されます。その後には、電卓メモリに組み込まれた画像がすべて表示されます。左右にスワイプして利用可能な画像を表示し、画像をタップします。

これで、その背景画像が[プロット表示]に表示されます。

[XY の範囲]オプションを選択した場合は、軸をドラッグするか 2 本指ピンチ ズーム ジェスチャを実行すると、特定の機能にスクロールしたり、画像を拡大または縮小したりできます。それ以外の場合、[プロット表示]のサイズが変わっても画像には影響ありません。

[プロット設定]のページ 3 では、別の HP Prime アプリから画像をインポートすることもできます。

別の HP Prime アプリから画像をインポートするには、以下の操作を行います。

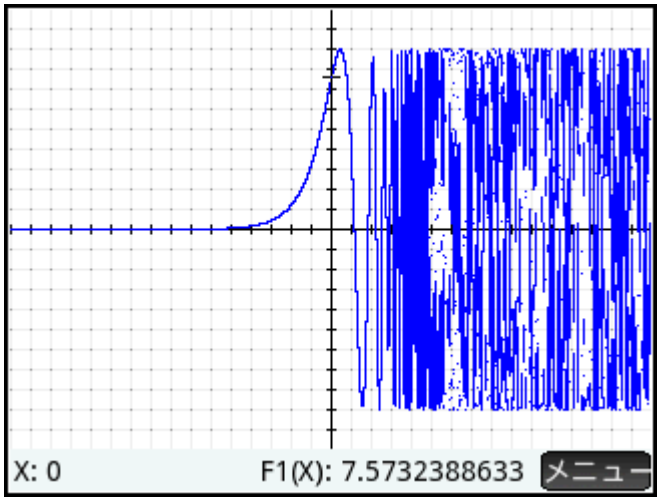
1. 読込 をタップします。
2. HP Prime アプリを選択します。
3. 左右にスワイプして、アプリに関連付けられているすべての画像を表示します。
4. 画像をタップしてから OK をタップすると、画像が現在のアプリにインポートされます。

HP Prime アプリに画像を関連付ける方法については、『HP 接続キット ユーザー ガイド』を参照してください。

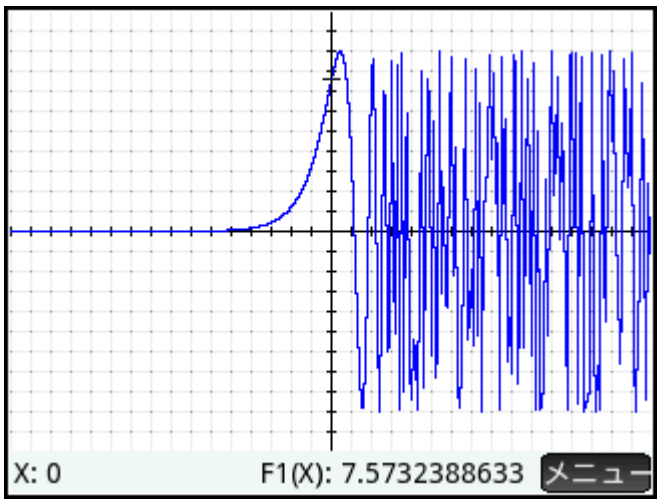
グラフ化の方式

HP Prime 電卓では、グラフ化の方式が 3 種類用意されています。以下では、関数 f(x) = 9*sin(e^x)への適用例を示しながら各方式について説明します。

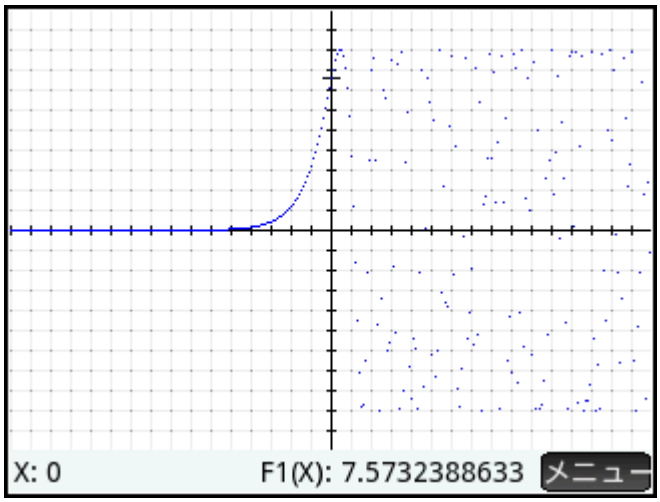
- **Adaptive**：非常に正確な結果が得られ、初期設定で使用されます。この方式を選択すると、複雑な関数ではプロットに時間がかかることがあります。その場合はメニュー バーに Sto ▸ が表示され、必要に応じてプロット処理を中止できます。



- **Fixed-Step セグメント**：x 値の標本を抽出して、対応する y 値を計算し、点をプロットして線で結びます。



- **Fixed-Step ドット**：[Fixed-Step セグメント]とほぼ同じですが、点を線で結びません。



初期設定の復元

このセクションの説明は、高度なグラフ作成、関数、パラメトリック、極座標、数列、解く、1 変数統計、2 変数統計、および幾何アプリケーションに適用されます。

特定のフィールドを初期設定に戻すには、以下の操作を行います。

1. 目的のフィールドを選択します。
2.

を押します。

すべてのフィールドを初期設定に戻すには、

を押します。

[数値表示]の共通操作

このセクションの説明は、高度なグラフ作成、関数、パラメトリック、および極座標アプリケーションに適用されます。

このセクションでは、多くのアプリケーションに共通する[数値表示]の機能について詳しく説明します。特定のアプリケーションでのみ使用できる機能については、各アプリケーションの章を参照してください。

[数値表示]には、評価の表が表示されます。[記号表示]の各定義が、独立変数の値範囲に基づいて評価されます。独立変数の範囲および精度は変更することも、初期設定のままにすることもできます。

[数値表示]を開くには、

を押します。

ズーム

[プロット表示]とは異なり、[数値表示]ではズーム操作をしても表示サイズは変わりません。代わりに、独立変数の連続値の増分間隔（つまり[数値のセットアップ]表示の**[ステップ値]**設定）が変わります (99 ページの[数値のセットアップ]表示の共通操作を参照)。拡大すると増分が小さくなり、縮小すると増分が大きくなります。ズーム操作をする前に選択していた行は変更されません。

通常のズーム オプションでは、拡大や縮小の度合いがズーム倍率によって決まります。[数値表示]では、[数値のセットアップ]表示の**[ズーム値]**フィールドによって決まります。初期設定値は 4 です。つまり、現在の増分（**[ステップ値]**）が 0.4 に設定されている場合、拡大すると増分間隔が 4 分の 1 になります。たとえば、x 値が 10、10.4、10.8、11.2 と続く場合、操作後は 10、10.1、10.2、10.3、10.4 になります。縮小した場合は、10、10.4、10.8、11.2 が 10、11.6、13.2、14.8、16.4 となります。

図 6-1 ズーム前

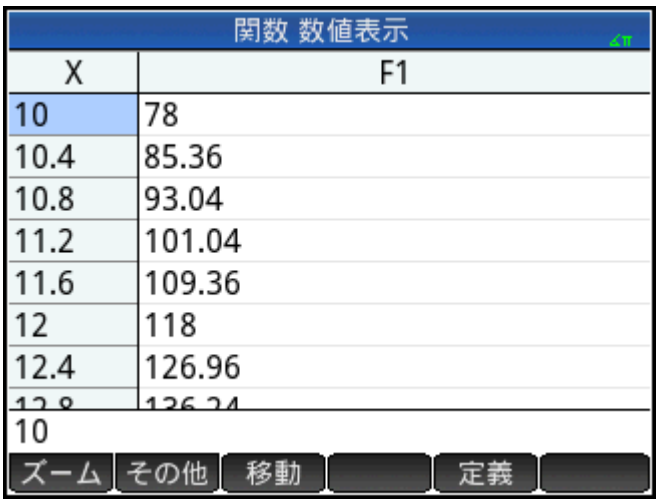
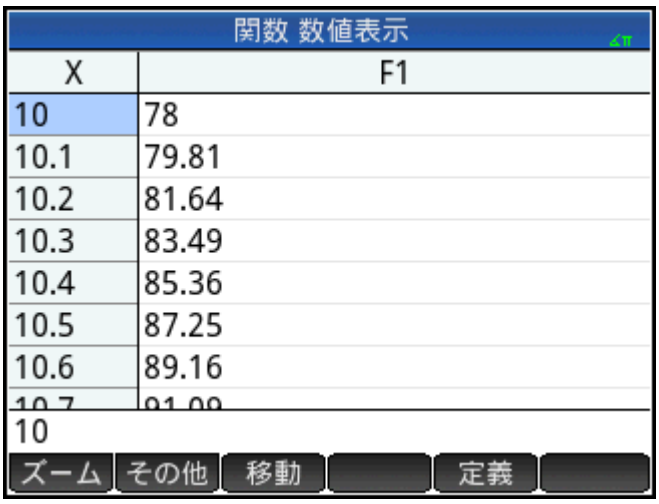


図 6-2 ズーム後



ズーム オプション

[数値表示]では、いくつかのズーム方法を使用できます。

- 上下方向での 2 本指ピンチ ズーム
- キーボード
- [数値表示]の ズーム メニュー

注記：[数値表示]のズーム操作と[プロット表示]のズーム操作は互いに独立しています。ただし、[数値表示]で**[表示]**メニュー（ View Copy ）からズーム オプションを選択した場合は、[プロット表示]でのプロットの表示にそのオプションが反映されます。つまり、**[表示]**メニューのズーム オプションは[プロット表示]にのみ適用されるということです。

[数値表示]でズーム操作をすると、[数値のセットアップ]表示の**[ステップ値]**設定が自動的に変更されます。

ズーム ジェスチャ

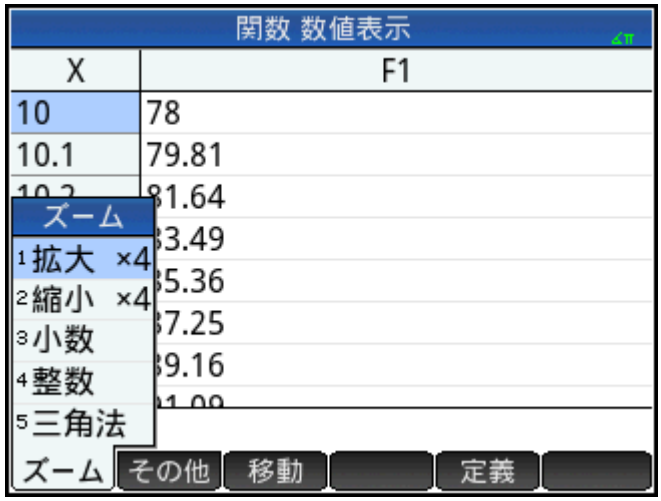
[数値表示]では、2 本指ピンチ ズームを上下方向に実行すると、選択された行が拡大または縮小されます。拡大すると x 値の公差が小さくなり、縮小すると大きくなります。

ズーム キー

ズーム キーは 2 つあります。 [+ Ans ;] を押すと拡大され、 [− Base ;] を押すと縮小されます。拡大や縮小の度合いは、前述のように[ズーム値]設定によって決まります。

[ズーム]メニュー

[数値表示]で ズーム をタップして、オプションをタップします。



ズーム オプションの説明を次の表に示します。

オプション	結果
拡大	独立変数の連続値の増分間隔を、現在の設定値を**[ズーム値]**で割った間隔にします（ショートカット： [+ Ans ;] を押します）
縮小	独立変数の連続値の増分間隔を、現在の設定値に**[ズーム値]**を掛けた間隔にします（ショートカット： [− Base ;] を押します）
小数	**[開始値]**および**[ステップ値]**を初期値に戻します。初期値はそれぞれ 0 と 0.1 です
整数	独立変数の連続値の増分間隔を 1 に設定します
三角法	• 角度単位設定がラジアンの場合は、独立変数の連続値の増分間隔を π/24（約 0.1309）に設定します。 • 角度単位設定が度の場合は、独立変数の連続値の増分間隔を 7.5 に設定します
ズーム解除	表示を直前の**[開始値]**設定値および**[ステップ値]**設定値に戻します。 **注記：**このオプションは、ズーム操作を実行した後でのみ使用できます

評価

[数値表示]の評価表では、 ▲ または ▼ を押して上下の行に移動できます。また、独立変数列で独立変数の値を入力して OK をタップすることで、その評価にすばやく移動することもできます。

たとえば、関数アプリケーションの[記号表示]で F1(X) = (X − 1)² − 3 と定義した場合、X が 625 のときの値を求めるには以下の操作を行います。

1. [数値表示]を開きます（ Num ↳Setup ）。
2. 独立変数列（一番左の列）の任意の行で「625」と入力します。
3. OK をタップします。

[数値表示]が更新されて、入力した値が先頭行に表示され、その右のセルに評価結果が表示されます。この例では、結果は 389373 になります。

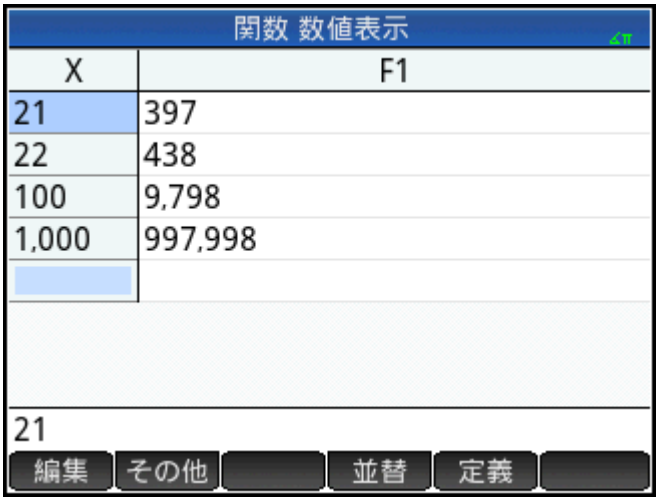
移動 をタップして独立変数の値を入力することもできます。次に、 OK をタップして、新しい値を使用して表を再構成します。

カスタム表

[数値タイプ]設定で**[自動]**を選択した場合、[数値表示]の評価表は、[数値のセットアップ]表示の設定に従って表示されます。この場合、独立変数は**[開始値]**設定の値で始まり、**[ステップ値]**設定の値ずつ増えます（これらの設定については、99 ページの[数値のセットアップ]表示の共通操作を参照してください）。このほか、入力した値だけを独立変数として表示する独自の表を作成することもできます。

1. [数値のセットアップ]表示を開きます（ Shift Num ↳Setup ）。

2. **[数値タイプ]**メニューから**[独自をビルド]**を選択します。



3. [数値表示]を開きます（ Num Setup ）。

   空の表が表示されます。

4. 独立変数列（一番左の列）に、評価する値を入力します。
5. OK をタップします。
6. 他の値も評価する場合は、手順 4 から繰り返します。

データの削除

カスタム表の 1 行分のデータを削除するには、その行にカーソルを置いて Del を押します。

カスタム表のすべてのデータを削除するには、以下の操作を行います。

1. Shift Esc Clear を押します。
2. OK をタップするか Enter ≈ を押して操作を確定します。

[数値表示]でのコピーおよび貼り付け

セルのコピーおよび貼り付け

[数値表示]では、セルの値をコピーして貼り付けることができます。

1. セルをコピーするには、セルをタップして Shift View Copy を押します。
2. セルをボックスまたは他の場所に貼り付けるには、カーソルをその場所に移動して Shift Menu Paste を押します。

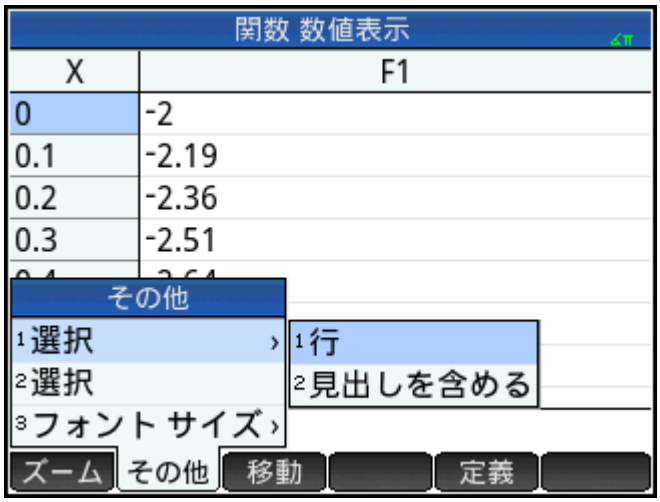
行のコピーおよび貼り付け

[その他]メニューを使用して、行全体を（列見出しを付けてまたは付けずに）コピーして貼り付けることができます。

次の例では、F1(X)=(X − 1)² − 3 に基づく自動的な表を使用します。

表の 2 行目を見出しと一緒にコピーするには、次のようにします。

1. 2 行目をタップします。
2. その他 、**[選択]**、**[見出しを含める]**の順にタップします。



2 行目が見出しと一緒にクリップボードにコピーされます。

行を見出しと一緒にスプレッドシート アプリケーションに貼り付けるには、次のようにします。

1. スプレッドシート アプリケーションを開きます。
2. 貼り付ける行の開始セルをタップします。
3. クリップボードを開くには、 Shift Menu Paste を押します。
4. 行をタップし（この例では最初の項目）、**[グリッド データ]**を選択します。

行は見出しと一緒にスプレッドシートに貼り付けられ、選択されたセルが開始セルになります。

複数のセルのコピーおよび貼り付け

複数のセルを矩形選択してコピーして貼り付けることができます。

1. 隅のセルをタップしたまま指でドラッグして、複数のセルを選択します。
2. すべてのセルを選択した後、 Shift View Copy を押します。
3. 貼り付け場所に移動します。
4. Shift Menu Paste を押します。
5. 矩形選択された要素（この例では最初の項目）をタップし、**[2 次元]**を選択します。

矩形選択された要素が、選択された場所から貼り付けられます。[その他]メニューを使用して、選択するために必要なのがドラッグ ジェスチャのみになるように選択モードを変更することもできます。

[数値表示]：メニュー ボタンの概要

ボタン	用途
ズーム	評価表での独立変数の連続値の増分間隔を変更します。92 ページのズームを参照してください
編集 ([独自をビルド]のみ)	選択した項目を編集可能にするために入力行にコピーします
その他	編集オプションのメニューを表示します。98 ページの[その他]メニューを参照してください
移動	指定したリスト項目にカーソルを移動します
並替 ([独自をビルド]のみ)	データを昇順または降順に並べ替えます
定義	選択した列の定義を表示します

[その他]メニュー

[その他]メニューにはデータのリストを編集するためのオプションが含まれています。オプションの説明を次の表に示します。

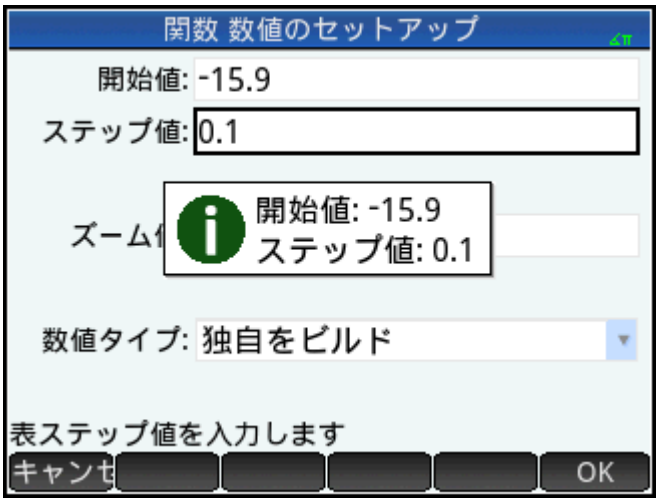
オプション	サブオプション	用途
[挿入] ([独自をビルド]のみ)	[行]	選択されたリストに新しい行を挿入します。新しい行には要素として 0 が含まれます
[削除] ([独自をビルド]のみ)	[列]	選択されたリストの内容を削除します 特定の項目を削除するには、その項目を選択して を押します
[選択]	[行]	現在選択されているセルを含む行を選択します。その後、行全体をコピーできます
	[両端のスワップ]	このオプションは、複数のセルを選択すると表示されます。現在の選択範囲の開始セルおよび終了セルの値を入れ替えます
	[見出しを含める]	現在選択されているセルを含む行およびその行の見出しを選択します。その後、選択範囲全体をコピーできます
[選択]		選択モードのオン/オフを切り替えます 選択モードがオフの場合、セルをタップしたまま指でドラッグして、複数のセルを矩形選択することもできます

オプション	サブオプション	用途
[フォント サイズ]	[小]	小フォントを使用可能にします
	[中]	中フォントを使用可能にします
	[大]	大フォントを使用可能にします

[数値のセットアップ]表示の共通操作

変更するフィールドを選択して新しい値を指定します。[数値表示]で表示する表のタイプ（自動またはカスタム）を選択する場合は、**[数値タイプ]**メニューからオプションを選択します。

開始値および増分値を現在の[プロット表示]に合わせる場合は、 →ヘプ をタップします。



初期設定の復元

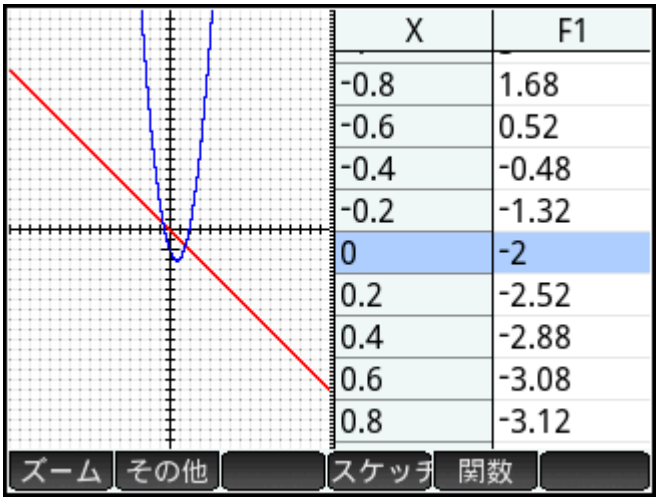
特定のフィールドを初期設定に戻すには、以下の操作を行います。

1. 目的のフィールドを選択します。
2. Del を押します。

すべてのフィールドを初期設定に戻すには、 Shift Esc Clear を押します。

[プロット表示]と[数値表示]の連携

[プロット表示]と[数値表示]を横に並べて表示できます。トレース カーソルを動かすと、[数値表示]の値表がスクロールします。X 列に値を入力することもできます。表がその値までスクロールし、選択されているプロットでトレース カーソルが該当するポイントまで移動します。



- [プロット表示]と[数値表示]を分割画面で連携させるには、 View Copy を押して、**[画面の分割：プロット表]**を選択します。
- [プロット表示]に戻るには、 Num ↳Setup を押します。[数値表示]に戻るには、 Num ↳Setup を押します。

アプリケーションへの注記の追加

アプリケーションに注記を追加できます。注記カタログで作成する通常の注記とは異なり、アプリケーション注記は注記カタログには表示されません。そのアプリケーションを開いているときにのみアクセスできます。

アプリケーション注記は、アプリケーションを他の電卓に送信した場合でも保持されます。

アプリケーションに注記を追加するには、以下の操作を行います。

1. アプリケーションを開きます。
2. Shift Apps Info を押します。

   このアプリケーションにすでに注記が作成されている場合は、その内容が表示されます。
3. 編集 をタップして、注記を入力または編集します。

   注記エディターと同じ書式および箇条書きオプションを使用できます。
4. 注記画面を終了するには、いずれかのキーを押します。注記は自動的に保存されます。

アプリケーションの作成

HP Prime 電卓に付属するアプリケーションは組み込み型であるため削除できません。これらは Apps Info を押すだけでいつでも利用できます。ほとんどのアプリケーションは、カスタマイズしたコピーを作成できます。以前作成したカスタム アプリケーションを基にしてコピーを作成することもできます。カスタム アプリケーションは、組み込みアプリケーションと同様に、アプリケーション ライブラリから開くことができます。

カスタム アプリケーションを作成する利点は、組み込みアプリケーションで他の問題に移った後も、いつでもカスタム アプリケーションに戻って以前のデータを使用できることです。たとえば、数列ア

プリケーションで、フィボナッチ数列を操作するためのカスタム コピーを作成したとします。その後、組み込みの数列アプリケーションで別の数列を作成して操作しても、フィボナッチ数列の操作が必要になったときにはいつでもカスタム数列アプリケーションに戻ることができます。または、解くアプリケーションで、直角三角形の一般的な問題を解くための方程式（たとえば H = O/SIN(θ)、A = H*COS(θ)、O = A*TAN(θ)など）を設定した「Triangles」という名前のカスタム コピーを作成したとします。その後、解くアプリケーションで異なるタイプの問題を解いても、直角三角形の問題が出てきたときには Triangles アプリケーションを使用できます。Triangles を開いて、使用する方程式を選択し（方程式を入力し直す必要はありません）、既知の変数を入力すれば、未知の変数の解を求めることができます。

組み込みアプリケーションと同様に、カスタム アプリケーションも別の HP Prime 電卓に送信できます。また、この章で前述したとおり、カスタム アプリケーションのリセット、削除、および並べ替えもできます。

次のアプリケーションはカスタム コピーを作成できません。

- 1 次エクスプローラー
- 2 次エクスプローラー
- 三角法エクスプローラー

例

組み込みの数列アプリケーションを基にしてカスタム アプリケーションを作成します。カスタム アプリケーションではフィボナッチ数列を生成および操作できるようにします。

1. Apps Info を押し、カーソル キーを使用して数列アプリケーションを選択します。アプリケーションは開かないでください。

2. 保存 をタップします。これにより、組み込みアプリケーションのコピーを作成し、新しい名前で保存できます。組み込みアプリケーション内のデータはすべて保持されるため、後で数列アプリケーションを開いたときにそれらのデータを再度使用できます。

3. [名前]フィールドで、新しいアプリケーションの名前（ここでは例として「Fibonacci」）を入力し、 Enter を 2 回押します。

   新しいアプリケーションがアプリケーション ライブラリに追加されます。アイコンは元の数列アプリケーションと同じですが、名前は自身の付けた名前になります（この例では**[Fibonacci]**）。

4. 以降は、組み込みの数列アプリケーションと同様にこのアプリケーションを使用できます。アプリケーションのアイコンをタップして開きます。元のアプリケーションと同じ表示およびオプションをすべて使用できます。

この例では、カスタマイズ アプリケーションの作成例としてフィボナッチ数列を取り上げました。フィボナッチ数列は、数列アプリケーション内で作成することも、数列アプリケーションを基にしたカスタム アプリケーション内で作成することもできます。

組み込みアプリケーションをコピーする以外に、HP Prime プログラミング言語を使用してカスタム アプリケーションの内部動作を変更することもできます。

アプリケーション関数およびアプリケーション変数

関数

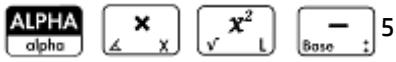
アプリケーション関数は、一般的な計算を実行するために HP アプリケーション内で使用される関数です。たとえば関数アプリケーションでは、[プロット表示]の**[関数]**メニューに、指定した関数の指定した点での傾きを計算する**[傾き]**関数があります。**[傾き]**関数は、ホーム表示やプログラムからも使用できます。

たとえば、$x^2 - 5$ で x = 2 の微分係数を求めるとします。1 つの方法としてアプリケーション関数を使用する場合は、以下の操作を行います。

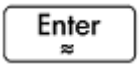
1. Mem を押します。
2. アプリ をタップして、[関数]> **[傾き]**の順に選択します。

   入力行に**[SLOPE()]**が表示され、関数および x 値を入力できる状態になります。

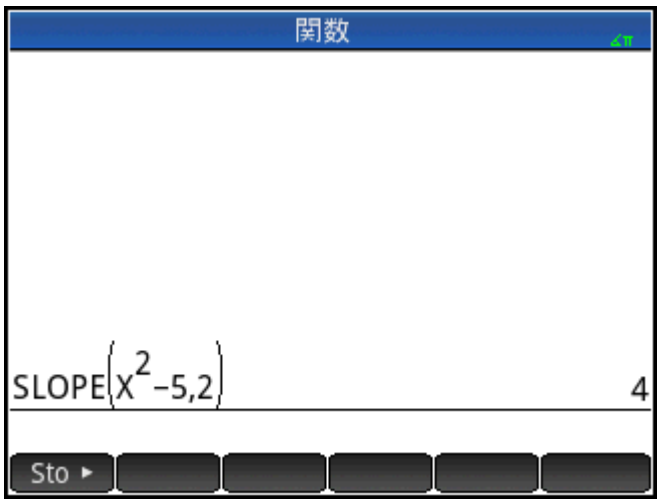
3. 関数を入力します。

4. パラメーターの区切り文字を入力します。

5. x 値を入力して、  を押します。

   x = 2 での傾き（微分係数）が計算されます：4.



変数

すべてのアプリケーションに、アプリケーション固有のさまざまな値を格納するための変数が用意されています。記号式や記号方程式、[プロット表示]や[数値表示]の設定、根や交点などの一部の計算結果のための変数があります。

たとえば、ホーム表示での作業中に、1 変数統計アプリケーションで最近計算したデータ セットの平均値を取得する場合は、以下の操作を行います。

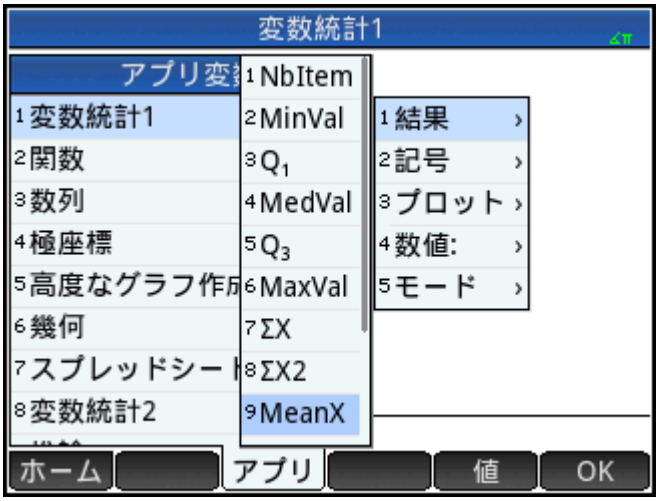
1. を押します。

   [変数]メニューが開きます。ここから、ホーム変数、ユーザー定義変数、およびアプリケーション変数にアクセスできます。

2. アプリ をタップします。

   アプリケーション変数のメニューが開きます。

3. [1 変数統計]> [結果]> **[MeanX]**の順に選択します。



選択した変数の現在値が入力行に表示されます。 Enter を押すと、その値を表示できます。または、作成中の式に変数を含めることもできます。たとえば、1 変数統計アプリケーションで計算した平均値の平方根を求める場合は、最初に Shift を押し、上記の手順 1 ~ 3 を実行してから、 Enter を押します。

変数の修飾

アプリケーション変数名に修飾子を付ければ、HP Prime 電卓内のどこからでもアクセスできます。たとえば、関数アプリケーションとパラメトリックアプリケーションのどちらにも**[Xmin]**という名前の変数があります。最後に使用したアプリケーションがパラメトリックアプリケーションであった場合は、ホーム表示で**[Xmin]**を入力すると、パラメトリック プリケーションの**[Xmin]**の値が取得されます。このとき、関数アプリケーションの**[Xmin]**の値を取得したい場合は、関数アプリケーションを開いてからホーム表示に戻ります。代わりに、変数名にアプリケーション名とピリオドの修飾子を付けて**[Function.Xmin]**と入力することもできます。

7 関数アプリケーション

関数アプリケーションでは、 x の実数値関数 y を最大 10 個まで操作できます。たとえば、$y = 1 - x$ や $y = (x - 1)^2 - 3$ のような関数を定義できます。

関数を定義した後は、以下の操作を実行できます。

- グラフを作成して、根、交点、傾き、符号付き面積、極値を求めます。
- 特定の値での関数の評価を示す表を作成します。

この章では、関数アプリケーションの基本的な関数について例を示しながら説明します。HP Prime 電卓では、より複雑な関数を解くこともできます。

関数アプリケーションでの作業の開始

関数アプリケーションでは、主要なアプリケーション表示をすべて使用できます。[記号表示]、[プロット表示]、[数値表示]の 3 つです。

各表示の標準的なメニュー ボタンもすべて使用できます。

この章では、1 次関数 $y = 1 - x$ および 2 次関数 $y = (x - 1)^2 - 3$ を例に説明します。

関数アプリケーションを開く

▲ 関数アプリケーションを開くには、 Apps Info を押して、**[関数]**を選択します。

アプリケーションのアイコンをタップして開くこともできます。また、カーソル キーを使用してアプリケーションを選択し、 Enter ≈ を押すこともできます。

関数アプリケーションを開くと、最初に[記号表示]が表示されます。これは定義用の表示です。ここで、操作する関数を記号によって定義（指定）します。

[プロット表示]のグラフおよび[数値表示]の数値データは、ここで定義する記号式から計算されます。

式の定義

関数を定義するためのフィールドは 10 個あります。これらには F1(X) ~ F9(X)および F0(X)のラベルが付いています。

1. 使用するフィールドをタップするかそこまでスクロールして選択します。新しい式を入力する場合は、入力を始めます。既存の式を編集する場合は、 編集 をタップして変更を加えます。

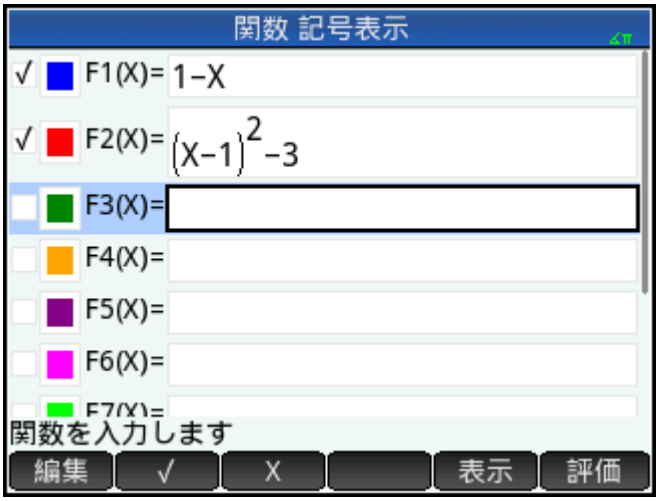
   式の定義または変更が完了したら、 Enter を押します。

2. F1(X)に 1 次関数を入力します。

   1 − xtθn Enter

3. F2(X)に 2 次関数を入力します。

   ( ) xtθn − 1 ▶ x² − 3 Enter



注記：式の入力時は、 X ボタンを使用すると便利です。関数アプリケーションでは、このボタンを押すのは xtθn を押すのと同じになります（他のアプリケーションでは、 xtθn を押すと別の文字が入力されます）。

4. 以下のいずれかの操作を行います。
   - 関数のプロット時の色を変更します。
   - 従属関数を評価します。
   - 操作対象にしない定義の選択を解除します。
   - 変数、数学コマンド、CAS コマンドを定義に組み込みます。

   この例ではこれらの操作は省略しますが、これらは便利な操作です。詳しくは、[記号表示]の共通操作の説明を参照してください。

プロットのセットアップ

x 軸と y 軸の範囲、および軸の目盛の間隔を変更できます。

▲ [プロットのセットアップ]表示を表示します。

Shift Plot Setup

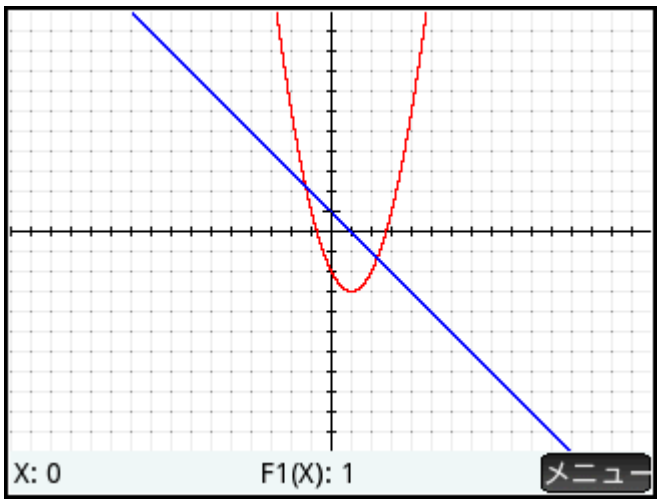
この例では、プロット設定は初期値のままにします。実際の設定が上の図と異なる場合は、Shift Esc Clear を押すと初期値に戻すことができます。

プロットの外観は、[プロットのセットアップ]表示の共通操作を使用して変更できます。

関数のプロット

▲ 関数をプロットします。

Plot Setup

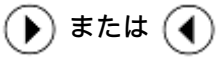


グラフのトレース

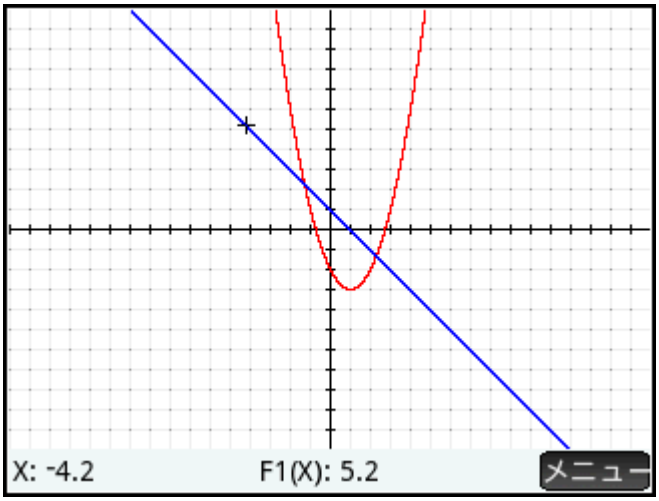
初期設定では、トレース機能が有効になっています。この機能を使用すると、カーソルをグラフに沿って動かすことができます。複数のグラフが表示されている場合は、[記号表示]の関数リストで最

も上にあるグラフが初期設定でトレースされます。この例では 1 次方程式が 2 次方程式よりも上にあるため、初期状態で 1 次方程式のグラフの上にトレース カーソルが表示されます。

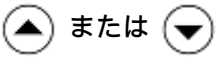
1. 1 次方程式をトレースします。

    または

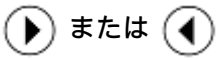
   ボタンを押すと、カーソルがプロットに沿って移動します。また、画面の下部にカーソルの座標が表示され、カーソルの動きに合わせて値が変化します。



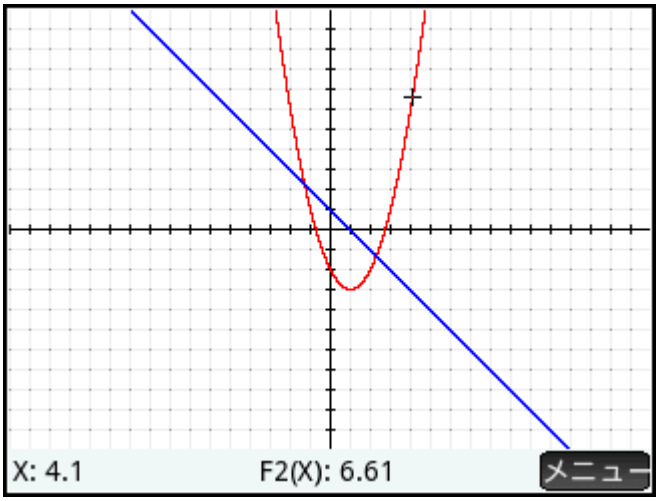
2. トレース カーソルを 1 次方程式から 2 次方程式に移動します。

    または

3. 2 次方程式をトレースします。

    または

   ここでも、画面の下部にカーソルの座標が表示され、カーソルの動きに合わせて値が変化します。



スケールの変更

スケールを変更して、グラフの表示範囲を調整できます。これを行うにはいくつかの方法があります。

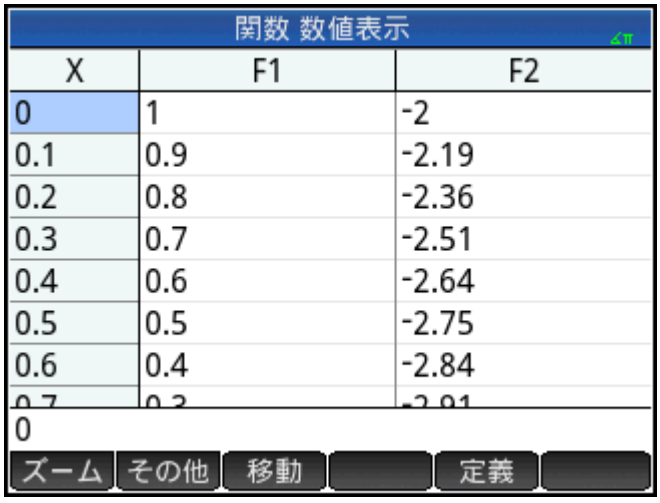
- 2 本指ピンチ ズームを斜め方向に実行すると、x 軸および y 軸が同時に拡大または縮小されます。
- 2 本指ピンチ ズームを左右方向に実行すると、x 軸のみが拡大または縮小されます。
- 2 本指ピンチ ズームを上下方向に実行すると、y 軸のみが拡大または縮小されます。
- 現在のカーソル位置を中心に拡大するときは [+ Ans ;]、縮小するときは [− Base ;] を押します。この操作では、**[ズーム]**メニューで設定されているズーム倍率が使用されます。初期値は x 軸と y 軸のどちらも 2 です。
- [プロットのセットアップ]表示の**[X 範囲]**で x 軸の範囲、**[Y 範囲]**で y 軸の範囲を数値で指定します。
- **[ズーム]**メニューのオプションを使用して、縦軸、横軸、または両方の軸を拡大または縮小します。
- **[表示]**メニュー（ [View Copy] ）のオプションを使用して、あらかじめ定義された表示を選択します。**[自動スケール]**オプションを選択すると、各プロットの重要な特徴ができるだけ多く表示されるように最適なスケールが自動的に判断されます。

注記：設定された x 範囲および y 範囲から外れている部分のプロットをすばやく表示するには、画面を指で水平方向または垂直方向にドラッグします。この方が、軸の範囲を設定し直すよりも簡単です。

[数値表示]の表示

▲ [数値表示]を表示します。

[Num ↳Setup]



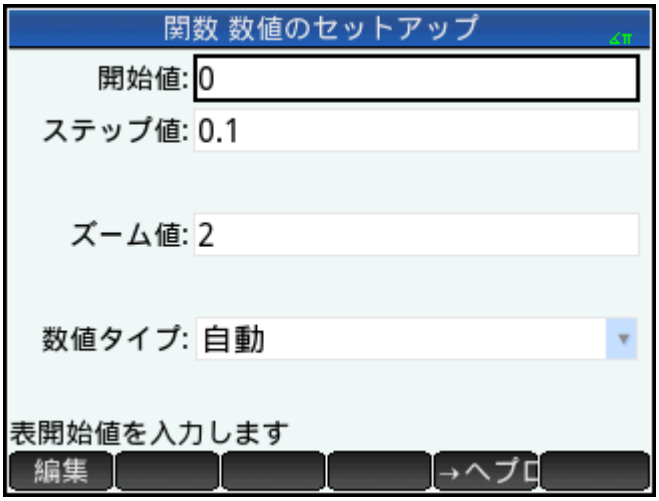
[数値表示]には、[記号表示]で定義した式によって生成されるデータが表示されます。[記号表示]で選択した式ごとに、さまざまな x 値での評価結果が表示されます。

使用可能なボタンについて詳しくは、「*数値表示: メニュー ボタンの概要*」（「*HP アプリケーションの基本*」の章）を参照してください。

[数値表示]のセットアップ

1. [数値のセットアップ]表示を表示します。

   Shift Num Setup



   x 列の開始値とステップ値（増分値）、および表の行の拡大または縮小時のズーム倍率を設定できます。[数値表示]では、ズーム操作をしても表示サイズは変わりません。代わりに、**[ステップ値]**の設定、つまり連続する x 値の増分間隔が変わります。拡大すると増分が小さくなり、縮小すると増分が大きくなります。

   [数値表示]のデータ表が自動的に入力されるようにするか、または特定の x 値を手動で入力するかを選択することもできます。**[数値タイプ]**メニューから**[自動]**または**[独自をビルド]**を選択します。これらはカスタム表のためのオプションです。

2. Shift Esc Clear を押して、すべての設定を初期値に戻します。

3. [数値表示]の X 列の設定（**[開始値]**および**[ステップ値]**）を、[プロット表示]でトレースカーソルが置かれているプロットの x 軸値（Xmin およびピクセル幅）に一致させます。

→ヘプ OK をタップします。

たとえば、[プロット表示]でプロットを拡大して、現在表示されている x 範囲が – 4 ~ 4 になっている場合、このオプションを使用すると**[開始値]**が – 4 に、**[ステップ値]**が 0.025… に設定されます。

[数値表示]の操作

▲ [数値表示]を表示します。

表内の移動

▲ 独立変数列（X 列）内を上下に移動するときは、カーソル キーを使用します。F1 列や F2 列には、[記号表示]で選択した式の x に X 列の値を代入したときの結果が表示されます（この例では 1 – x および$(x - 1)^2 - 3$ の結果）。また、従属変数の列（下の図の F1 列および F2 列）間を左右に移動することもできます。

タップ操作やドラッグ ジェスチャでも表内を縦横に移動できます。

値への直接移動

▲ X 列にカーソルを置いて、目的の値を入力します。たとえば、x = 10 の行に移動するには次のように押します。

10 OK

ズーム オプションへのアクセス

2 本指ピンチ ズーム ジェスチャを使用して、選択した行を拡大または縮小できます。拡大すると増分が小さくなり、縮小すると増分が大きくなります。拡大または縮小する行の値は変化しません。

ズーム倍率のさらに正確な制御を行うには、 + Ans : （または − Base : ）を押してください。このときの拡大や縮小の度合いは、[数値のセットアップ]表示の**[ズーム値]**設定に従います。初期設定値は 4 です。つまり、現在の増分（**[ステップ値]**の値）が 0.4 に設定されている場合、x 値が 10 の行を中心に拡大すると増分間隔が 4 分の 1 になります。たとえば、x 値が 10、10.4、10.8、11.2 と続く場合、操作後は 10、10.1、10.2、10.3、10.4 になります。縮小した場合は、10、10.4、10.8、11.2 が 10、11.6、13.2、14.8、16.4 となります。

また、 ズーム をタップすると、さらに多くのズーム オプションを使用できます。

その他のオプション

[数値表示]のメニューには以下のオプションがあります。

- フォント サイズを変更します。小、中、大から選択できます。
- 関数値の生成元になっている定義を表示します。

[プロット表示]と[数値表示]を連携させることもできます。

関数の分析

[プロット表示]の[関数]メニュー（ 関数 ）を使用して、[関数]アプリで定義した関数の根、交点、傾き、符号付き面積、および極値を求めることができます。関数グラフに接線を追加できます。さらに、関数を指でスケッチし、スケッチを関数グラフに変換して、その式を[記号表示]に保存することもできます。次に[プロット表示]でその関数を平行移動および拡大したり、式を編集したりできます。

[プロット表示]メニューの表示

[関数]メニューは、[プロット表示]メニューのサブメニューです。最初に[プロット表示]メニューを表示します。

Plot ↳Setup メニュー

メニュー ボタンは以下のとおりです。

ボタン	用途
ズーム	拡大および縮小のオプションが含まれる[ズーム]メニューを開きます。
トレース	トレース カーソルの有効/無効を切り替えます。無効にした場合は、カーソルを自由に動かせます。
移動	ジャンプ先の x 値を指定するための入力フォームを表示します。
スケッチ	指で関数をスケッチするためのスケッチ モードを開始します
関数	[関数]メニューを開きます。123 ページの[関数]演算のまとめを参照してください。
メニュー	[プロット表示]メニューの表示/非表示を切り替えます。

関数のスケッチ

関数を指でスケッチし、スケッチを関数のグラフに変換することができます。

スケッチ モードを開始してスケッチを保存するには、以下の操作を行います。

1. [プロット表示]メニューで スケッチ をタップします。
2. メニュー バーに**[関数をスケッチします]**と表示された後、指を使用して次のいずれかの関数タイプをスケッチします。

ヒント： Esc Clear を押すと、いつでも現在のスケッチをキャンセルしてスケッチ モードを終了できます。

- **[1 次]**—m*x + b
- **[2 次]**—a*x^2 + b*x + c
- **[3 次]**—a*x^3 + b*x^2 + c*x + d
- **[指数]**—a*e^(b*x + c)
- **[対数]**—a*LN(x) + b
- **[正弦波]**—a*SIN(b*x + c) + d

3. 電卓表示から指を離すと、スケッチは、リストされた種類のどれかの関数に変換されます。グラフは太線スタイルで表示され、式は表示の左下に表示されます。このグラフおよび式を、[記号表示]内で最初に使用可能な定義（F0 ~ F9）に保存するには、OK をタップします。このグラフおよび式を保存しない場合、新しいスケッチを描画します。既存のスケッチが上書きされます。
4. OK をタップした後、さらに他の関数のスケッチを続行できます。
5. スケッチが終了したら、OK をタップしてスケッチ モードを終了して[プロット表示]に戻ります。

[プロット表示]では、**[定義]**をタップしてスケッチした関数の定義を編集したり、**[変換]**をタップして関数を平行移動および拡大したりできます。

関数グラフの変更

[関数]メニューの[定義]オプションおよび[変換]オプションを使用すると、関数の定義を動的に変換および編集できます。

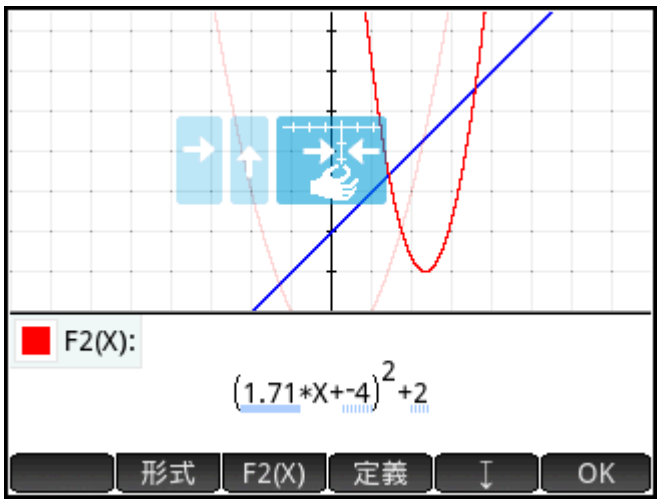
[プロット表示]で選択した関数を編集するには、以下の操作を行います。

1. [関数]メニューで**[定義]**をタップして、エディターを開きます。
2. 以下のオプションのどれかを選択します。
   - 編集—選択した定義の最後にカーソルを移動して、編集できるようにします。式の任意の場所をタップしてカーソルを移動して、式を編集することもできます。式を編集してから OK をタップすると、新しいグラフが表示されます。
   - F1(X)—[記号表示]で現在定義されている関数のリストを開きます。関数を選択して編集できます。

     ヒント：このボタンに表示される数値は、[記号表示]で現在定義されている関数の番号に対応します（1 ~ 9 および 0）。
   - 変換—変換モードを開始します。変換モードでは、関数グラフを直接平行移動および拡大したり、関数定義のパラメーターの変化を確認したりできます。また、[関数]メニューで**[変換]**を選択することもできます。

- ↓—エディターを閉じます。
- メニュー—エディターを閉じて、[プロット表示]メニューを開きます。

3. 変換 を選択した場合、青色の長方形に白い手が表示されます。



垂直方向または水平方向にグラフをドラッグすることはできますが、斜め方向にドラッグすることはできません。関数定義の影響を受けるパラメーターは、平行移動を反映してリアルタイムで変化します。

また、水平方向に 2 本指ピンチ ズームを実行してグラフを拡大することもできます。

グラフに行った変換を記録するときに役立つインジケーターがいくつかあります。

- 明るい青色の長方形は最後の数回の変換を記録します。濃い青色の三角形は現在の変換を示します。
- 影響を受けるパラメーターはすべて青色の下線が引かれます。青色の点線は直前の変換を示します。濃い青色の下線は現在の変換を示します。
- 背景には元のグラフが透過的に表示されます。

変換を実行すると、 形式 ボタンが表示されます。関数定義に別の形式を選択するには、このボタンをタップします。利用可能な形式は、選択した定義によって異なります。

形式 をタップして定義の形式を変更した場合は、 簡略化 ボタンが表示されます。選択した定義を簡略化するには、このボタンをタップします。また、パラメーターの値は小数点以下 1 桁または 2 桁に丸められます。

4. OK をタップすると変更が保存されます。
5. さらに式を編集する必要がある場合は、式または 定義 をタップします。正確な式を入力します。
6. OK をタップすると変更が保存されます。
7. ↓ または メニュー をタップすると、エディターを閉じます。

2 次方程式の根を求める

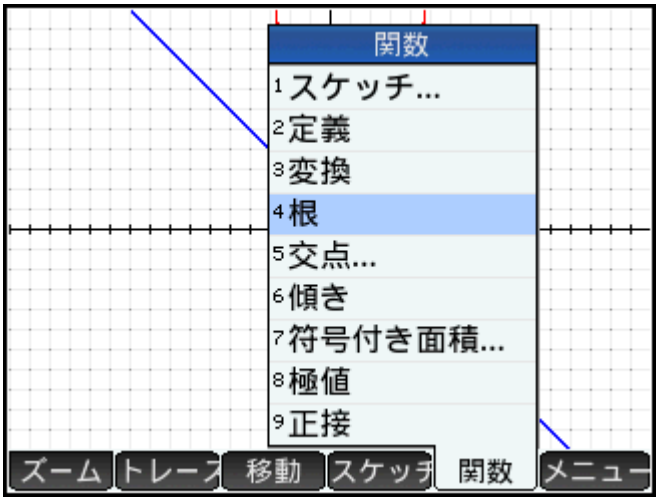
前の例で定義した 2 次方程式の根を求めます。2 次方程式の根は複数ある場合もあるため、求めたい根に近い位置にカーソルを置きます。この例では、x = 3 に近い根を求めます。

1. 2 次方程式をまだ選択していない場合は選択します。

   ▲ または ▼

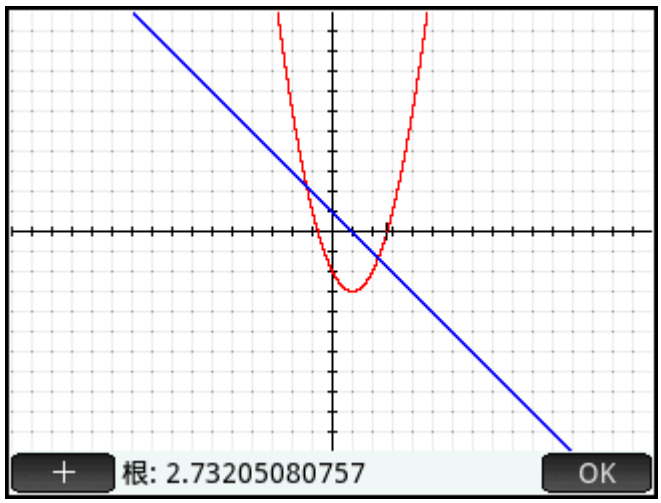
2. ▶ または ◀ を押して、カーソルを x = 3 の近くに移動します。

3. 関数 をタップして、**[根]**を選択します。



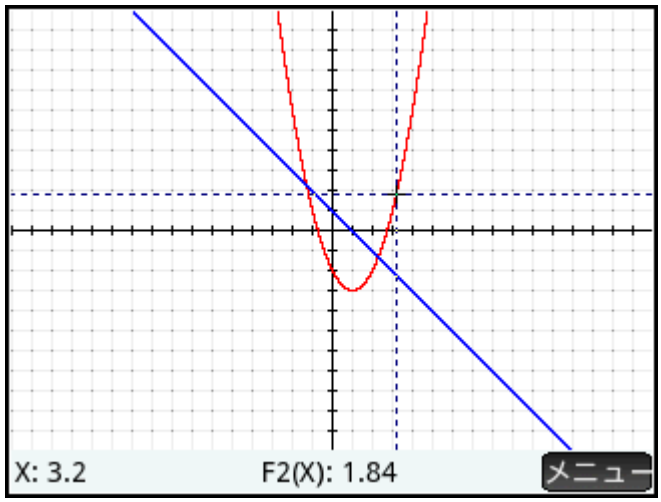
画面の下部に根が表示されます。

この状態でトレース カーソルを x = −1（2 次方程式と x 軸が交わるもう一方の位置）の近くに移動して**[根]**を選択すると、もう 1 つの根が表示されます。



＋ ボタンを使用すると便利です。このボタンをタップすると、トレース カーソルの現在位置を示す縦横の点線が表示されます。これにより、カーソルの位置を常に把握できます。[プロット設定]でカーソルを点滅させるように選択することもできます。**[関数]**メニューに含まれる関数では常に、現在トレース カーソルが置かれている関数が計算対象になり、トレース カーソルの x 座標が初期値と

して使用されます。[プロット表示]内をタップすると、その位置の x 値と現在トレース カーソルが置かれている関数との交点にトレース カーソルが移動します。この方法を使用すれば、トレース カーソルを動かすよりもすばやく目的の点を選択できます。位置を精密に指定したい場合は、カーソル キーを使用してトレース カーソルを動かします。



2 つの関数の交点を求める

2 次方程式の根と同様に、2 つの関数の交点も複数あることがあります。そのため、根を求めるときと同様に、求めたい点に近い位置にカーソルを置きます。この例では、x = −1 に近い交点を求めます。

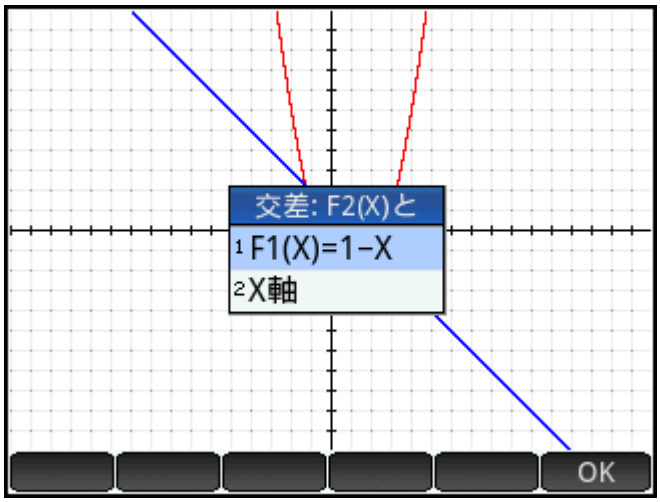
トレース カーソルを特定の点に移動する別の方法として、ここでは**[移動]**コマンドを使用してみます。

1. OK をタップしてメニューを再表示し、 移動 をタップして、 +/− 1 と入力してから、 OK をタップします。

   トレース カーソルが現在の関数の x = 1 の位置に移動します。

2. 関数 をタップして、**[交点]**を選択します。

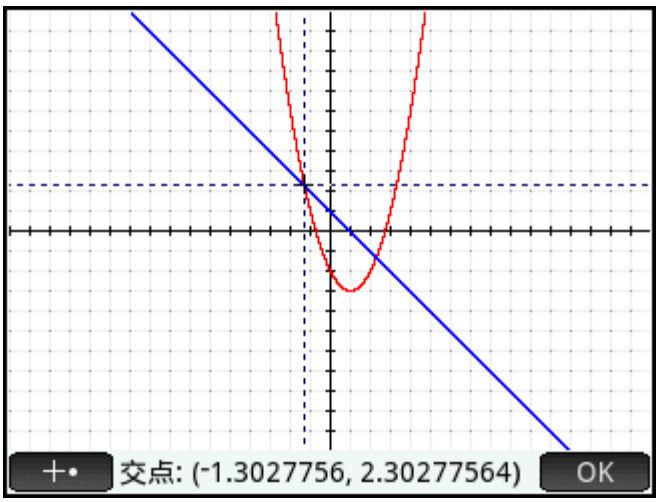
   関数または軸を選択するためのリストが表示されます。



3. 現在選択している関数とどの関数の交点を調べるかを選択します。

画面の下部に交点の座標が表示されます。

画面上の交点付近にある ＋ をタップして、手順 2 を繰り返します。タップした位置に近い交点の座標が画面の下部に表示されます。



2 次方程式の傾きを求める

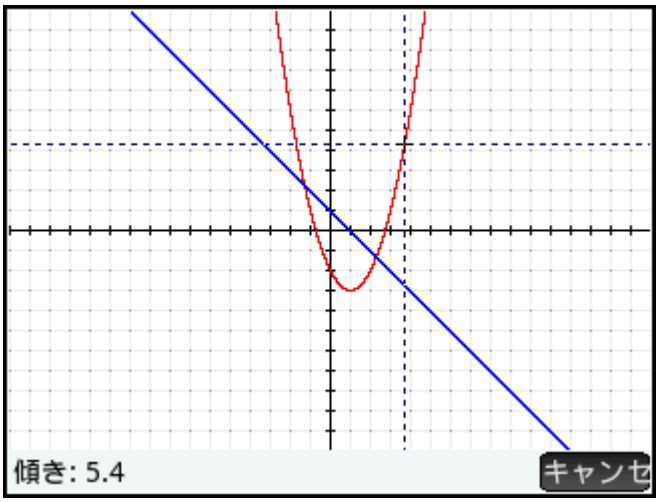
2 次方程式の交点における傾きを求めます。

1. OK をタップしてメニューを表示し、 関数 をタップして、**[傾き]**を選択します。

画面の下部に交点での関数の傾き（勾配）が表示されます。

◀ または ▶ を押してカーブ上を移動すると、他の点での傾きが表示されます。また、

▼ または ▲ を押して別の関数に移動して、各点での傾きを調べることもできます。



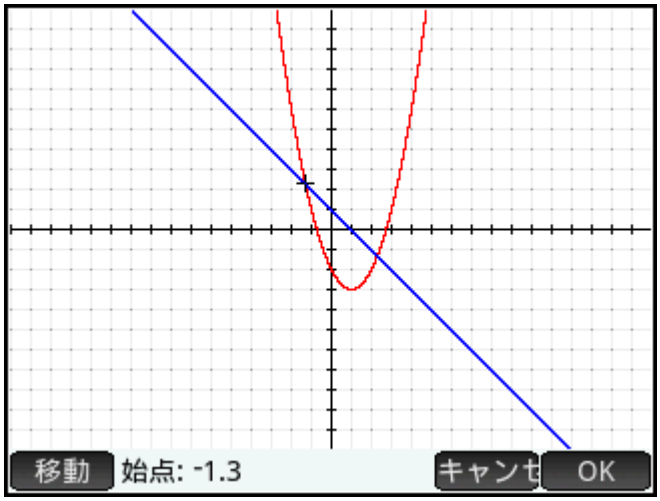
2. キャンセ を押して[プロット表示]メニューを表示します。

2 つの関数間の符号付き面積を求める

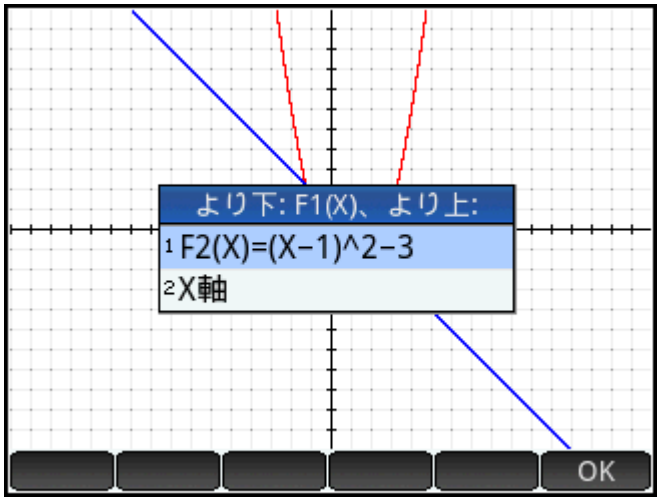
−1.3 ≤ x ≤ 2.3 の範囲における 2 つの関数間の面積を求めます。

1. 関数 をタップして、**[符号付き面積]**を選択します。
2. x の開始値を指定します。

   移動 をタップして、 +/− 1 • 3 Enter と押します。



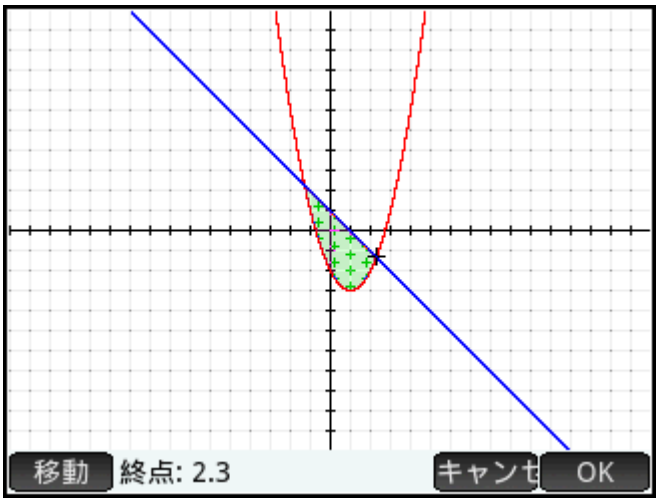
3. OK をタップします。
4. 積分の境界になるもう 1 つの関数を選択します。現在選択している関数が F1(X)の場合は F2(X)、F2(X)の場合は F1(X)を選択します。



5. x の終了値を指定します。

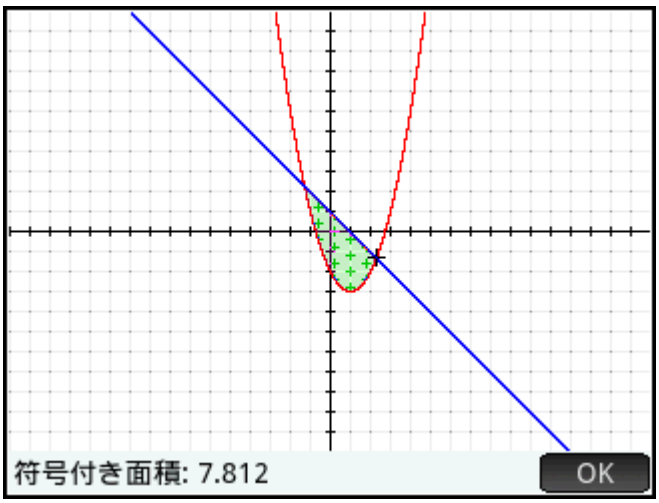
   移動 をタップして、2 [.] 3 [Enter] と押します。

   カーソルが x = 2.3 の位置に移動し、2 つの関数間の領域が影付きで示されます。



6. 積分の数値を表示するには、 OK をタップします。

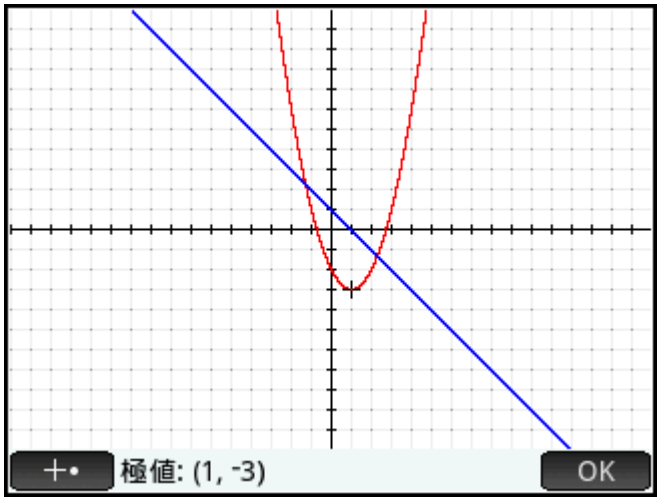
7. OK をタップして[プロット表示]メニューに戻ります。計算対象となる面積の符号は、トレース カーソルが置かれている関数、および終点を左から右または右から左のどちらから入力するかによって決まります。



ヒント : [移動]オプションを使用できるときは、数字を入力するだけで**[移動]**画面を表示できます。入力した数字が入力行に表示されます。 OK をタップして操作を確定します。

2 次方程式の極値を求める

▲ 2 次方程式の極値の座標を求めるには、求めたい極値に近い位置にトレース カーソルを置き（必要な場合）、 関数 をタップして、**[極値]**を選択します。



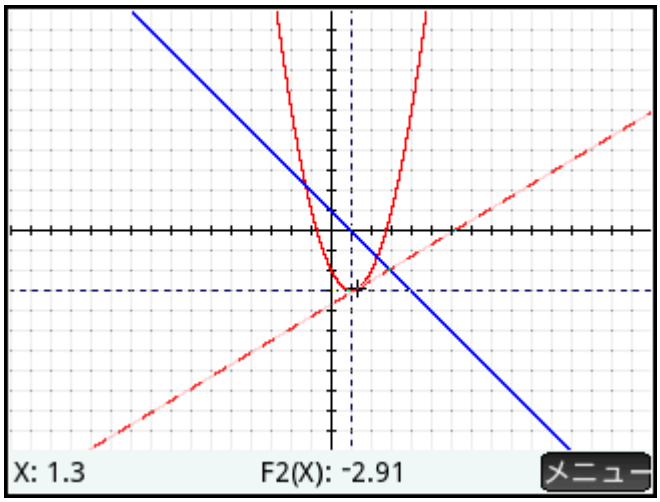
画面の下部に極値の座標が表示されます。

注記：[根]、[交点]、および[極値]演算では、対象の関数に根、交点、または極値が複数ある場合でも、1 つの値しか返されません。カーソル位置に最も近い値のみが返されます。他の値を求めるには、その根、交点、または極値に近い位置にカーソルを移動させる必要があります。

関数への接線の追加

トレース カーソルの位置を通る接線を関数に追加するには、以下の操作を行います。

1. ▲ または ▼ を押して、トレース カーソルを目的の関数に移動します。
2. 関数 をタップしてから、**[正接]**を選択します。トレース カーソルの動きに従って接線が表示されます。このオプションは切り替え式になっています。オプションをもう一度選択すると接線が消えます。



関数の変数

関数アプリケーションでの数値分析の結果はそれぞれ変数に代入されます。これらの変数には以下の名前が付いています。

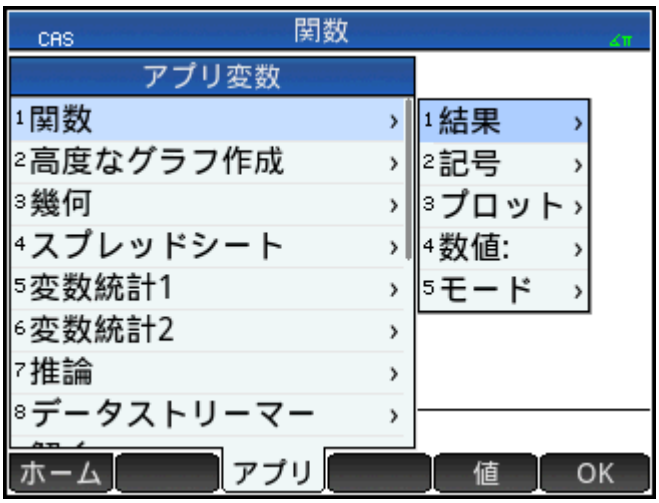
- Root
- Isect（「Intersection（交点）」の略）
- Slope
- SignedArea
- Extremum

分析を行うたびに前の結果が新しい結果で上書きされます。たとえば、2 次方程式の 1 つ目の根を求めた後、2 つ目の根を求めると、Root の値は 1 つ目の根から 2 つ目の根に変わります。

関数変数へのアクセス

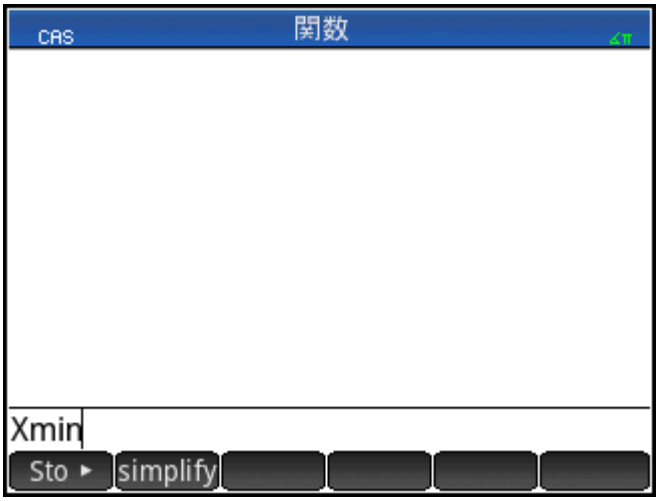
関数変数は、ホーム表示および CAS で計算の引数として使用できます。また、[記号表示]でも使用できます。

1. 変数にアクセスするには、 Vars Chars A を押し、 アプリ をタップして、**[関数]**を選択します。
2. **[結果]**を選択して、目的の変数を選択します。



挿入ポイントに変数名がコピーされます。挿入先の式を評価するときは、変数の値が使用されます。 値 をタップして、変数名の代わりに変数の値を入力することもできます。

たとえば、ホーム表示または CAS で**[変数]**メニューから**[SignedArea]**を選択して、 × 3 Enter ≈ と入力すると、**[SignedArea]**の現在値に 3 を掛けた結果が表示されます。



[記号表示]の関数定義に関数変数を含めることもできます。たとえば、$x^2 - x -$ Root という関数を定義できます。

[関数]演算のまとめ

演算	説明
スケッチ	指で関数をスケッチするためのスケッチ モードを開始します。 **注記：**これは スケッチ と同じ機能です
定義	[プロット表示]で選択した関数定義のエディターを開きます。エディターでは、関数の定義を直接編集したり、グラフを変換したりできます
変換	変換モードを開始します。 変換モードでは、選択した関数を垂直方向または水平方向に平行移動したり、選択した関数を水平方向に拡大したり、関数定義を直接編集したりできます
根	**[根]**を選択すると、現在の関数でトレース カーソルに最も近い位置にある根を求めることができます。カーソルが x 軸上の根の値に移動します。結果の x 値は**[Root]**という名前の変数に保存されます。根が見つからず極値のみが見つかった場合は、**[根]**の代わりに**[極値]**のラベル付きでその結果が表示されます
交点	**[交点]**を選択すると、現在トレース カーソルが置かれているグラフとその他のグラフとの交点を求めることができます。[記号表示]で 2 つ以上の式を選択しておく必要があります。トレース カーソルに最も近い位置にある交点が求められます。その座標が表示され、カーソルが交点に移動します。結果の x 値は**[Isect]**という名前の変数に保存されます
傾き	**[傾き]**を選択すると、現在の関数でトレース カーソルの現在位置での微分係数の表示/非表示が切り替わります。結果は**[Slope]**という名前の変数に保存されます
符号付き面積	**[符号付き面積]**を選択すると、数値積分を求めることができます。複数の式が表示されている場合は、2 つ目の式（x 軸を含む）を選択するためのリストが表示されます。さらに始点および終点を選択します。結果は**[SignedArea]**という名前の変数に保存されます
極値	**[極値]**を選択すると、現在の関数でトレース カーソルに最も近い位置にある最大値または最小値を求めることができます。カーソルが極値に移動し、その座標が表示されます。結果の x 値は**[Extremum]**という名前の変数に保存されます
正接	**[正接]**を選択すると、トレース カーソルの現在位置を通る現在の関数グラフの接線の表示/非表示が切り替わります

導関数または積分による関数定義

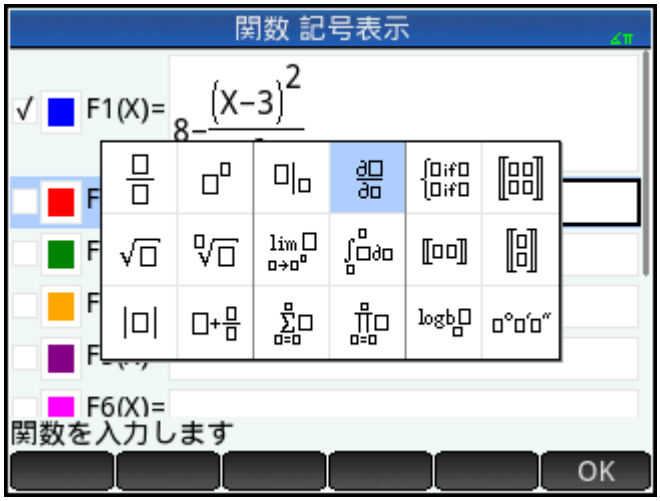
[関数]アプリケーションでは、導関数または積分によって関数を定義できます。このセクションでは、例を示しながらその方法を説明します。

導関数による関数定義

$f(x)=\frac{\delta\left(8-(x-3)^2/6\right)}{\delta x}$ によって定義される関数 f(x)のグラフを作成します。この関数をそのまま入力することもできますが、ここでは F1(X)として関数 $8-\frac{(x-3)^2}{6}$ を定義し、F2(X)にその導関数を定義します。

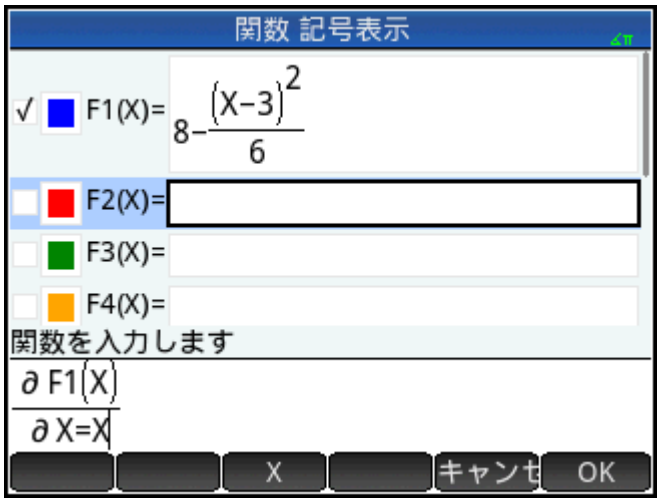
1. Symb Setup を押して[記号表示]に移動します。
2. F1(X)フィールドを選択して、次の図のとおりに関数を入力します。

3. F2(X)フィールドを選択し、 を押してテンプレート メニューを開いてから、導関数のテンプレートを選択します。

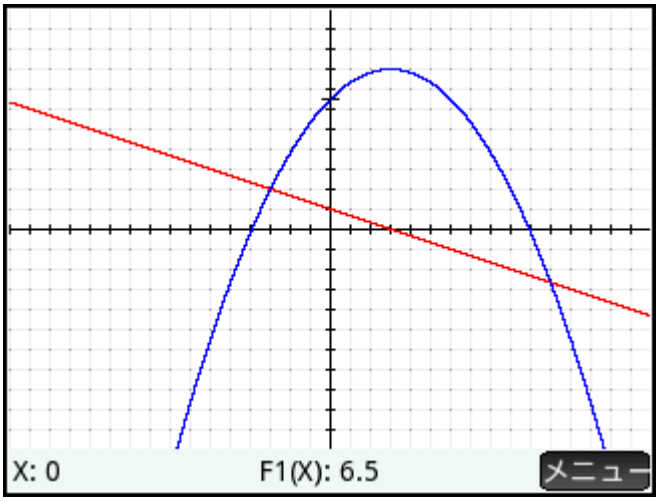


4. 分子に F1(X)を入力します。

5. CAS 以外では、このテンプレートを使用して、関数の特定の点での微分係数を求めることができます。その場合は、分母は X = a の形式で、a に実数を指定します。この例では、形式を示すために、図に示すように分母に X = X を入力します。



6. Plot Setup を押すと、初期設定のウィンドウに関数（青）とその導関数（赤）の両方のグラフが表示されます。



7. Num Setup を押すと、関数とその導関数の値表が表示されます。

関数 数値表示

X	F1	F2
0	6.5	1
0.1	6.59833333333	0.966666666667
0.2	6.69333333333	0.933333333333
0.3	6.785	0.9
0.4	6.87333333333	0.866666666667
0.5	6.95833333333	0.833333333333
0.6	7.04	0.8
0.7	7.11833333333	0.766666666667

0

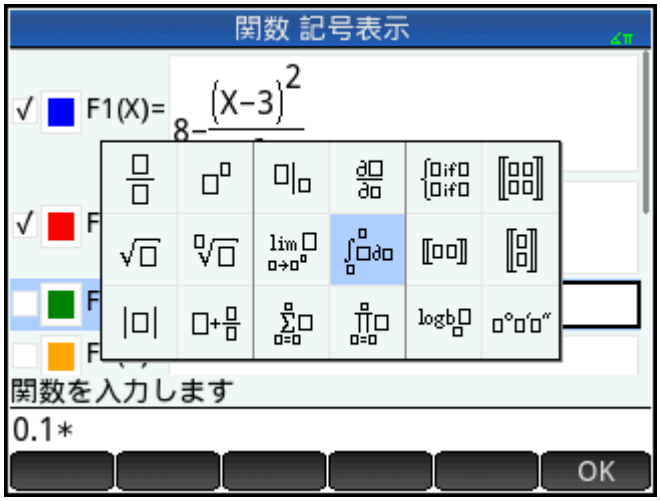
ズーム | その他 | 移動 | | 定義 |

積分による関数定義

引き続き、F3(X)として $0.1 \cdot \int_0^X F1(T)\delta T$ を定義します。

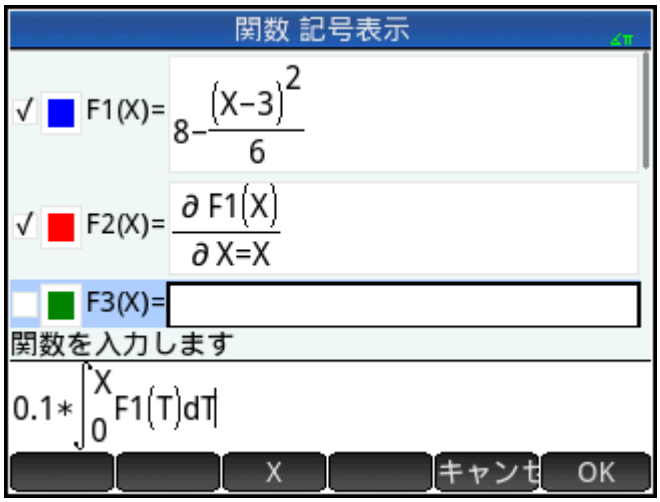
1. [記号表示]に戻り、F3(X)を選択して、`0.1` × と入力します。

2. を押してテンプレート メニューを開き、積分のテンプレートを選択します。

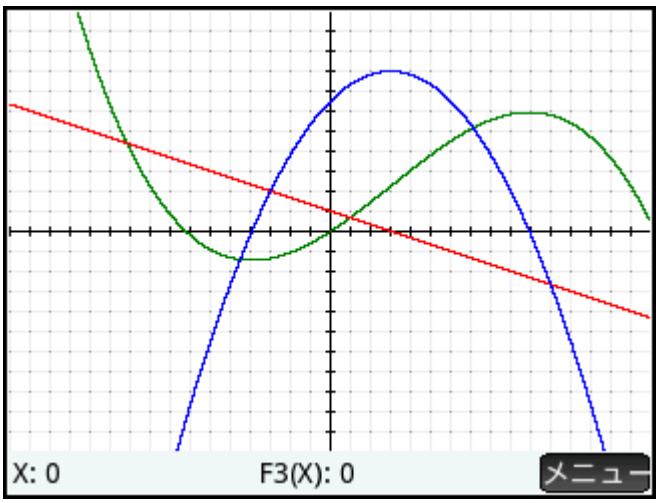


3. 下限に 0、上限に X を入力します。

4. 残りの項は下の図のように入力します。



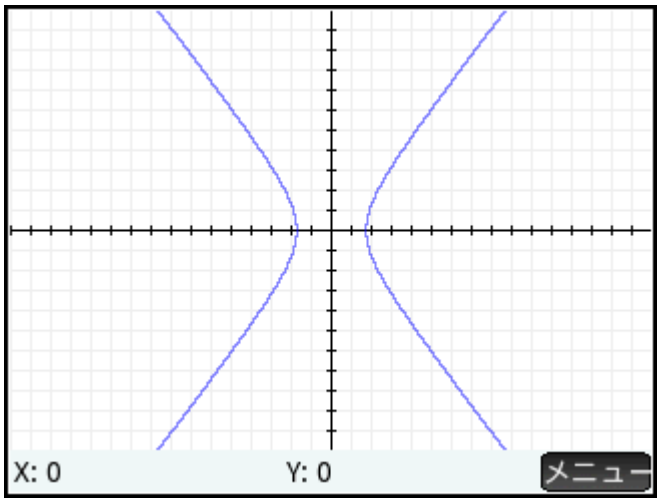
5. Plot Setup を押すと、積分関数のグラフ（緑）が表示されます。



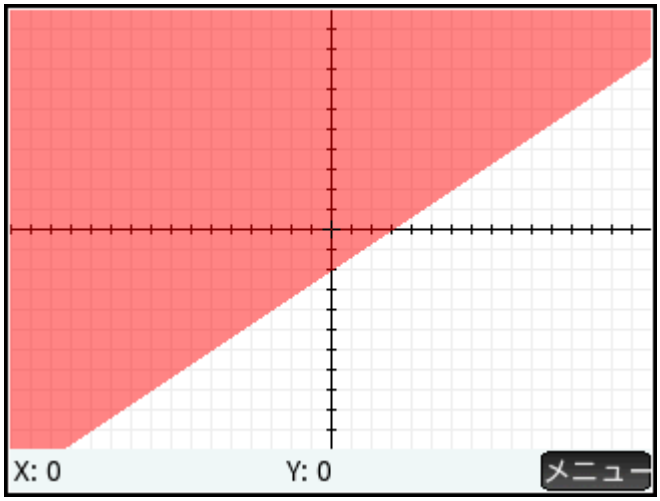
8 高度なグラフ作成アプリケーション

高度なグラフ作成アプリケーションでは、x、y、またはその両方を含む記号開放文やそれらを含まない記号開放文のグラフを定義および操作できます。円錐曲線、標準形式または一般形式の多項式、不等式、および関数をプロットできます。プロットできる開放文の例を以下に示します。

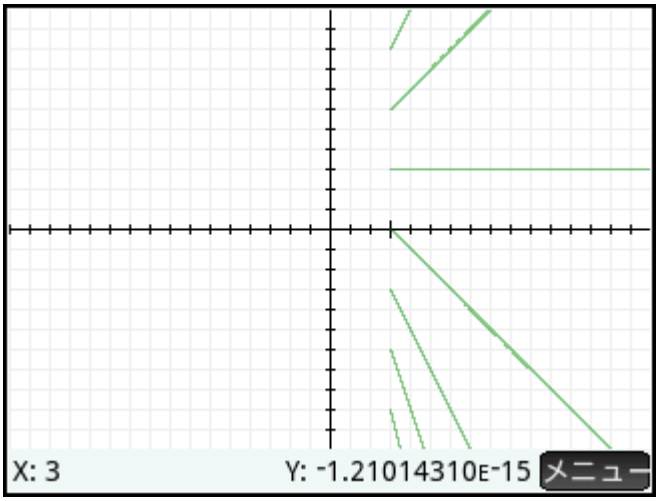
- $x^2/3 - y^2/5 = 1$



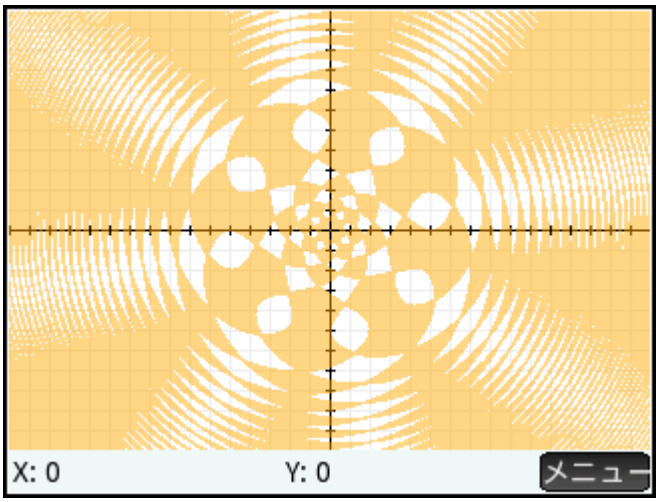
- $2x - 3y \leq 6$



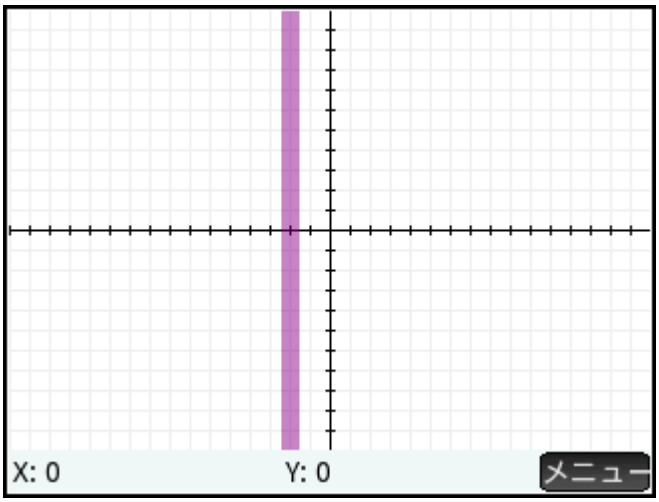
- y mod x = 3



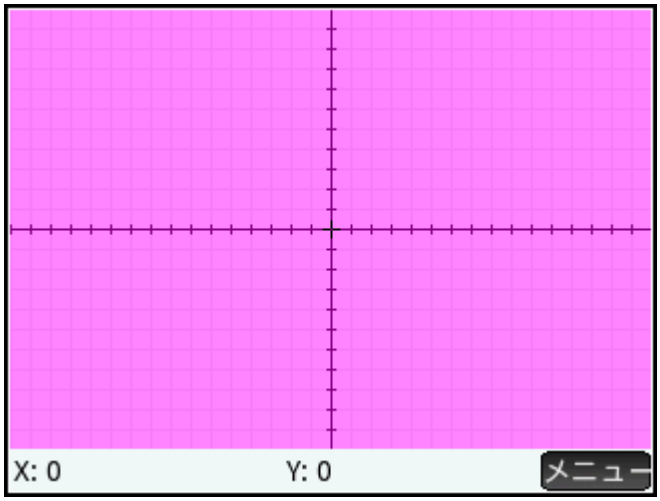
- $\sin\left(\left(\sqrt{x^2+y^2}-5\right)^2\right) > \sin\left(8 \cdot \operatorname{atan}\left(\frac{y}{x}\right)\right)$



- $x^2 + 4x = -4$



- 1 > 0



高度なグラフ作成アプリケーションでの作業の開始

高度なグラフ作成アプリケーションでは、主要なアプリケーション表示をすべて使用できます。[記号表示]、[プロット表示]、[数値表示]の 3 つです。

各表示のボタンもすべて使用できます。

高度なグラフ作成アプリケーションでは、後述するとおり、[トレース]オプションの動作が他のアプリケーションとは異なります。

この章では、次の式によって定義される円錐曲線を例に説明します。

$$\frac{x^2}{2}-\frac{7xy}{10}+\frac{3y^2}{4}-\frac{x}{10}+\frac{y}{5}-10<0$$

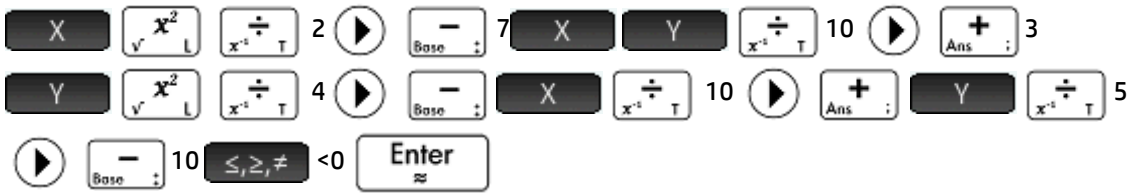
高度なグラフ作成アプリケーションを開く

▲ Apps Info を選択して、**[高度なグラフ作成]**を選択します。

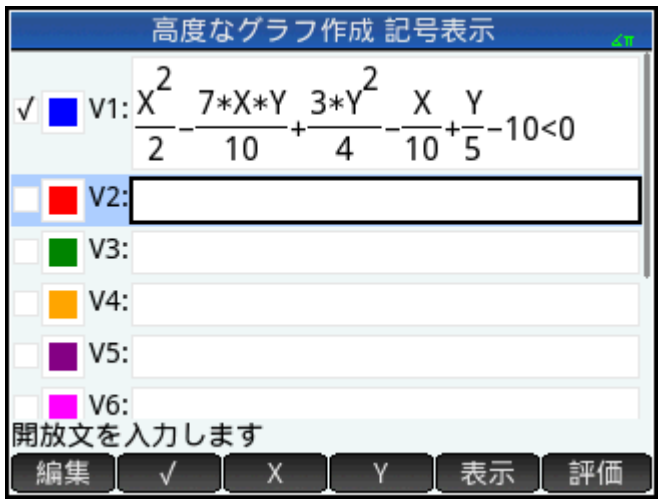
アプリケーションが開き、最初に[記号表示]が表示されます。

開放文の定義

1. 開放文を定義します。



注記： ≤,≥,≠ を押すと関係パレットが表示され、関係演算子を簡単に選択できます。これは、Shift 6 を押したときに表示されるパレットと同じです。



2. 以下のいずれかの操作を行います。
   - 開放文のプロット時の色を変更する
   - 従属関数を評価する
   - 操作対象にしない定義の選択を解除する
   - 変数、数学コマンド、CAS コマンドを定義に組み込む

   この例ではこれらの操作は省略しますが、これらは便利な操作です。詳しくは、[記号表示]の共通操作の説明を参照してください。

プロットのセットアップ

x 軸と y 軸の範囲、および軸の目盛の間隔を変更できます。

▲ [プロットのセットアップ]表示を表示します。

Shift Plot Setup

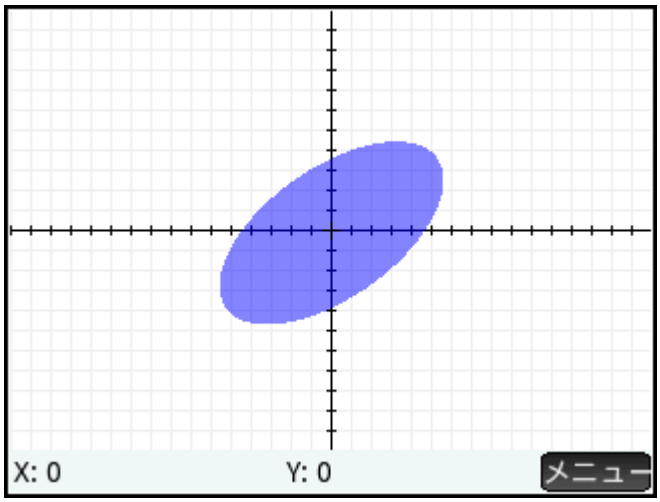
この例では、プロット設定は初期値のままにします。実際の設定が上の図と異なる場合は、 Shift Esc Clear を押すと初期値に戻すことができます。

プロットの外観は、[プロットのセットアップ]表示の共通操作を使用して変更できます。

選択した定義のプロット

▲ 選択した定義をプロットします。

Plot Setup



グラフの操作

1. メニュー をタップして[プロット表示]のメニュー項目を表示します。

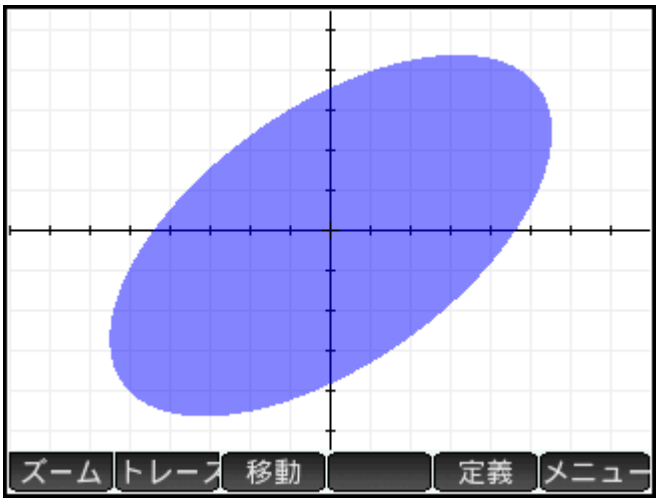
   ズーム、トレース、指定したポイントへの移動、選択したグラフの定義表示のためのオプションがあります。

   ズーム機能や画面分割機能を使用できます。[プロット表示]をスクロールしたり、2 本指ピンチ ズームを使用して拡大と縮小の両方を行ったりすることができます。水平方向のピンチ ズーム

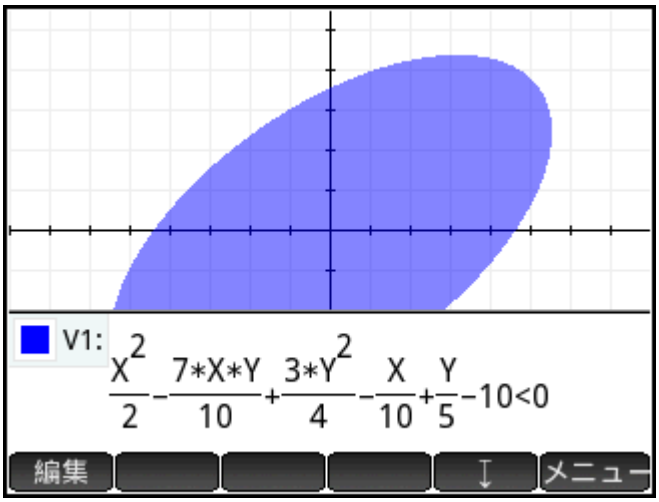
では x 軸のみが拡大または縮小します。垂直方向のピンチ ズームでは y 軸のみが拡大または縮小します。斜め方向のピンチ ズームでは両方が同時に拡大または縮小します。 + Ans ; または − Base ; を押して、カーソルの位置でそれぞれ拡大または縮小することもできます。

2. ズーム をタップして、**[拡大]**を選択します。

高度なグラフ作成アプリケーション独自の機能として、[プロット表示]内でグラフの定義を編集できます。



3. 定義 をタップします。[記号表示]で入力した定義が画面の下部に表示されます。

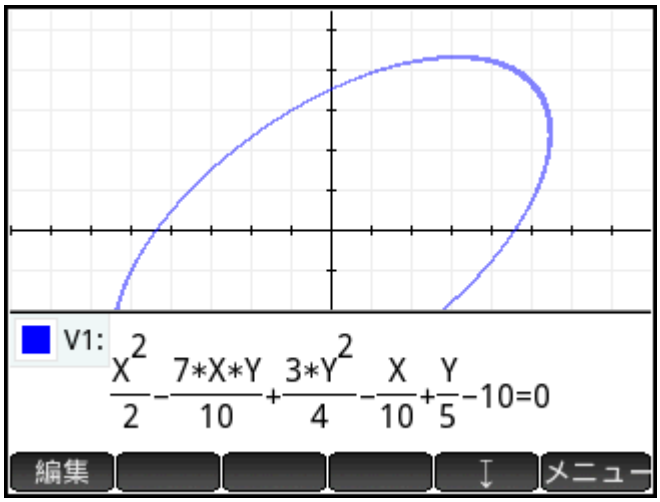


4. 編集 をタップします。

これで定義を編集できる状態になります。

5. 「<」を「=」に変更して、 OK をタップします。

   新しい定義に合わせてグラフが変更されます。[記号表示]の定義も変更されます。

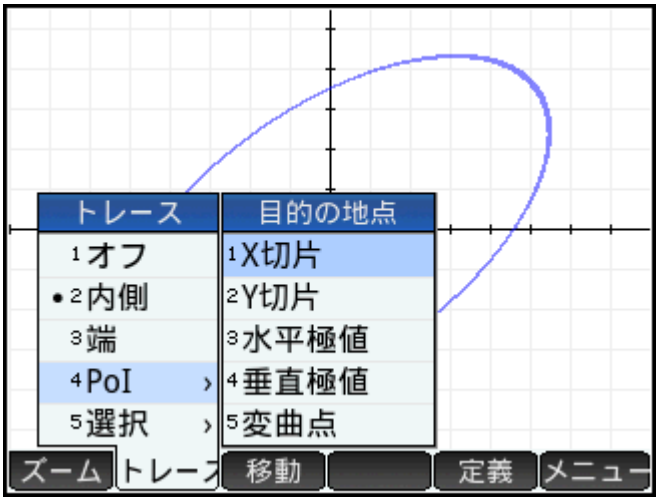


6. ↕ をタップして定義を画面の下部に下げ、グラフ全体が見えるようにします。スペースを縮小するために、定義がテキストブック モードから代数モードに変換されます。

[プロット表示]でのトレース

ほとんどの HP アプリケーションでは、[プロット表示]にトレース機能のオン/オフを切り替えるための トレース があります。高度なグラフ作成アプリケーションでは、[プロット表示]でプロットされる関係が常に関数であるとは限りません。そのため、 トレース は切り替えメニューではなく、トレース カーソルの動作を設定するためのメニューになります。[トレース]メニューには以下のオプションがあります。

- オフ
- 内側
- PoI（「Points of Interest（特定の点）」の略）
  - X 切片
  - Y 切片
  - 水平極値
  - 垂直極値
  - 変曲点



- 選択

トレース カーソルは現在の[プロット表示]ウィンドウの外には移動しません。次の表に、各オプションの簡単な説明を示します。

オプション	説明
オフ	トレース機能をオフにして、カーソルを[プロット表示]内で自由に動かせるようにします
内側	トレース カーソルの動きを現在の関係が真である領域内に制限します。領域内では好きな方向に移動できます。このオプションは不等式などで使用します
端	トレース カーソルの動きを現在の関係の境界のみに制限します(端がある場合)。このオプションは関数や不等式などで使用します
PoI > X 切片	現在のグラフ上で x 切片間を移動します
PoI > Y 切片	現在のグラフ上で y 切片間を移動します
PoI > 水平極値	現在のグラフ上で横方向の極値間を移動します
PoI > 垂直極値	現在のグラフ上で縦方向の極値間を移動します
PoI > 変曲点	現在のグラフ上で変曲点間を移動します
選択	トレースする関係を選択できるメニューが開きます。 および では関係間を移動できないため、このオプションを使用します。高度なグラフ作成アプリケーションでは、4 つのカーソル キーはすべてトレース カーソルを移動するために使用します

[数値表示]

ほとんどの HP アプリケーションでは、[数値表示]の数値表で 2 つの変数の関係を調べることができます。高度なグラフ作成アプリケーションでは、操作対象が関数でない場合もあるため、[数値表示]の機能がかなり異なります。ただし、その目的は変わりません。以下のセクションでは、[数値表示]の独自機能について説明します。

▲ Symb Setup を押して[記号表示]に戻り、V1 に「Y=SIN(X)」と定義します。

注記 : すでに入力されている定義を消去する必要はありません。そのまま新しい定義を入力して OK を押してください。

[数値表示]の表示

▲ Num Setup を押して[数値表示]を表示します。

初期設定では、各行に x 値および y 値が表示されます。その右には、x と y の組み合わせが開放文を満たしているかどうか（[真]または[偽]）を示す列が表示されます。

[数値表示]の操作

▲ X 列にカーソルを置いて新しい値を入力し、OK をタップします。入力した値まで表がスクロールします。

Y 列で値を入力して OK をタップすることもできます。 ▶ および ◀ を押すと列間を移動できます。

[プロット表示]のトレーサーをカスタマイズするために使用可能なものと同じオプションを使用して、表に表示される値をカスタマイズできます。たとえば、x 切片または変曲点のみを表示できます。表示される値は、[プロット表示]で表示される目的の地点に対応します。

また、[ズーム]メニューで利用可能なオプションを使用して、X 変数または Y 変数を拡大または縮小することもできます。[数値表示]では、拡大または縮小を行うと、連続する x 値および y 値の増分が減少または増加します。拡大すると増分が小さくなり、縮小すると増分が大きくなります。このオプションやその他のオプションは[数値表示]の共通操作と同じです。

[数値のセットアップ]表示

[数値表示]に表示される X 値および Y 値は、値を入力したり拡大/縮小したりすることによって操作するほかに、[数値のセットアップ]表示で直接設定することもできます。

▲ [数値のセットアップ]表示を表示します。

Shift Num Setup

X 列および Y 列の開始値とステップ値（増分値）、および表の行の拡大または縮小時のズーム倍率を設定できます。[数値表示]のデータ表が自動的に入力されるようにするか、または特定の x 値および y 値を手動で入力するかを選択することもできます。**[数値タイプ]**メニューから**[自動]**または**[独自をビルド]**を選択します。これらはカスタム表のためのオプションです。

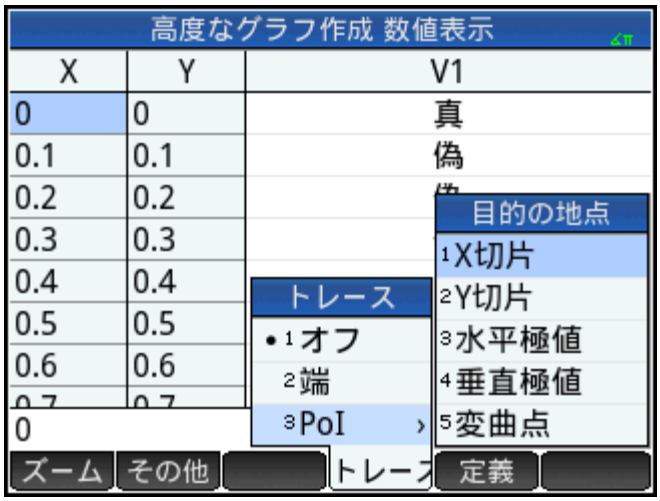
[数値表示]のトレース

[数値表示]では、初期設定の表示内容以外に、[トレース]メニューのオプションを使用できます。[数値表示]のトレース オプションは[プロット表示]のトレース オプションと同じです。これらを使用して、関係の特性を数値表で詳しく調べることができます。以下のいずれかをオプションを選択して表に表示できます。

- 端（X または Y を基準）
- PoI（「Points of Interest（特定の点）」の略）
  - X 切片
  - Y 切片
  - 水平極値

— 垂直極値

— 変曲点

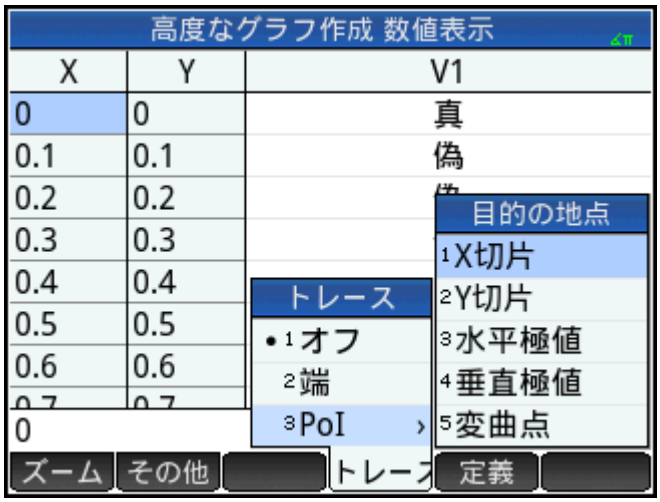


[トレース]オプションで表示される値は、[プロット表示]ウィンドウの内容によって決まります。つまり、表に表示される値は、[プロット表示]で表示範囲内にある点に制限されます。[プロット表示]を拡大または縮小することで、[数値表示]で表に表示される値の範囲を変更できます。

端

1. トレース をタップして、**[端]**を選択します。

   関係が真になる値の組み合わせが表に表示されます（ある場合）。初期設定では先頭が Y 列で、Y 値と組み合わせて関係が真になる X 値が複数ある場合はその後に複数の X 列が続きます。先頭を X 列にしてその後に Y 列を表示する場合は、 X をタップします。次の図では、Y = 0 のとき、Y = SIN(X)が真になる X 値は初期設定の[プロット表示]内に 10 個あることが示されています。これらは表の 1 行目に表示されています。X 値間の差が π になることが明確にわかります。

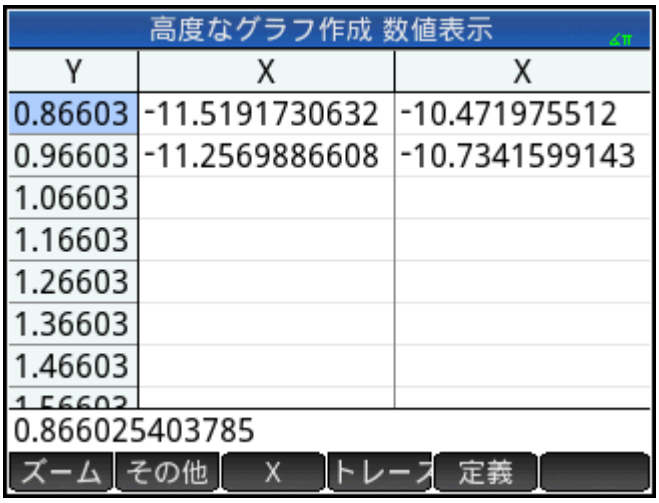


ここでも、Y 値を入力してその行に移動することができます。

2. Y 列で**[0]**が選択されている状態で、 $\frac{\sqrt{3}}{2}$ を入力します。

Shift x^2 3 ÷ 2 Enter

3. 列 をタップして 4 を選択します。



表の 1 行目から、2 系統の解があることが分かります。各系統の連続する解は 2π ずつ離れています。

PoI

1. トレース をタップし、[PoI]を選択して、[垂直極値]を選択します。表に極値が表示されます。

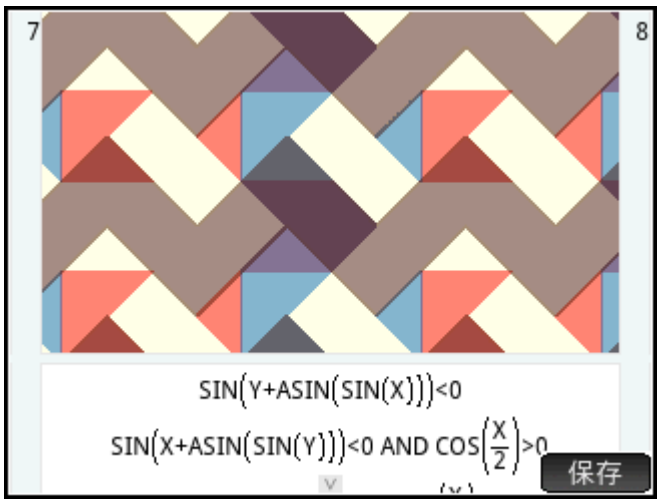
2. 列 をタップし、2 を選択して、表示する列を 2 つだけにします。

[プロット表示]で表示範囲内にある 5 個の極小値と 5 個の極大値が表示されます。

プロット ギャラリー

電卓には、楽しいグラフとその元となる式のギャラリーが用意されています。ギャラリーは[プロット表示]から開きます。

1. [プロット表示]を開き、**[メニュー]**キーを押します。ここでは画面の[メニュー]タッチ ボタンではなく、[メニュー]キーを押すことに注意してください。
2. メニューから**[プロット ギャラリーを表示]**を選択します。ギャラリーの 1 つ目のグラフとその式が表示されます。



3. ▶ を押すと次のグラフが表示されます。ボタンを繰り返し押して順次見て回ることができます。
4. ギャラリーを閉じて[プロット表示]に戻るには、 Plot Setup を押します。

プロット ギャラリーでのプロットの操作

プロット ギャラリーで気になるプロットを見つけたときは、そのコピーを保存できます。コピーは、高度なグラフ作成アプリケーションの新しいカスタム アプリケーションとして保存されます。このカスタム アプリケーションは、組み込みの高度なグラフ作成アプリケーションと同様に変更および操作できます。

プロット ギャラリーのプロットを保存するには、以下の操作を行います。

1. 目的のプロットを表示して、 保存 をタップします。
2. 新しいアプリケーションの名前を入力して、 OK をタップします。
3. OK をもう一度タップします。新しいアプリケーションが開き、[記号表示]にプロットの生成元の式が表示されます。このアプリケーションはアプリケーション ライブラリに追加され、好きなときに開くことができます。

9 幾何

幾何アプリケーションでは、幾何学的作図を描画および操作できます。幾何学的作図は、任意の数の幾何オブジェクト（点、直線、多角形、曲線、接線など）から構成できます。面積や距離などを測定したり、オブジェクトを操作したり、測定値の変化を確認したりできます。

アプリケーション表示は 5 つあります。

- [プロット表示]：描画ツールを使用して幾何オブジェクトを作図できます。
- [記号表示]：[プロット表示]に表示されるオブジェクトの定義を編集できます。
- [数値表示]：[プロット表示]に表示されるオブジェクトの計算を実行できます。
- [プロットのセットアップ]表示：[プロット表示]の外観を変更できます。
- [記号のセットアップ]表示：システム全体の設定の一部を変更できます。

このアプリケーションには[数値のセットアップ]表示はありません。

幾何アプリケーションを開くには、 Apps Info を押して、**[幾何]**を選択します。アプリケーションを開くと、最初に[プロット表示]が表示されます。

幾何アプリケーションでの作業の開始

以下の例では、曲線の導関数をグラフで表現する方法、および曲線に沿って接点を動かしたときに微分係数を自動的に更新する方法について説明します。対象となる曲線は y = 3sin(x)です。

この例では計算の精度はあまり重要でないため、最初に数値形式を変更して小数点以下第 3 位までに固定します。この方が幾何アプリケーションの作業空間がすっきりします。

準備

1. Shift CAS Settings を押します。
2. **[CAS 設定]**の 1 ページ目で、数値形式を**[標準]**に設定し、小数位の桁数を**[4]**に設定します。

アプリケーションを開く/グラフのプロット

1. Apps Info を押して、**[幾何]**を選択します。

   不要なオブジェクトが表示されている場合は、 Shift Esc Clear を押し、 OK をタップして操作を確定します。

   アプリケーションを開くと、最初に[プロット表示]が表示されます。この表示にはデカルト平面と、下部にメニュー バーが表示されます。メニュー バーの隣にはカーソルの座標が表示されます。アプリケーションで操作を行うと、画面の下部に、現在アクティブなツールまたはコマン

ド、そのヘルプ、および現在ポインターが置かれていると認識されているすべてのオブジェクトのリストが表示されます。

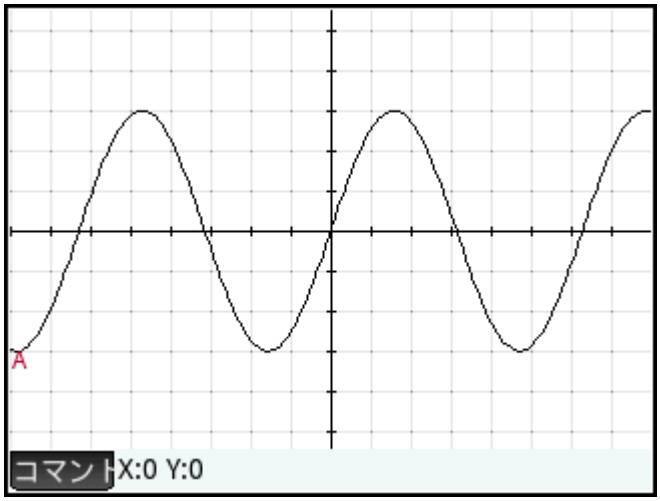
2. プロットするグラフのタイプを選択します。この例では、単純な正弦関数をプロットします。

   コマンド > [プロット]> **[関数]**

3. 入力行に `plotfunc(` と入力されるので、続けて 3*sin(x)を入力します。

   3 [×] [SIN ASIN G] [ALPHA alpha] [Shift] [×] [Enter ≈]

   幾何アプリケーションでは x を小文字で入力する必要があります。



グラフが上の図のようにならない場合は、[プロットのセットアップ]表示（ [Shift] [Plot Setup] ）で **[X 範囲]**および**[Y 範囲]**を調整してください。

次に、位置が曲線上に制限される点を追加します。

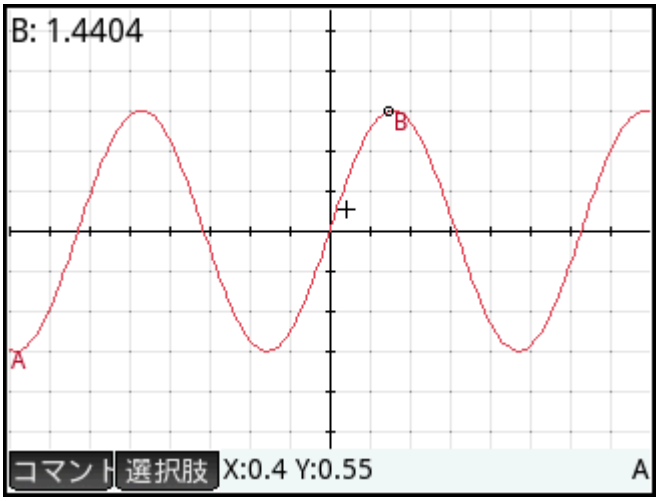
制限される点の追加

1. コマンドをタップし、**[点]**をタップして、**[点オン]**を選択します。

   [点]ではなく**[点オン]**を選択すると、点の動きが追加先のオブジェクト内に制限されます。

2. グラフ上の任意の位置をタップし、 Enter ≈ を押して、 Esc Clear を押します。

グラフに名前付きの点（この例では点**[B]**）が追加されます。画面の空白領域をタップして、すべての選択を解除します（選択されているオブジェクトは青で表示されます）。



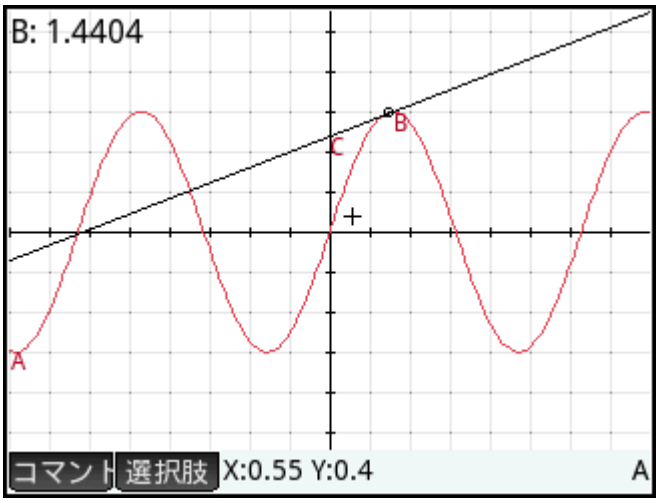
接線の追加

1. 次に、曲線に、点**[B]**を通る接線を追加します。

コマント > 線> **[正接]**

2. 曲線の選択を求められたら、曲線をタップして Enter ≈ を押します。点の選択を求められたら、点**[B]**をタップして Enter ≈ を押します。これで接線が表示されます。 Esc Clear を押して接線ツールを閉じます。

点**[B]**の位置によっては、グラフが次の図とは異なることがあります。次に、接線を明るい色にして目立たせます。



3. 接線をタップして選択します。接線を選択すると、新しいメニュー キー 選択肢 が表示されます。 選択肢 をタップするか Menu Paste を押して、**[色の選択]**を選択します。

4. 色を選択してから、画面の空白領域をタップします。これで接線が新しい色で表示されます。

5. 点**[B]**をタップし、曲線に沿ってドラッグします。それに従って接線も移動します。接線自体をドラッグすることもできます。

6. 点**[B]**をタップし、 Enter ≈ を押して点を選択します。点の色が、選択されたことを示す青に変わります。この状態にすると、点**[B]**を指でドラッグして動かすことも、カーソル キーを使用して位置を精密に制御しながら動かすこともできるようになります。点**[B]**の選択を解除するには、 Esc Clear を押すか、点**[B]**をタップして Enter ≈ を押します。

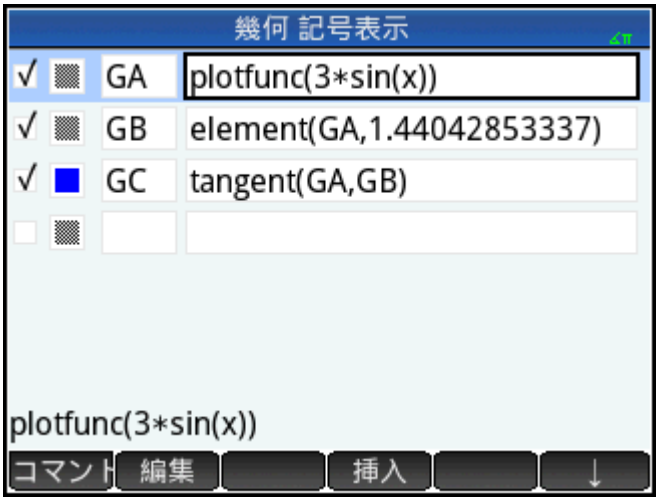
どのような操作を行おうと、点**[B]**は曲線から離れることはありません。また、点**[B]**を動かすと接線も一緒に移動します。点が画面の外に出てしまった場合は、画面を指で適切な方向にドラッグして画面内に戻すことができます。

微分係数の点の作成

グラフ上のある点の微分係数は、その点における接線の傾きを示します。そこで、点**[B]**の制約を受け、縦座標の値が点**[B]**の微分係数に一致する新しい点を作成します。*x* 座標（横座標）が常に点**[B]**の x 座標に一致し、*y* 座標（縦座標）が常にその点の接線の傾きに等しくなるように制約を設定します。

1. 他の幾何オブジェクトの属性に従属する点を定義するため、 Symb x Setup を押して[記号表示]に移動します。

   [記号表示]には、これまでに作成したすべてのオブジェクトが表示されます。[記号表示]でのオブジェクト名は、[プロット表示]でのオブジェクト名の先頭に「G」を付けたものになります。たとえば、[プロット表示]で**[A]**という名前のグラフは、[記号表示]では**[GA]**になります。



2. **[GC]**の定義の後の空欄を選択して、 新しい をタップします。

   他のオブジェクトに従属するオブジェクトを作成するときは、[記号表示]での順序が重要になります。各オブジェクトは、[記号表示]で定義されている順序で[プロット表示]に描画されます。ここでは**[GB]**および**[GC]**の属性に従属する点を作成するので、その定義は**[GB]**および**[GC]**の後に置く必要があります。そのため、 新しい をタップする前に、既存の定義の下にある空欄を選択しています。それよりも上に新しい定義を作成すると、次の手順で作成する点が[プロット表示]で無効になります。

3. コマンド をタップして、[点]> **[点]**の順に選択します。

   ここで、新しい点の *x* 座標および *y* 座標を指定する必要があります。x 座標は点**[B]**（[記号表示]では**[GB]**）の横軸として定義し、y 座標は接線**[C]**（[記号表示]では**[GC]**）の傾きとして定義します。

4. 入力行に `point()` と入力されるので、かっこ内に次のように入力します。

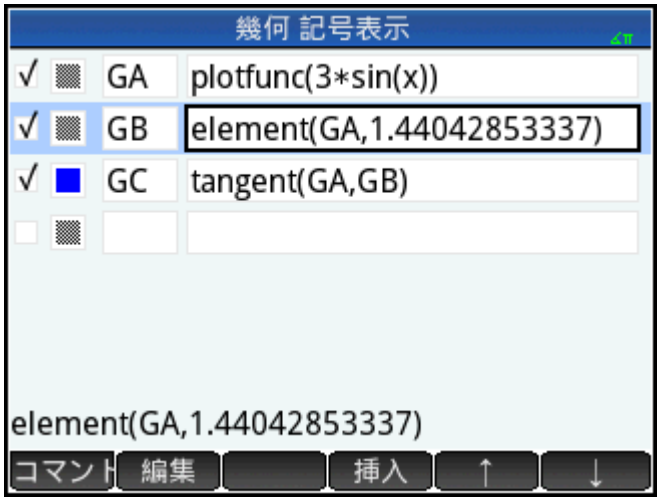
   ```
   abscissa(GB),slope(GC)
   ```

   abscissa コマンドを入力するには、 Mem 8 を押して、 カタログ をタップします。 Vars Chars A を押して A から始まるコマンドに移動し、**[abscissa]**までスクロールして OK をタップします。

   slope コマンドを入力するには、 Mem 8 を押して、 カタログ をタップします。 9 !,∞,→ 5 を押して S から始まるコマンドに移動し、**[slope]**までスクロールして OK をタップします。もちろん、コマンド名を 1 文字ずつ入力してもかまいません。最初に小文字入力にロックするために ALPHA alpha Shift ALPHA alpha を押します。入力後は ALPHA alpha をもう一度押してロックを解除します。

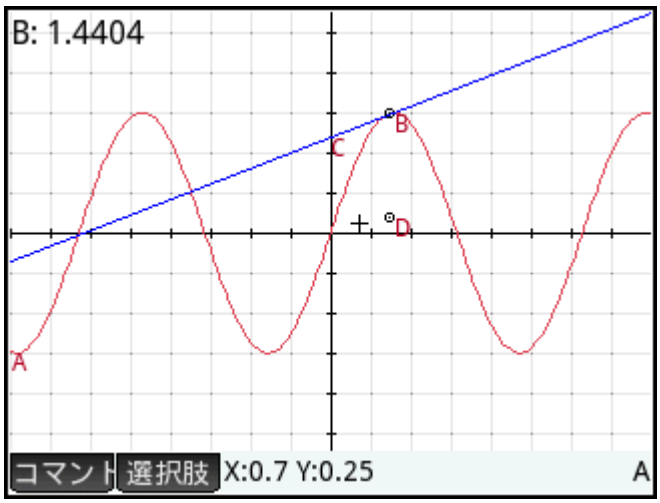
5. OK をタップします。

[記号表示]に新しい点の定義が追加されます。[プロット表示]に戻ると、**[D]**という名前の点が追加され、その *x* 座標が点**[B]**と一致していることが分かります。



6. Plot Setup を押します。

点**[D]**が見えない場合は、見えるまで画面をスクロールします。点**[D]**の *y* 座標は、点**[B]**での曲線の微分係数と一致します。



画面から座標を読み取ることは難しいため、正確な微分係数（小数点以下第 3 位まで）の計算を追加して[プロット表示]に表示します。

計算の追加

1. Num Setup を押します。

計算は[数値表示]で入力します。

2. 新しい をタップします。

3. コマンド をタップして、[測定]> **[傾き]**の順に選択します。

4. かっこ内に接線の名前「GC」を追加して、 OK をタップします。

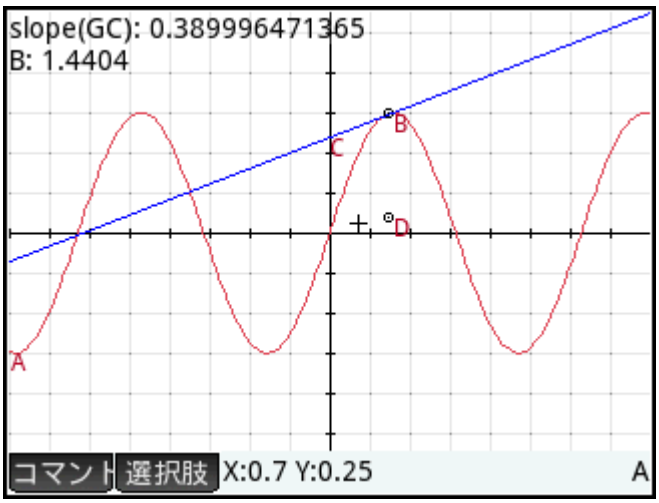
   現在の傾きが計算されて表示されます。この値は動的であり、[プロット表示]で接線の傾きが変化すると、[数値表示]で傾きの値が自動的に更新されます。

5. [数値表示]で新しい計算が選択されている状態で、 √ をタップします。

   [数値表示]で選択された計算は、[プロット表示]に表示されます。

6. Plot Setup を押して[プロット表示]に戻ります。

   [数値表示]で作成した計算が画面の左上に表示されます。



   続けて、[数値表示]に計算をさらに 2 つ追加して[プロット表示]に表示します。

7. Num Setup を押して[数値表示]に戻ります。

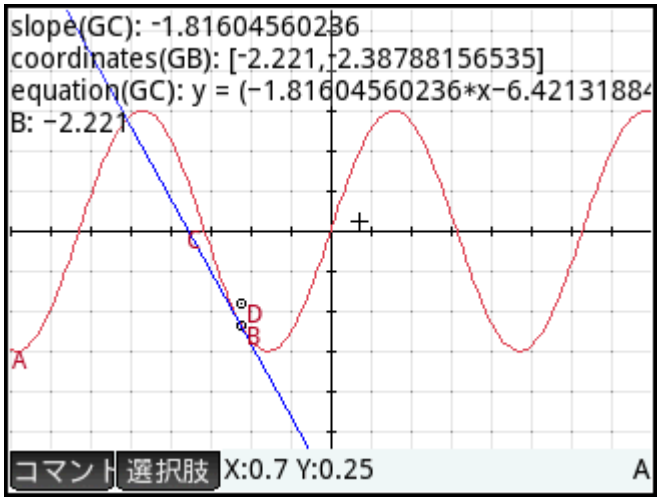
8. 下にある空欄のフィールドをタップして選択し、 新しい をタップして新しい計算の入力を開始します。 コマント をタップし、**[デカルト]**を選択して、**[座標]**を選択します。かっこ内に「GB」と入力して、 OK をタップします。

9. 3 つ目の計算を追加します。 コマント をタップし、**[デカルト]**を選択して、**[方程式]**を選択します。かっこ内に「GC」と入力して、 OK をタップします。

10. 追加した新しい式を両方とも選択します（両方を選択して √ を押します）。

11. Plot Setup を押して[プロット表示]に戻ります。

新しい計算が表示されます。



12. 点**[B]**をタップし、 Enter ≈ を押して点を選択します。

13. カーソル キーを使用して、点**[B]**をグラフに沿って動かします。動かすたびに、画面の左上に表示されている計算結果が変化します。点**[B]**の選択を解除するには、点**[B]**をタップして Enter ≈ を押します。

[プロット表示]での計算

初期設定では、[プロット表示]での計算は画面の左上に固定されます。計算をドラッグして固定を解除すれば、画面内の好きな場所に移動できます。ただし、固定を解除すると、計算が表示とともにスクロールします。ラベルを編集するには、計算をタップして保持します。編集行が表示され、独自のラベルを入力できます。 選択 をタップして、計算の色およびラベルを変更することもできます。完了したら OK をタップします。

微分係数のトレース

点**[D]**の縦座標は、曲線における点**[B]**での微分係数に一致します。この点のプロットを見れば、計算して結果を比較するよりも微分係数の変化を簡単に把握できます。プロットを表示するには、点**[B]**の動きに対する点**[D]**の動きをトレースします。

最初に、トレース曲線が見やすくなるように計算を非表示にします。

1. Num Setup を押して[数値表示]に戻ります。

2. 各計算を順に選択して √ をタップします。すべての計算の選択が解除されます。

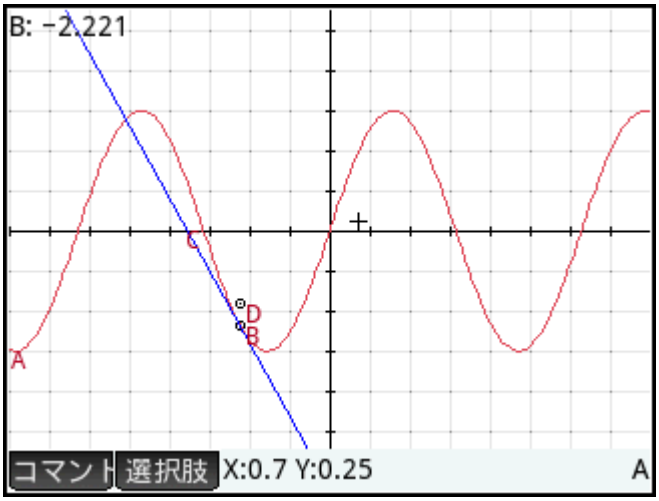
3. Plot Setup を押して[プロット表示]に戻ります。

4. 点**[D]**をタップし、 Enter ≈ を押して点を選択します。

5. 選択肢 をタップするか Menu Paste を押して、**[トレース]**を選択します。 Enter ≈ を押して点**[D]**の選択を解除します。

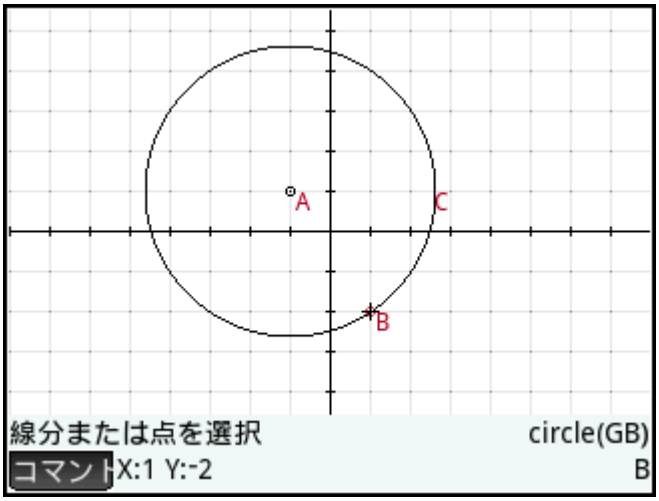
6. 点**[B]**をタップし、 Enter ≈ を押して点を選択します。

7. カーソル キーを使用して、点**[B]**をグラフに沿って動かします。点**[B]**の動きに合わせて点線の曲線がトレースされます。これが 3sin(x)の導関数の曲線です。点**[B]**をタップし、 Enter ≈ を押して点の選択を解除します。



[プロット表示]の詳細

[プロット表示]では、さまざまな描画ツールを使用して、画面上にオブジェクトを直接描画できます。たとえば円を描画するには、コマント をタップして**[曲線]**をタップし、**[円]**を選択します。続いて、円の中心になる位置をタップして、 Enter ≈ を押します。次に、円周になる位置をタップして、 Enter ≈ を押します。最初にタップした位置を中心とし、その位置と次にタップした位置間の距離を半径とする円が描画されます。



作業中は画面に説明が表示されます。これらの説明は画面の下部付近、アクティブなツール（円、点など）のコマンド リストの横に表示されます。

[プロット表示]では幾何オブジェクトを好きなだけ描画できます。描画できるオブジェクトのタイプについては、163 ページの[プロット表示] : [コマンド]メニューを参照してください。選択した描画ツール（直線、円、六角形など）は、選択を解除するまで選択されたままになります。そのため、同じタイプのオブジェクトをいくつか描画するとき（六角形をいくつか描画するときなど）にすばやく作業できます。特定のタイプのオブジェクトを描画し終えたら、 Esc Clear を押してその描画ツールの選択を解除します。描画ツールがまだ選択されているかどうかは、画面の下部に説明およびコマンド名が表示されているかどうかで判断できます。

[プロット表示]内のオブジェクトはさまざまな方法で操作でき、その数学的特性を簡単に調べることができます（161 ページの全オブジェクトのリストを参照）。

オブジェクトの選択

オブジェクトは少なくとも 2 つの手順で選択します。オブジェクトをタップして、 Enter ≈ を押します。 Enter ≈ を押すのは、オブジェクトの選択を確定するためです。

ある場所をタップすると、ポインターが置かれていると認識されたオブジェクトが赤で表示され、画面の右下隅にあるオブジェクト リストに追加されます。 Enter ≈ を押すことで、これらのオブジェクトのいずれかまたはすべてを選択できます。画面をタップしてからカーソル キーを使用することで、 Enter ≈ を押す前にポインターを正確な位置に置くことができます。

ポインターが置かれていると認識されたオブジェクトが複数あるときは、ほとんどの場合、 Enter ≈ を押すと、その位置にある点が優先的に選択されます。ほかに、ポップアップ ボックスが表示されて、目的のオブジェクトを選択できる場合もあります。

選択ボックスを使用して複数のオブジェクトを選択することもできます。画面上で、選択ボックスのいずれかの隅の位置を指でタップして保持します。そのまま対角の隅の位置までドラッグします。ドラッグ中に青い選択ボックスが描画されます。このボックスに触れるオブジェクトが選択されます。

名前の非表示

[プロット表示]ではオブジェクト名を非表示にできます。

1. ラベルを非表示にするオブジェクトを選択します。
2. 選択肢 をタップするか Menu Paste を押します。
3. **[ラベルの非表示]**を選択します。

非表示にした名前を再表示するときは、同じ手順で**[ラベルの表示]**を選択します。

オブジェクトの移動

オブジェクトはさまざまな方法で移動できます。1 つ目はオブジェクトを選択せずにドラッグする方法で、オブジェクトをすばやく移動できます。

2 つ目は、まずオブジェクトをタップし、 Enter ≈ を押して選択します。その後、ドラッグしてすばやく移動するか、カーソル キーを使用して 1 ピクセルずつ移動できます。2 つ目の方法では、複数のオブジェクトを選択してまとめて移動できます。オブジェクトの移動が完了したら、オブジェクトがない場所をタップし、 Enter ≈ を押してすべての選択を解除します。オブジェクトを 1 つだけ選択した場合は、そのオブジェクトをタップし、 Enter ≈ を押して選択を解除できます。

3 つ目の方法は、オブジェクト上の点を移動することです。[プロット表示]では、オブジェクト上の点ごとに、その点の名前が付いた計算が表示されます。これをタップしたまま保持すると、スライダー バーが表示されます。スライダーをドラッグするか、カーソル キーを使用してスライダーを動かします。新しいメニュー キーとして 編集 が表示されます。このキーをタップするとダイアログ ボックスが表示され、そこでスライダーの開始値、ステップ値、および停止値を指定できます。また、スライダーを使用して、この点を基にしたアニメーションを作成できます。アニメーションの速度、一時停止、およびタイプを設定できます。アニメーションを開始または停止するには、選択し、選択肢 をタップしてから、**[アニメーション]**オプションを選択または選択解除します。

オブジェクトの色の設定

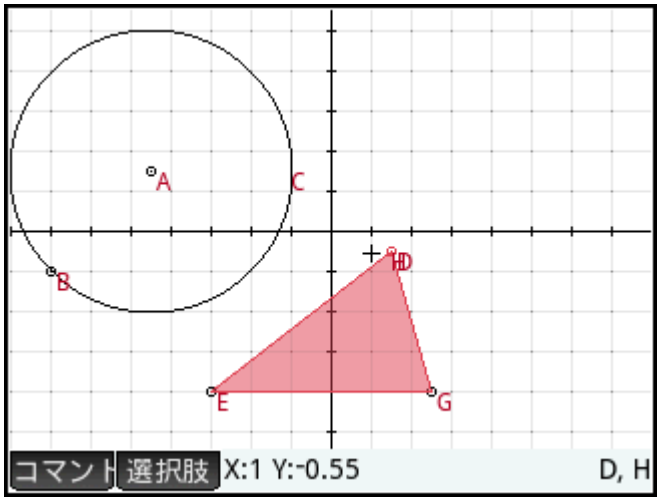
オブジェクトの色は、初期設定では黒になります。オブジェクトの色の変更方法はアプリケーション表示によって異なります。[記号表示]および[数値表示]では、項目ごとにカラー ボックスが用意されています。これらのボックスをタップして、色を選択します。[プロット表示]では、オブジェクトを選択し、 選択肢 をタップするか Menu Paste を押して、**[色の選択]**を選択してから、色を選択します。

オブジェクトの塗りつぶし

円や多角形のように閉じた輪郭を持つオブジェクトは色で塗りつぶすことができます。

1. オブジェクトを選択します。
2. 選択肢 をタップするか Menu Paste を押します。

3. **[塗りつぶし]**を選択します。

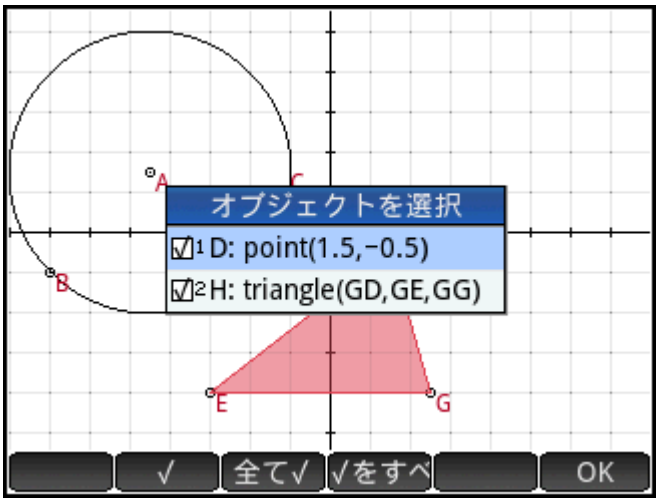


[塗りつぶし]は切り替え式のオプションです。塗りつぶしを解除するときは、同じ操作を行います。

特定のオブジェクトの消去

特定のオブジェクトを消去するには、そのオブジェクトを選択して Del をタップします。オブジェクトは、それを作成するために入力した点とは独立しています。そのため、オブジェクトを削除しても、そのオブジェクトを定義する点は削除されません。これらの点はアプリケーション内に残ります。たとえば、円を選択して Del を押すと、円は削除されますが、中心点および半径を示す点は残ります。

削除するオブジェクトに従属するオブジェクトがある場合は、選択したオブジェクトとそのすべての従属オブジェクトを削除するかどうかを確認するメッセージが表示されます。操作を確定するには OK をタップします。



複数のオブジェクトを選択して削除できます。一度に 1 つずつ選択するか選択ボックスを使用して複数のオブジェクトを選択し、 Del を押します。

オブジェクトの定義後にオブジェクトに追加した点は、オブジェクトと一緒に消去されます。たとえば、円上に点**[D]**を追加した場合、その円を削除すると、円と点**[D]**の両方が削除されます。ただし、円を定義するために入力した中心点と半径の点は残ります。

すべてのオブジェクトの消去

すべての幾何オブジェクトを消去するには、Shift Esc Clear を押します。操作の確認を求められます。[記号表示]で定義されているすべてのオブジェクトを消去する場合は OK をタップします。そのまま残す場合は キャンセ をタップします。同じ方法で、[数値表示]内のすべての測定値および計算も消去できます。

[プロット表示]のジェスチャ

画面を指でドラッグしてスクロールできます：上、下、左、または右にスクロールできます。カーソルがディスプレイの端にある場合は、カーソル キーを使用してスクロールすることもできます。ピンチ ジェスチャで画面を拡大/縮小することもできます。画面に 2 本の指を置きます。指を離すように動かせば拡大、近付けるように動かせば縮小されます。 + Ans ; を押してポインターを中心に拡大、 − Base ; を押してポインターを中心に縮小することもできます。

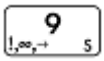
ズーム

ズームは以下のいずれの方法でも実行できます。

- 2 本指のピンチ ズームを使用します。
- + Ans ; または − Base ; を押して、それぞれ拡大または縮小します。
- ズーム をタップして、ズーム オプションを選択します。ズーム オプションは、電卓に組み込まれている多くのアプリケーションの[プロット表示]のオプションと同じです。

[プロット表示]：ボタンおよびキー

ボタンまたはキー	用途
コマンド	コマンド メニューを開きます。163 ページの[プロット表示]：[コマンド]メニューを参照してください
選択肢	選択したオブジェクトのオプション メニューを開きます
Vars Chars A	軸の表示/非表示を切り替えます
Units C	円描画ツールを選択します。画面の説明に沿って操作します（168 ページの[円]を参照）
a b/c E	すべてのトレース線を消去します
TAN ATAN I	交点描画ツールを選択します。画面の説明に沿って操作します（164 ページの[交点]を参照）

ボタンまたはキー	用途
	直線描画ツールを選択します。画面の説明に沿って操作します（165 ページの[直線]を参照）
	点描画ツールを選択します。画面の説明に沿って操作します（163 ページの[点]を参照）
	線分描画ツールを選択します。画面の説明に沿って操作します（165 ページの[線分]を参照）
	三角形描画ツールを選択します。画面の説明に沿って操作します（166 ページの[三角形]を参照）
	選択したオブジェクトを削除します(入力行がアクティブな場合はカーソルの左側の文字を削除します)
	現在の描画ツールの選択を解除します
	[プロット表示]では、すべての幾何オブジェクトを消去します。[数値表示]では、すべての測定値および計算を消去します

オプション メニュー

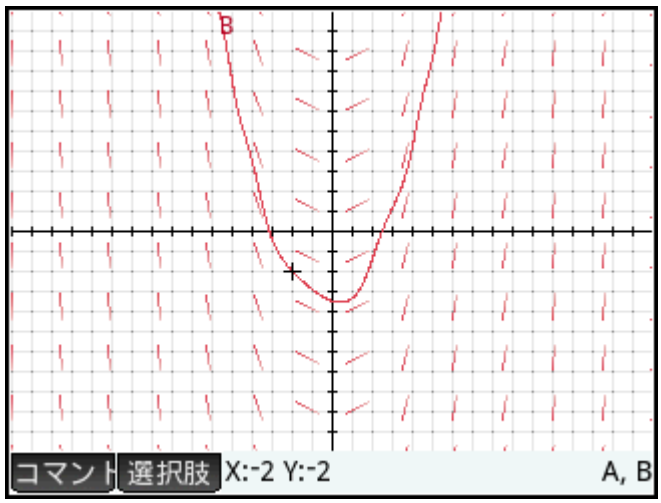
オブジェクトを選択すると、次のメニュー キーが追加されます： 選択肢 。このキーをタップすると、選択したオブジェクトのオプション（色など）を表示および選択できます。オプション メニューの内容は、選択したオブジェクトのタイプによって異なります。幾何オブジェクトで表示されるすべてのオプションを次の表に示します。これらは を押したときにも表示されます。

オプション	用途
[色の選択]	一連のカラー アイコンが表示され、選択したオブジェクトに適用する色を選択できます
[非表示]	選択したオブジェクトを非表示にします。これは、[記号表示]でオブジェクトの選択を解除する操作のショートカットです。非表示にしたオブジェクトを再表示するには、[記号表示]または[数値表示]でオブジェクトを選択します
[ラベルの非表示]	選択したオブジェクトのラベルを非表示にします。選択したオブジェクトのラベルが非表示になっている場合は、このオプションは**[ラベルの表示]**になります
[塗りつぶし]	選択したオブジェクトを色で塗りつぶします。塗りつぶしを解除するには、このオプションの選択を解除します
[トレース]	選択した点のトレースを開始します。もう一度選択すると、選択した点のトレースが停止されます
[トレースのクリア]	選択した点の現在のトレースを消去します。トレースは停止されません
[アニメーション]	オブジェクト上で選択した点の現在のアニメーションを開始します。選択した点のアニメーションがすでに開始している場合は、アニメーションが停止します

勾配場コマンドの使用

[記号表示]で勾配場コマンドを選択すると、コマンド ラインに plotfield()と入力されます。コマンドを完了するには、y’ の式を入力し、必要に応じてその他のパラメーターの値も入力します。

[プロット表示]で勾配場コマンドを選択すると、勾配場ウィザードが開きます。ウィザードでは y’ の式を入力できます。必要に応じて、その他のパラメーターの値も入力します。



たとえば、式に y’ = x - sin(y)、[ステップ]パラメーターの値に 2 を入力できます。

注記：変数は小文字で入力します。

Plot Setup を押すと、初期設定の[プロット表示]ウィンドウに勾配場が表示されます。

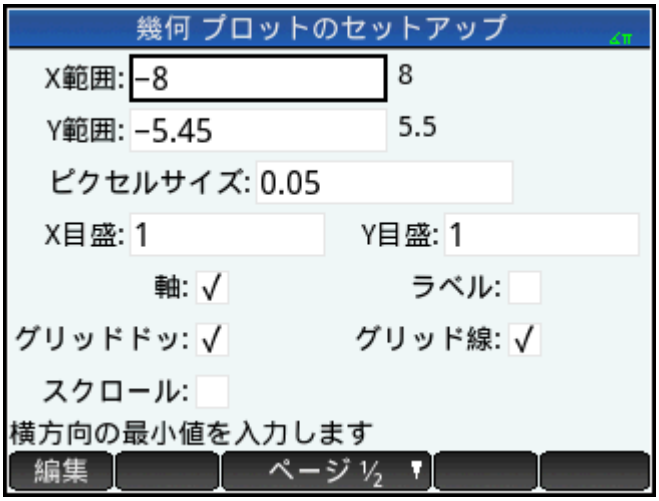
式の解のプロットを求めるには、カーソルを点に移動して Enter を押します。

たとえば、カーソルを点(-2, -2)に移動して Enter を押します。方程式 y’ = x - sin(y)に対する初期条件 x = -2、y = -2 での解のプロットが表示されます。

勾配場の定義(plotfield)および式の解(plotode)を表示するには、 Symb Setup キーを押します。

[プロットのセットアップ]表示

[プロットのセットアップ]表示では、[プロット表示]の外観を設定できます。



以下のフィールドおよびオプションがあります。

- **[X 範囲]**：2 つのボックスがありますが、x の最小値のみ編集可能です。x の最大値は、最小値およびピクセル サイズに基づいて自動的に計算されます。x 範囲は、[プロット表示]で画面をスクロールまたは拡大/縮小することによって変更することもできます。
- **[Y 範囲]**：2 つのボックスがありますが、y の最小値のみ編集可能です。y の最大値は、最小値およびピクセル サイズに基づいて自動的に計算されます。y 範囲は、[プロット表示]で画面をスクロールまたは拡大/縮小することによって変更することもできます。
- **[ピクセルサイズ]**：[プロット表示]のピクセルはすべて正方形です。これらのピクセルのサイズを変更できます。[プロット表示]の左下隅の表示は変わらず、右上隅の座標が自動的に再計算されます。
- **[軸]**：[プロット表示]の軸の表示/非表示を切り替えます。

  キーボード ショートカット： Vars Chars A
- **[ラベル]**：軸のラベルの表示/非表示を切り替えます。
- **[グリッド ドット]**：グリッドの点の表示/非表示を切り替えます。
- **[グリッド線]**：グリッドの線の表示/非表示を切り替えます。

[記号表示]の詳細

オブジェクト（点、線分、直線、多角形、曲線など）にはすべて名前が付けられ、その定義が[記号表示]（ Symb Setup ）に表示されます。名前は、[プロット表示]での表示名の先頭に「G」を付けたものになります。たとえば、[プロット表示]で A という名前の点は、[記号表示]では GA という名前になります。

「G」が先頭に付く名前は、数式処理システム（CAS）で読み取ることのできる変数であることを示します。そのため、これらの変数は CAS で計算に使用できます。たとえば下の図の GC は、[プロット表

示]で描画した円を表す変数の名前です。CAS での作業中にこの円の面積が必要になったときは、「area(GC)」と入力して Enter を押します。

注記：Geometry（幾何）変数は、CAS 内または幾何アプリケーションの[数値表示]内（160 ページの[数値表示]の詳細を参照）の計算で参照できます。

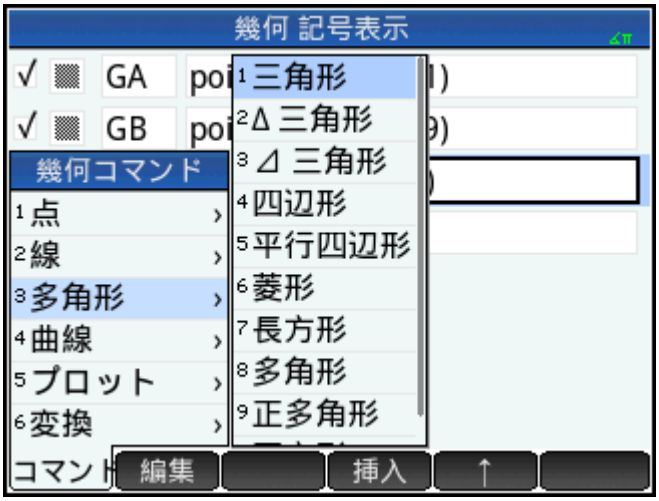
オブジェクトの定義を変更するには、目的の定義を選択し、編集をタップして、定義パラメーターを変更します。それに応じて[プロット表示]でオブジェクトが変更されます。たとえば、上の図で点**[GB]**を選択し、編集をタップして、点の座標の一方または両方を変更してから、OK をタップすると、[プロット表示]で円の大きさが変わります。

オブジェクトの作成

[記号表示]でオブジェクトを作成することもできます。新しいをタップし、オブジェクトの定義（例：point(4,6)）を入力して、Enter を押します。オブジェクトが作成され、[プロット表示]にも表示されます。

別の例：点 P と点 Q を結ぶ直線を描画するには、[記号表示]で「line(GP,GQ)」と入力して、Enter を押します。[プロット表示]に戻ると、点 P と点 Q を結ぶ直線が表示されます。

[記号表示]で使用できるオブジェクト作成コマンドは、コマンドをタップして確認できます。各コマンドの構文については、182 ページの幾何関数およびコマンドを参照してください。



エントリの並べ替え

[記号表示]のエントリを並べ替えることができます。各オブジェクトは、[記号表示]で定義されている順序で[プロット表示]に描画されます。エントリを並べ替えるには、エントリを選択し、下に移動する場合は ↓ 、上に移動する場合は ↑ をタップします。

オブジェクトの非表示

オブジェクトが[プロット表示]で表示されないようにするには、[記号表示]でその選択を解除します。

1. 非表示にするオブジェクトの定義を選択します。
2. √ をタップします。

   または

   オブジェクトのチェック ボックスを選択し、 + Ans ; を押して選択し、 − Base ; を押して選択を解除します。

オブジェクトを再表示するときは、同じ操作を行います。

オブジェクトの削除

[プロット表示]でオブジェクトを削除できるのと同様に(153 ページの特定のオブジェクトの消去を参照)、[記号表示]でもオブジェクトを削除できます。

1. 削除するオブジェクトの定義を選択します。
2. Del を押します。

すべてのオブジェクトを削除するときは、Shift Esc Clear を押します。確認メッセージが表示されたら、OK をタップして操作を確定します。

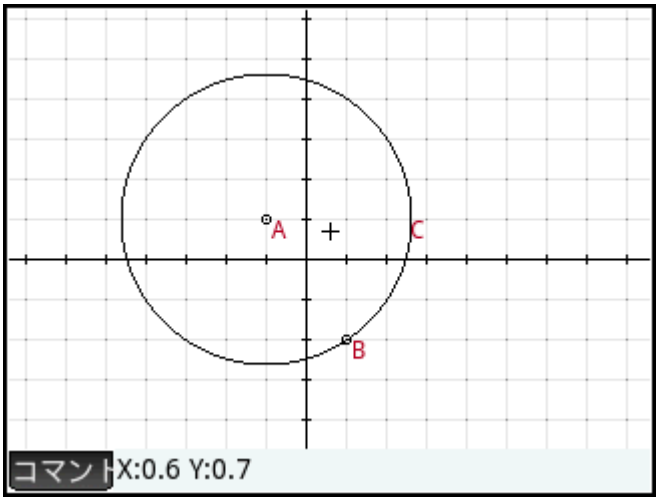
[記号のセットアップ]表示

幾何アプリケーションの[記号表示]は多くのアプリケーションと同じです。この表示で、システム全体の設定の一部を変更できます。

[数値表示]の詳細

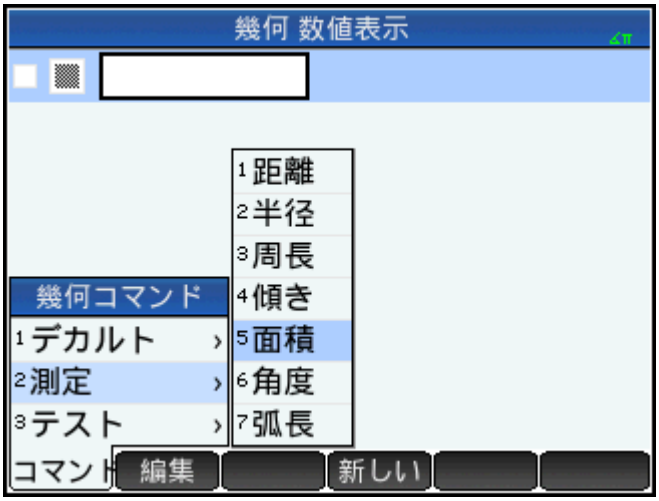
[数値表示]（ Num Setup ）では、幾何アプリケーションでの計算を実行できます。表示される結果は動的です。[プロット表示]または[記号表示]でオブジェクトを操作すると、[数値表示]でそのオブジェクトを参照する計算が、オブジェクトの新しいプロパティを反映して自動的に更新されます。

たとえば、次の図のような円**[C]**があるとします。C の面積および半径を求めるには、以下の操作を行います。



1. Num Setup を押して[数値表示]を開きます。
2. 新しい をタップします。
3. コマント をタップして、[測定]> **[面積]**の順に選択します。

   入力行に**[area()]**と入力され、面積を求めるオブジェクトを入力できる状態になります。



4. 変数 をタップし、**[曲線]**を選択して、面積を求める曲線を選択します。

   オブジェクトの名前がかっこ内に入力されます。

   コマンドおよびオブジェクト名をメニューから選択する代わりに、手動で入力することもできます。オブジェクト名を手動で入力する場合、計算で使用するときは[プロット表示]での名前の先頭に「G」を付けることを忘れないでください。[プロット表示]で**[C]**という名前の円は、[数値表示]および[記号表示]では**[GC]**になります。

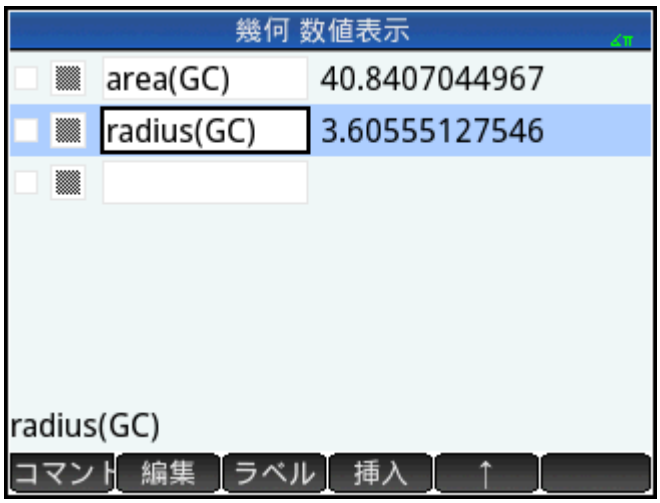
5. Enter ≈ を押すか OK をタップします。面積が表示されます。

6. 新しい をタップします。

7. 「`radius(GC)`」と入力して、 OK をタップします。半径が表示されます。これらの測定値が[プロット表示]に表示されるように、両方の √ をタップします。

   ここで使用した構文は、CAS で幾何オブジェクトのプロパティを計算するときに使用する構文と同じです。

   幾何関数およびその構文については、182 ページの幾何関数およびコマンドを参照してください。

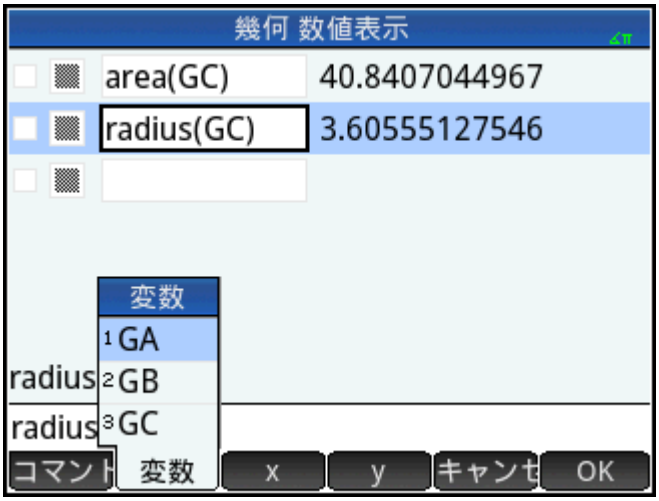


8. Plot Setup を押して[プロット表示]に戻ります。ここで、円を操作して面積および半径を変えてみます。たとえば、中心点**[A]**を選択し、カーソル キーを使用して別の位置に移動します。点を動かすと同時に、面積および半径の計算が自動的に更新されます。操作が完了したら、忘れずに Esc Clear を押してください。

注記：[数値表示]でエントリが長すぎて画面に収まらない場合は、▶ を押してスクロールすることで見えない部分を表示できます。元の状態に戻すには、◀ を押してスクロールします。

全オブジェクトのリスト

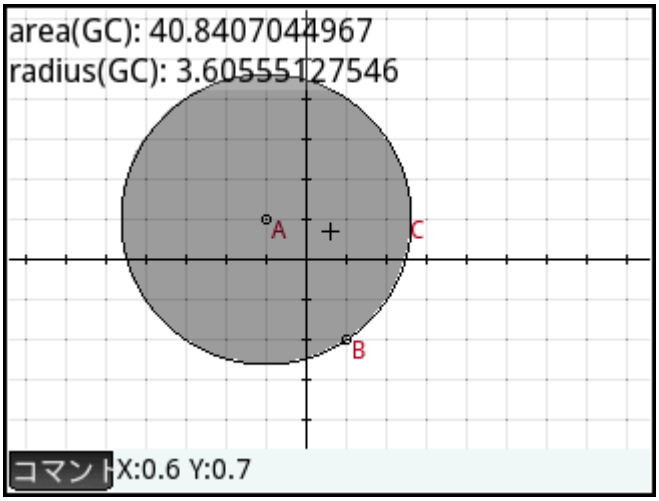
[数値表示]で新しい計算を作成するときは、変数 メニュー項目が表示されます。 変数 をタップすると、幾何アプリケーションの作業空間にあるすべてのオブジェクトのリストが表示されます。



計算を入力するときは、このメニューからオブジェクトの変数名を選択できます。選択したオブジェクトの名前が入力行の挿入ポイントに入力されます。

[プロット表示]での計算の表示

[数値表示]で作成した計算を[プロット表示]で表示するには、[数値表示]でその計算を選択して √ をタップします。計算の横にチェックマークが表示されます。



[プロット表示]で計算を非表示にするときは、同じ操作を行います。チェックマークが消えます。

計算の編集

1. 編集する計算を選択します。
2. 計算内容を変更する場合は 編集 をタップします。ラベルを変更する場合は ラベル をタップします。
3. 変更が完了したら、 OK をタップします。

計算の削除

1. 削除する計算を選択します。
2. Del を押します。

すべての計算を削除するときは、Shift Esc Clear を押します。計算を削除しても、[プロット表示]および[記号表示]から幾何オブジェクトが削除されることはありません。

[プロット表示] : [コマンド]メニュー

このセクションで説明する幾何オブジェクトは、[プロット表示]または[記号表示]のどちらのコマンドメニュー（コマンド）でも作成できます。このセクションでは、[プロット表示]でのコマンドの使用方法について説明します。[記号表示]では、[プロット表示]よりも多くのタイプのオブジェクトを作成できます。これらについては182 ページの幾何関数およびコマンドを参照してください。[プロット表示]では、測定およびその他の計算を実行することもできます。

[プロット表示]では、オブジェクトを描画するための描画ツールを選択します。このセクションではこれらの各ツールについて説明します。描画ツールを選択すると、選択を解除するまで選択されたままになります。そのため、同じタイプのオブジェクトを複数描画するとき（円をいくつか描画するときなど）にすばやく作業できます。現在の描画ツールの選択を解除するときは、Esc Clear を押します。描画ツールがまだ選択されているかどうかは、画面の左下にヘルプ、その右に現在のコマンド名が表示されているかどうかで判断できます。

このセクションで説明する手順はタッチ入力を基本にしています。たとえば、点を追加するときは、点を追加する位置を「タップ」して Enter ≈ を押すという操作を示します。実際には、カーソルキーを使用して、点を追加する位置までカーソルを移動し、Enter ≈ を押すという操作もできます。

このセクションで説明する幾何オブジェクトの描画ツールは、画面の下部にあるコマンド メニュー（コマンド）から選択できます。一部のオブジェクトは、キーボード ショートカットを使用して入力することもできます。たとえば、÷ を押すと三角形描画ツールを選択できます。154 ページの[プロット表示] : ボタンおよびキーを参照してください。

[点]

[点]

点を追加する位置をタップして、Enter ≈ を押します。

キーボード ショートカット： EEX Sto▸ P

[点オン]

新しい点を追加するオブジェクトをタップして、 Enter を押します。オブジェクト上に追加した点は、選択して動かすと、動きがそのオブジェクト内に制限されます。たとえば、円に追加した点は、どのように移動しても、その円上から離れることはありません。

[中点]

一方の端点になる位置をタップして、 Enter を押します。もう一方の端点になる位置をタップして、 Enter を押します。これら 2 つの端点間の中心に点が自動的に作成されます。

最初にオブジェクト（線分など）を選択した場合は、中点ツールを選択して Enter を押すと、そのオブジェクトの両端間の中心に点が追加されます。円の場合は、円の中心に中点が作成されます。

[中央]

円をタップして、 Enter を押します。円の中心に点が作成されます。

[交点]

交差部分をタップして、 Enter を押します。交差している位置の 1 つに点が作成されます。

キーボード ショートカット： TAN ATAN I

[交点]

点以外のオブジェクトをタップして、 Enter を押します。別のオブジェクトをタップして、 Enter を押します。2 つのオブジェクトが交差する位置に点が作成され、名前が付けられます。選択した 2 つのオブジェクトが交差していない場合でも、[記号表示]には交点オブジェクトが作成されます。

注記：このコマンドでは点が作成されます。この点の位置に基づいて交点が判断されます。点を移動して、移動先近くにある別の交点を選択できます。

[ランダム点]

Enter を押して、[プロット表示]内の好きな位置に点を作成します。続けて、 Enter を押して点を追加していきます。完了したら Esc Clear を押します。

[線]

[線分]

一方の端点になる位置をタップして、 Enter を押します。もう一方の端点になる位置をタップして、 Enter を押します。2 つの端点の間に線分が描画されます。

キーボード ショートカット： 9

[半直線]

起点になる位置をタップして、 Enter を押します。半直線の通過点になる位置をタップして、 Enter を押します。起点から始まって通過点を通る半直線が描画されます。

[直線]

直線の通過点になる位置をタップして、 Enter を押します。直線の通過点になる別の位置をタップして、 Enter を押します。2 つの通過点を通る直線が描画されます。

キーボード ショートカット： x^2

さらに別の位置（点**[C]**）をタップして Enter を押すと、点**[A]**を通り、線**[AB]**と線**[AC]**間の角度を 2 等分する直線が描画されます。

[平行]

通過点になる位置（点**[P]**）をタップして、 Enter を押します。線（線**[L]**）をタップして、 Enter を押します。点**[P]**を通り、線**[L]**に平行な新しい直線が描画されます。

[垂直]

通過点になる位置（点**[P]**）をタップして、 Enter を押します。線（線**[L]**）をタップして、 Enter を押します。点**[P]**を通り、線**[L]**に垂直な新しい直線が描画されます。

[正接]

曲線（曲線**[C]**）をタップして、 Enter を押します。通過点になる位置（点**[P]**）をタップして、 Enter を押します。点 P**[]**が曲線**[C]**上にある場合は、1 本の接線が描画されます。点**[P]**が曲線**[C]**上にない場合は、0 本または複数本の接線が描画されます。

[中線]

頂点（点**[A]**）をタップして、 Enter を押します。線分をタップして、 Enter を押します。点**[A]**と線分の中点を結ぶ直線が描画されます。

[垂線]

頂点（点**[A]**）をタップして、 Enter を押します。線分をタップして、 Enter を押します。点**[A]**を通り、線分（または線分の延長線）に垂直な直線が描画されます。

[∡二等分線]

角度を 2 等分する頂点（点**[A]**）をタップして、 Enter を押します。別の点（点**[B]**）をタップして、 Enter を押します。

[多角形]

[多角形]メニューには、さまざまな多角形を描画するためのツールがあります。

[三角形]

各頂点をタップして、タップするごとに Enter を押します。

キーボード ショートカット： $\div$

[Δ三角形]

2 つの頂点および 1 つの角度を指定して二等辺三角形を描画します。2 つの頂点によって 2 等辺の一方が決まり、角度によって 2 等辺間の角度が決まります。`equilateral_triangle` と同様に、3 つ目の点の座標を CAS 変数に保存するオプションがあります。

`isosceles_triangle(point1, point2, angle)`

例：

`isosceles_triangle(GA, GB, angle(GC, GA, GB))` は、2 等辺の一方が AB で、2 等辺間の角度が∡ACB である二等辺三角形を描画します。

[◿三角形]

2 つの点およびスケール係数を指定して直角三角形を描画します。2 つの点によって直角三角形の隣辺の一方が決まり、1 つ目の点が直角の頂点になり、スケール係数と 1 つ目の隣辺の長さを掛けることによってもう一方の隣辺の長さが決まります。

`right_triangle(point1, point2, realk)`

例：

`right_triangle(GA, GB, 1)` は、点 A が直角になり、直角の隣辺がどちらも線分 AB の長さと等しい直角二等辺三角形を描画します。

[四辺形]

各頂点をタップして、タップするごとに Enter ≈ を押します。

[平行四辺形]

1 つ目の頂点をタップして、Enter ≈ を押します。2 つ目の頂点をタップして、Enter ≈ を押します。2 つ目の頂点をタップして、Enter ≈ を押します。4 つ目の頂点の位置が自動的に計算されて、平行四辺形が描画されます。

[菱形]

2 つの点および 1 つの角度を指定して菱形を描画します。他の多くの多角形コマンドと同様に、オプションで、指定した以外の 2 つの頂点の座標を点として保存するための CAS 変数の名前を指定できます。

`rhombus(point1, point2, angle)`

例：

`rhombus(GA, GB, angle(GC, GD, GE))` は、線分 AB を辺とし、頂点 A の角度が∡DCE に等しい菱形を描画します。

[長方形]

同一辺上の 2 つの頂点、および 2 つの頂点によって定義される辺の対辺上の 1 点、または 2 つの頂点によって定義される辺に垂直な辺のスケール係数を指定して、長方形を描画します。他の多くの多角形コマンドと同様に、オプションで、指定した以外の 2 つの頂点の座標を点として保存するための CAS 変数の名前を指定できます。

`rectangle(point1, point2, point3)` または `rectangle(point1, point2, realk)`

例：

`rectangle(GA, GB, GE)` は、最初の 2 つの頂点を点 A および点 B、つまり 1 辺を線分 AB とする長方形を描画します。線分 AB の対辺は点 E を通る直線上に作成されます。

`rectangle(GA, GB, 3, p, q)` は、最初の 2 つの頂点を点 A および点 B、つまり 1 辺を線分 AB とする長方形を描画します。線分 AB に垂直な辺の長さは 3*AB になります。3 つ目の点および 4 つ目の点は、それぞれ p、q という名前の CAS 変数に保存されます。

[多角形]

すべての頂点を指定して多角形を描画します。

`polygon(point1, point2, …, pointn)`

例：

`polygon(GA, GB, GD)`は、ΔABD を描画します。

[正多角形]

最初の 2 つの頂点および辺の数を指定して正多角形を描画します。辺の数は 1 より大きくなります。辺の数が 2 の場合は、線分が描画されます。計算された点の座標を、点が作成された順に保存するための CAS 変数名を指定できます。多角形の向きは反時計回りになります。

`isopolygon(point1, point2, realn)`。ここで、realn は 1 より大きい整数です。

例：

`isopolygon(GA, GB, 6)`は、最初の 2 つの頂点が点 A および点 B である正六角形を描画します。

[正方形]

1 つ目の頂点をタップして、 Enter ≈ を押します。2 つ目の頂点をタップして、 Enter ≈ を押します。3 つ目および 4 つ目の頂点の位置が自動的に計算されて、正方形が描画されます。

[曲線]

[円]

円の中心をタップして、 Enter ≈ を押します。円周上の 1 点をタップして、 Enter ≈ を押します。最初にタップした点を中心点とし、タップした 2 つの点の距離を半径とする円が描画されます。

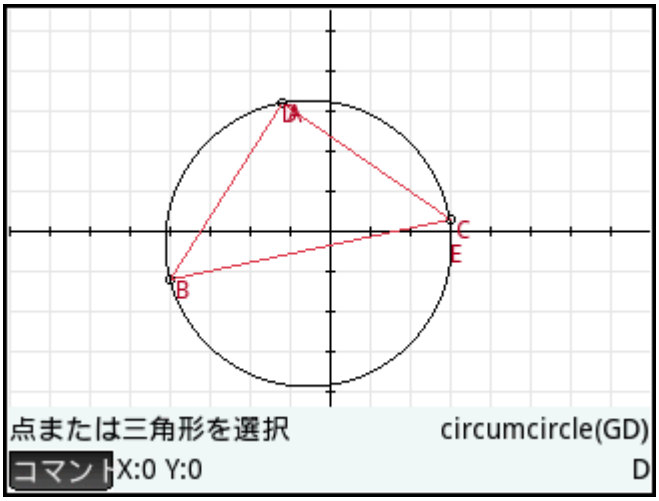
キーボード ショートカット： Units C

[記号表示]で円を定義して作成することもできます。構文は`circle(GA,GB)`で、**[A]**および**[B]**に 2 つの点を指定します。指定した**[A]**と**[B]**の距離を直径とする円が[プロット表示]に描画されます。

[外接円]

外接円は、三角形の 3 つの頂点を通って三角形を囲む円です。

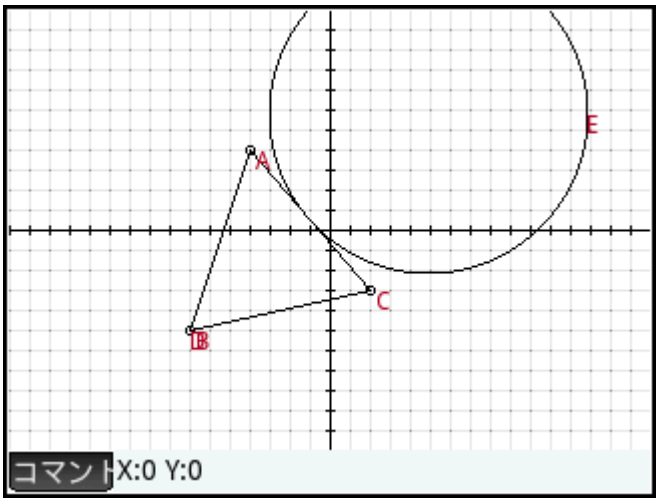
三角形の各頂点をタップして、タップするごとに Enter ≈ を押します。



[傍接円]

傍接円は、三角形の 1 辺と他の 2 辺の延長線に接する円です。三角形の各頂点をタップして、タップするごとに Enter ≈ を押します。

最後にタップした 2 つの頂点間の辺に接するように傍接円が描画されます。次の図では、A および C（または C および A）を最後にタップしています。このため、辺 AC に接する傍接円が描画されています。



[内接円]

内接円は、三角形の 3 つすべての辺に接する円です。三角形の各頂点をタップして、タップするごとに Enter ≈ を押します。

[楕円]

1 つ目の焦点をタップして、 Enter を押します。2 つ目の焦点をタップして、 Enter を押します。円周上の 1 点をタップして、 Enter を押します。

[双曲線]

1 つ目の焦点をタップして、 Enter を押します。2 つ目の焦点をタップして、 Enter を押します。双曲線の一方の枝上の 1 点をタップして、 Enter を押します。

[放物線]

焦点をタップして、 Enter を押します。直線（準線）または半直線か線分をタップして、 Enter を押します。

[円錐]

x と y の式で定義される円錐断面のグラフをプロットします。

`conic(expr)`

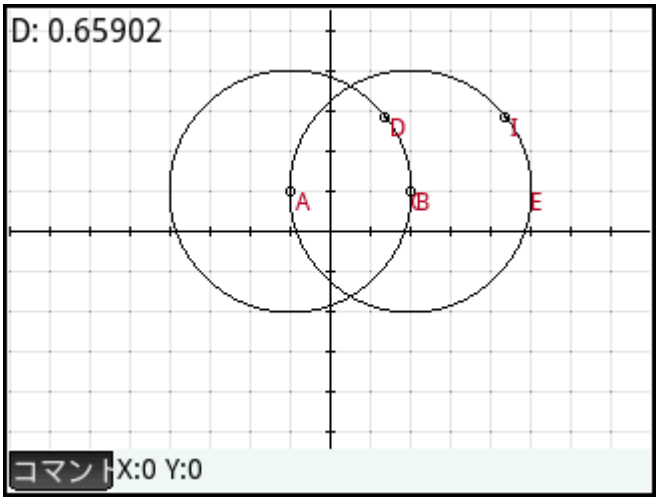
例：

`conic(x^2+y^2-81)` は、中心(0,0)、半径 9 の円を描画します。

[軌跡]

引数として 2 つの点を指定します。1 つ目は軌跡を描くための点、2 つ目はオブジェクト上の点です。2 つ目の点がオブジェクト上を動くと、その動きに従って 1 つ目の点が軌跡を描きます。

次の図では、円 C が描画され、点 D が前述の**[点オン]**で円 C 上に描画されています。点 I は点 D を平行移動した点です。[曲線]> [特殊]> **[軌跡]**の順に選択すると、入力行に**[locus(]**と入力されます。引数を指定して「`locus(GI,GD)`」とし、点 D を円上で動かすと（点 D の動きは円上に制限されています）、点 I が点 D と平行する軌道をたどり、軌跡が描画されます。

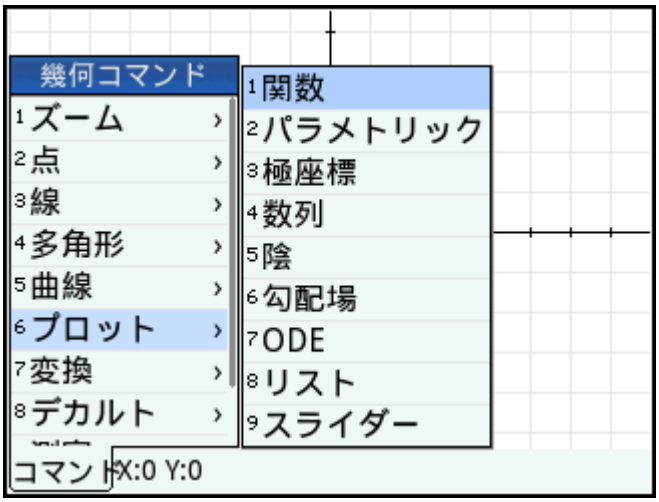


[プロット]

[プロット表示]では、以下のタイプの式をプロットできます。

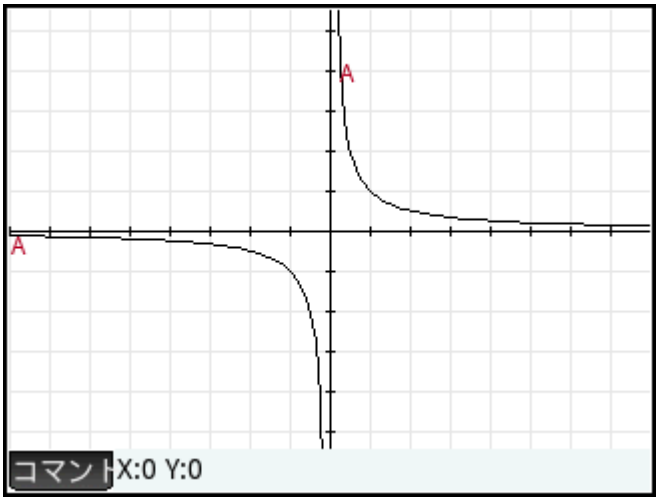
- 関数
- パラメトリック
- 極座標
- 数列

タップし、**[プロット]**を選択して、プロットする式のタイプを選択します。入力行がアクティブになり、式の定義を入力できます。



式に含める変数は小文字で入力する必要があります。

この例では、プロットのタイプとして**[関数]**を選択し、y = 1/x のグラフがプロットされています。



[関数]

構文：`plotfunc(Expr)`

独立変数 x の式を与えると、関数のプロットを描画します。編集行が表示されるので、式を入力して Enter ≈ を押します。x は小文字で入力してください。

変数を宣言するのであれば、異なる変数で式を入力することもできます。これを行う場合、構文は `plotfunc(expr(var, var)` となります。

例：

`plotfunc(3*sin(x))` は y=3*sin(x)のグラフを描画します

`plotfunc(a^2, a)` は、放物線のグラフをプロットします

[パラメトリック]

構文：`plotparam(f(Var)+i*g(Var), Var= Start..Stop, [tstep=Value])`

1 変数の複素数の式と、その変数の区間を引数として指定します。複素数の式 `f(t) + i*g(t)` を `x = f(t)` および `y = g(t)` として解釈し、2 番目の引数で指定した区間にわたってパラメトリック方程式をプロットします。編集行が表示されるので、複素数の式および区間を入力します。

例：

`plotparam(cos(t)+ i*sin(t), t=0..2*π)` は単位円をプロットします

`plotparam(cos(t)+ i*sin(t), t=0..2*π, tstep=π/3)` は、単位円に内接する正六角形をプロットします（tstep 値に注意してください）

[極座標]

構文：`plotpolar(Expr,Var=Interval, [Step])` または `plotpolar(Expr, Var, Min, Max, [Step])`

[プロット表示]に極座標グラフを描画します。編集行が表示されるので、x の式および区間（さらにオプションでステップ）を入力します。

`plotpolar(f(x),x,a,b)` は、x が `[a,b]` のときの極座標曲線 `r=f(x)` を描画します

[数列]

構文：`plotseq(f(Var), Var={Start, Xmin, Xmax}, Integer n)`

x の式と、3 つの値を含むリストを与え、直線 y=x と、最後の 2 つの値の間の区間として定義される定義域にわたって式によって定義される関数のプロットを描画し、式によって再帰的に定義される（最初の値で開始される）数列の最初の n 項についてのクモの巣状のプロットを描画します。

例：

`plotseq(1-x/2, x={3 -1 6}, 5)` は、`y=x` および `y=1-x/2`（`x=-1` から `x=6`）をプロットし、`u(0)=3` で始まる `u(n)=1-(u(n-1)/2)` のクモの巣状のプロットの最初の 5 項を描画します

[陰]

構文：`plotimplicit(Expr, [XIntrvl, YIntrvl])`

式（x および y）によって陰関数で定義された曲線をプロットします。特に、式=0 についてプロットします。小文字の x および y を使用することに注意してください。オプションの x 区間および y 区間を指定すると、その区間内でのみプロットします。

例：

`plotimplicit((x+5)^2+(y+4)^2-1)` は、点(-5, -4)を中心とする半径 `1` の円をプロットします

[勾配場]

構文：`plotfield(Expr, [x=X1..X2 y=Y1..Y2], [Xstep, Ystep], [Option])`

指定された x 範囲および y 範囲で微分方程式 y’=f(x,y)の勾配場のグラフをプロットします。オプションが `normalize` の場合、描画される勾配場の線分の長さが同じになります。

例：

`plotfield(x*sin(y), [x=-6..6, y=-6..6],normalize)` は、`y'=x*sin(y)` の勾配場を`-6` から `6` まで両方向に描画します。線分はすべて同じ長さになります

[ODE]

構文：`plotode(Expr, [Var1, Var2, ...], [Val1, Val2. ...])`

変数 Val1、Val2,...の初期条件を含む微分方程式 y’=f(r1, Var2, ...)の解を描画します。最初の引数は式 f(Var1, Var2,...)、2 番目の引数は変数のベクトル、3 番目の引数は初期条件のベクトルです。

例：

`plotode(x*sin(y), [x,y], [-2, 2])` は、点`(-2, 2)` を初期条件として渡す `y’=x*sin(y)` の解のグラフを描画します

[リスト]

構文：`plotlist(Matrix 2xn)`

n 個の点のセットをプロットし、これらを線分で結びます。点は 2xn 行列で定義され、最初の行が横座標、2 番目の行が縦座標となります。

例：

`plotlist([[0,3],[2,1],[4,4],[0,3]])` は三角形を描画します

[スライダー]

パラメーターの値の制御に使用できるスライダー バーを作成します。ダイアログ ボックスに、スライダー バーの定義およびスライダーのアニメーションが表示されます。

[変換]

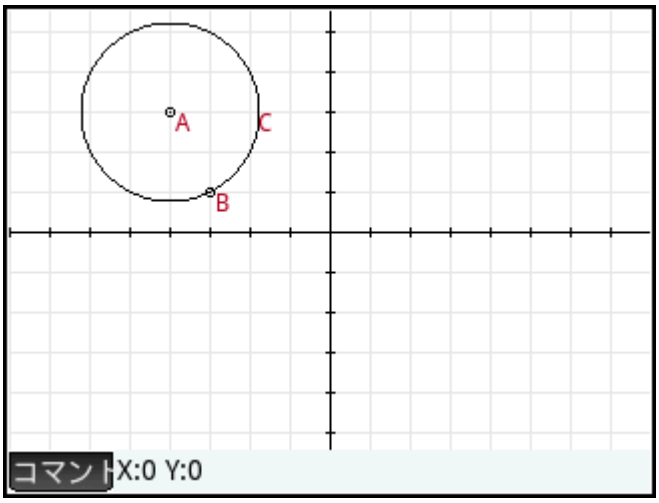
[変換]メニューには、[プロット表示]の幾何オブジェクトに対して変換を実行するためのさまざまなツールがあります。[記号表示]で変換を定義することもできます。

[平行移動]

平行移動は、一連の点を同じ方向に同じ距離だけ移動させる変換方法です。T : (x,y)→(x+a, y+b)

たとえば、次の図の円 B を右下に移動するには、以下の操作を行います。

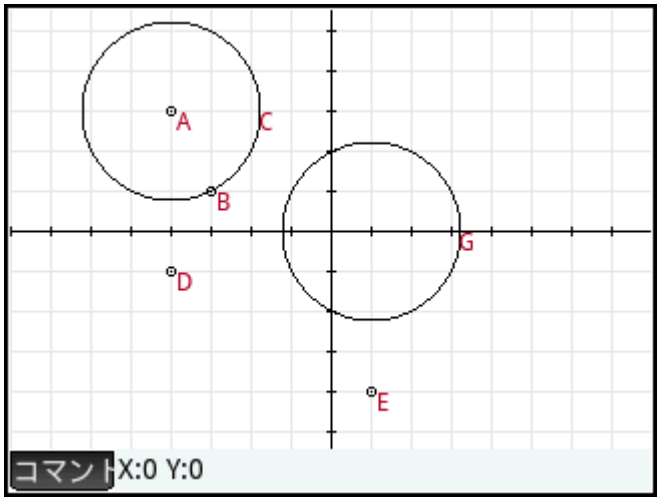
1. コマンドをタップし、**[変換]**をタップして、**[平行移動]**を選択します。
2. 移動させるオブジェクトをタップして、 Enter を押します。



3. 初期位置をタップして、 Enter を押します。

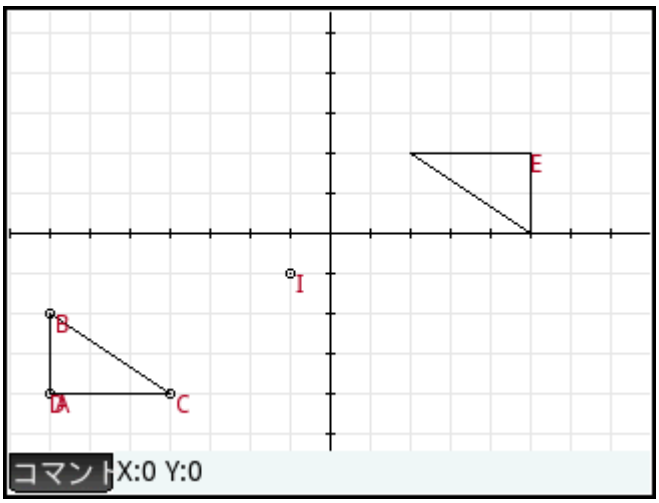
4. 最終位置をタップして、 Enter を押します。

初期位置から最終位置までの距離および方向と同じ距離および方向にオブジェクトが移動します。元のオブジェクトは元の位置に残ります。



[鏡映]

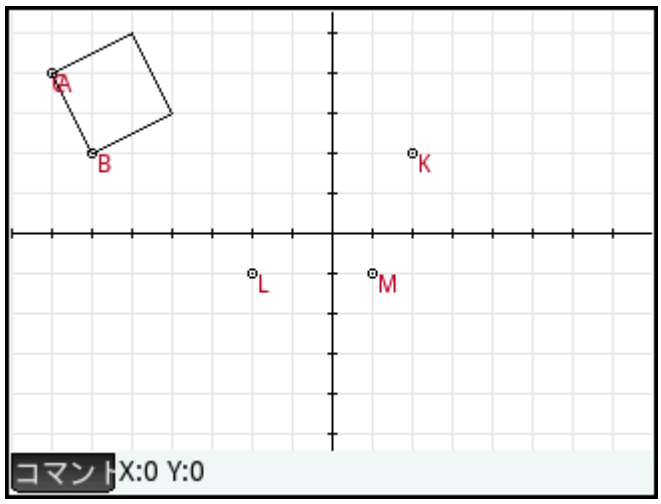
鏡映は、オブジェクトまたは一連の点を鏡映しの写像にする変換方法です。鏡には点または線を指定できます。点を鏡とする鏡映は半回転とも呼ばれます。いずれの場合でも、鏡像の各点と鏡との距離は、それぞれの元の点と鏡との距離に等しくなります。次の図では、元の三角形**[D]**が点**[I]**を鏡として鏡映変換されています。



1. コマント をタップし、**[変換]**をタップして、**[鏡映]**を選択します。
2. 対称軸（鏡）になる点または線オブジェクト（線分、半直線、または直線）をタップして、 Enter を押します。
3. 対称軸を中心に鏡映させるオブジェクトをタップして、 Enter を押します。手順 2 で定義した対称軸を中心にオブジェクトが鏡映されます。

[回転]

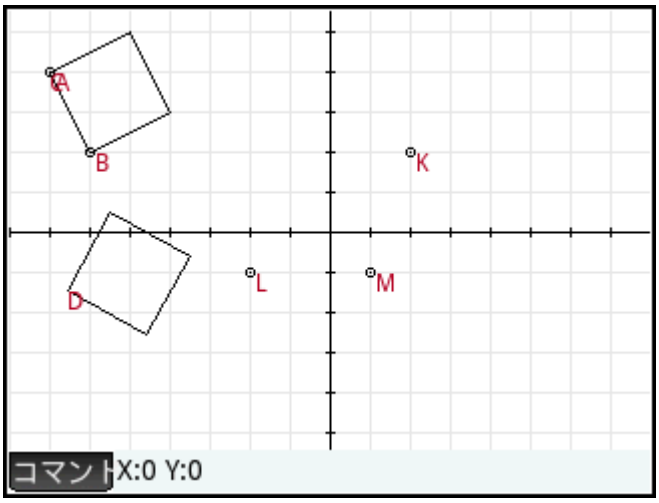
回転は、各点を、ある点を中心に特定の角度だけ回転させる変換方法です。角度は `angle()` コマンドで定義し、1 番目の引数として角度の頂点を指定します。たとえば、下の図の正方形（**[GC]**）を、点 K（**[GK]**）を中心に**[∠LKM]**の角度だけ回転させるには、以下の操作を行います。



1. コマンド をタップし、**[変換]**をタップして、**[回転]**を選択します。入力行に `rotation()` と入力されます。
2. かっこ内に次のように入力します。

   `GK,angle(GK,GL,GM),GC`

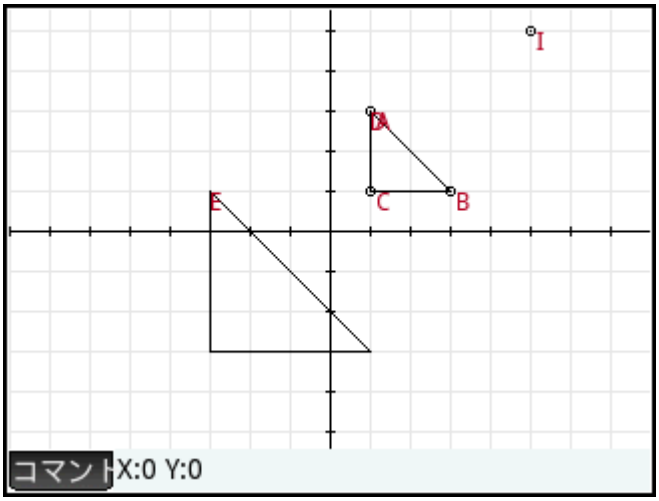
3. Enter ≈ を押すか OK をタップします。
4. Plot ↵Setup を押して[プロット表示]に戻ると、回転した正方形が表示されます。



[拡大縮小]

拡大縮小（相似または均等スケーリングとも呼ばれます）は、特定の点を中心として特定のスケール係数だけオブジェクトを拡大または縮小する変換方法です。

次の図では、スケール係数が 2 で、画面右上にある点**[I]**を中心点としています。新しい三角形の各点は、元の三角形の各点と点 I を結ぶ線の延長線上にあります。また、点 I から新しい三角形の各点までの距離は、点 I から元の三角形の各点までの距離の 2 倍になります(スケール係数が 2 であるため)。



1. コマントをタップし、**[変換]**をタップして、**[拡大縮小]**を選択します。
2. 拡大縮小の中心となる点をタップして、 Enter ≈ を押します。
3. スケール係数を入力して、 Enter ≈ を押します。
4. 拡大縮小するオブジェクトをタップして、 Enter ≈ を押します。

[相似]

幾何オブジェクトを同一の中心点について拡大および回転します。

`similarity(point, realk, angle, object)`

例：

`similarity(0, 3, angle(0,1,i),point(2,0))` は点(2,0)を相似比 3 で拡大し（点(6,0)）、結果を反時計回りに 90°回転させて点(0, 6)を作成します。

[投影]

投影は、1 つ以上の点を、各点およびその像を通る直線が像点で直角に交わるようにオブジェクトに写像する変換方法です。

1. コマントをタップし、**[変換]**をタップして、**[投影]**を選択します。
2. 点を投影するオブジェクトをタップして、 Enter ≈ を押します。
3. 投影する点をタップして、 Enter ≈ を押します。

   投影先のオブジェクトに新しい点が追加されます。

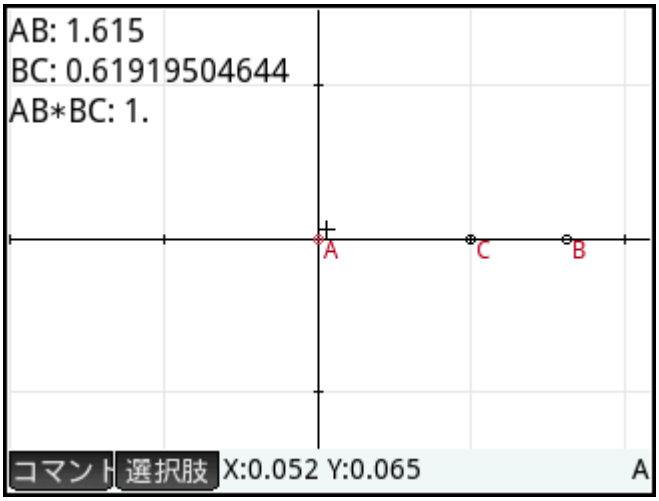
[反転]

反転は、中心点およびスケール係数に基づく変換方法です。具体的には、点 A をスケール係数 k で中心 C の向こうに反転すると、A の写像が A’ に作成されます。A’ は直線 CA 上にあり、CA*CA’ =k（CA および CA’ は対応する線分の長さを示す）を満たします。k=1 の場合、CA の長さは CA’ の長さの逆数です。

たとえば、点 A に関する点 B の反転を求めるには、以下の操作を行います。

1. コマンドをタップし、**[変換]**をタップして、**[反転]**を選択します。
2. 点**[B]**をタップして、 Enter を押します。
3. 反転比を入力して（ここでは初期値の 1 を使用します）、 Enter を押します。
4. 点**[A]**をタップして、 Enter を押します。

   図では、点**[C]**が、点**[A]**に関する点**[B]**の反転になります。



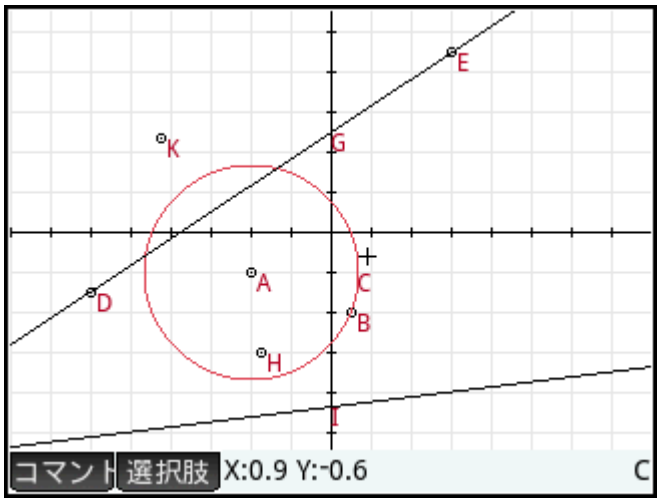
[逆数変換]

逆数変換は、円を使用した特殊な反転です。円に関する逆数変換では、平面上の各点がそれぞれの極線に変換されます。反対に、平面上の各線はそれぞれの極に写像されます。

1. コマンドをタップし、**[変換]**をタップして、**[逆数変換]**を選択します。
2. 円をタップして、 Enter を押します。
3. 点をタップして、 Enter を押します。その極線が表示されます。

4. 線をタップして、 Enter ≈ を押します。その極が表示されます。

次の図では、点**[K]**は線**[DE]**（G）の逆変換写像で、下部にある線**[I]**は点**[H]**の逆変換写像です。



[デカルト]

[横座標]

点をタップし、 Enter ≈ を押して選択します。画面の左上に、点の横座標（x 座標）が表示されます。

[縦座標]

点をタップし、 Enter ≈ を押して選択します。画面の左上に、点の縦座標（y 座標）が表示されます。

点→複素数

点またはベクトルをタップし、 Enter ≈ を押して選択します。点の座標（またはベクトルの x および y の長さ）が複素数として画面左上に表示されます。

[座標]

点をタップし、 Enter ≈ を押して選択します。画面の左上に、点の座標が表示されます。

[方程式]

点以外のオブジェクトをタップし、 Enter ≈ を押して選択します。オブジェクトの方程式（x と y、またはそのいずれか）が表示されます。

[パラメトリック]

点以外のオブジェクトをタップし、 Enter を押して選択します。オブジェクトのパラメトリック方程式（x(t)+i*y(t)）が表示されます。

[極座標]

点をタップし、 Enter を押して選択します。画面の左上に、点の極座標が表示されます。

[測定]

[距離]

点をタップし、 Enter を押して選択します。同じ操作で 2 つ目の点を選択します。2 つの点の間の距離が表示されます。

[半径]

円をタップし、 Enter を押して選択します。円の半径が表示されます。

[周長]

円をタップし、 Enter を押して選択します。円の円周が表示されます。

[傾き]

直線オブジェクト（線分、直線など）をタップし、 Enter を押して選択します。オブジェクトの傾きが表示されます。

[面積]

円または多角形をタップし、 Enter を押して選択します。オブジェクトの面積が表示されます。

[角度]

点をタップし、 Enter を押して選択します。同じ操作で 3 つの点を選択します。1 つ目の点を頂点とし、2 つ目の点から 3 つ目の点に向かう有向角の大きさが表示されます。

[弧長]

曲線をタップし、 Enter を押して選択します。次に、開始値および終了値を入力します。曲線上の 2 つの x 値間の弧の長さが表示されます。

[テスト]

[共線]

点をタップし、 Enter を押して選択します。同じ操作で 3 つの点を選択します。画面の上部にテストとその結果が表示されます。すべての点が同一直線上にある場合は、1 を返します。それ以外の場合は 0 を返します。

[円周上]

点をタップし、 Enter を押して選択します。同じ操作で 4 つの点を選択します。画面の上部にテストとその結果が表示されます。すべての点が同一円周上にある場合は、1 を返します。それ以外の場合は 0 を返します。

[オブジェクト上]

点をタップし、 Enter を押して選択します。次に、オブジェクトをタップし、 Enter を押します。画面の上部にテストとその結果が表示されます。点がオブジェクト上にある場合、テストは、点が含まれている部分を表す数を返します（側面の 1 ~ n の数）。それ以外の場合は 0 を返します。

[平行]

直線オブジェクト（線分、直線など）をタップし、 Enter を押して選択します。次に、別の直線オブジェクトをタップし、 Enter を押します。画面の上部にテストとその結果が表示されます。オブジェクトが互いに平行である場合は、1 を返します。それ以外の場合は 0 を返します。

[垂直]

直線オブジェクト（線分、直線など）をタップし、 Enter を押して選択します。次に、別の直線オブジェクトをタップし、 Enter を押します。画面の上部にテストとその結果が表示されます。オブジェクトが互いに垂直である場合は、1 を返します。それ以外の場合は 0 を返します。

[二等辺]

三角形をタップし、 Enter を押して選択します。または、3 つの点を順に選択します。三角形が二等辺三角形でない場合、または 3 つの点が二等辺三角形を形成しない場合は、0 を返します。三角形が二等辺三角形である場合、または 3 つの点が二等辺三角形を形成する場合は、長さの等しい 2 辺が共有する点の番号（1、2、または 3）を返します。三角形が正三角形である場合、または 3 つの点が正三角形を形成する場合は、4 を返します。

[等辺形]

三角形をタップし、 Enter を押して選択します。または、3 つの点を順に選択します。三角形が正三角形である場合、または 3 つの点が正三角形を形成する場合は、1 を返します。それ以外の場合は 0 を返します。

[平行四辺形]

点をタップし、 Enter を押して選択します。同じ操作で 4 つの点を選択します。画面の上部にテストとその結果が表示されます。点が平行四辺形を形成しない場合は、0 を返します。平行四辺形を形成する場合は、1 を返します。菱形を形成する場合は、2 を返します。長方形を形成する場合は、3 を返します。正方形を形成する場合は、4 を返します。

[共役]

円をタップし、 Enter を押して選択します。次に、2 つの点または 2 つの線を選択します。2 つの点または線が円の共役である場合は、1 を返します。それ以外の場合は 0 を返します。

幾何関数およびコマンド

このセクションで説明する幾何固有の関数およびコマンドは、[記号表示]および[数値表示]で コマンド をタップして使用できるか、[カタログ]メニューからのみ使用できます。

幾何アプリケーションの[数値表示]または CAS で幾何オブジェクトを参照する計算を行うときは、[記号表示]で割り当てられる、先頭に G の付いた名前を使用する必要があります。

たとえば、`altitude(GA,GB,GC)` のような形式にします。

多くの場合、以下の構文で指定するパラメーターには、点の名前（GA など）または点を表す複素数を指定できます。

たとえば、`angle(A,B,C)` は次のいずれかの形式で指定できます。

- `angle(GP, GR, GB)`
- `angle(3+2i, 1-2i, 5+i)`
- `angle(GP,1-2*i,i)`（名前による点の指定と複素数による点の指定の組み合わせ）

[記号表示] : [コマンド]メニュー

[記号表示]のコマンド メニューは、[プロット表示]のコマンド メニューとほぼ同じです。ただし、[記号表示]には[ズーム]のカテゴリはなく、[数値表示]にある[デカルト]、[測定]、および[テスト]のカテゴリもありません。[記号表示]では、コマンドは構文を使用して入力します。コマンドを強調表示して Help User を押すと、その構文を確認できます。[記号表示]で定義を入力または編集する利点は、点の正確な位置を指定できることです。点の正確な位置を入力すると、従属オブジェクト（線、円など）のプロパティが CAS で正確に報告されます。これにより、テスト コマンドを使用して幾何オブジェクトに対する推測をテストできます。以下のコマンドはすべて CAS 表示で使用でき、同じオブジェクトが返されます。

[点]

[点]

点の座標を指定して点を作成します。各座標は、値で指定するか、幾何学的作図内の他のオブジェクトの変数または測定を含む式で指定できます。

`point(real1, real2)` または `point(expr1, expr2)`

例：

`point(3,4)` は、座標が(3,4)の点を作成します。この点は後から選択して移動できます。

`point(abscissa(A), ordinate(B))` は、x 座標が点 A の x 座標と等しく、y 座標が点 B の y 座標と等しい点を作成します。点 A または点 B を移動すると、この点もそれに合わせて移動します。

[点オン]

指定された値を横座標とする点を幾何オブジェクト上に作成するか、指定された区間に実数値を作成します。

`element(object, real)` または `element(real1..real2)`

例：

`element(plotfunc(x^2),-2)` は、$y = x^2$ のグラフ上に点を作成します。この点の初期座標は(−2,4)になります。この点は移動できますが、動きはこの関数グラフ上に制限されます。

`element(0..5)` は、初期値が 2.5 のスライダー バーを作成します。この値をタップしたまま保持すると、スライダーが表示されます。 ▶ または ◀ を押すと、スライダー バーの値が増減します。スライダー バーを閉じるときは、 Esc Clear を押します。設定する値は、この後プロットする関数の係数、または一部の他のオブジェクトや計算の係数として使用できます。

[中点]

線分の中点を返します。引数は、線分の名前、または線分を定義する 2 つの点で指定できます。後者の場合、線分が実際に描画されている必要はありません。

`midpoint(segment)` または `midpoint(point1, point2)`

例：

`midpoint(0,6+6i)` は、 `point(3,3)` を返します。

[中央]

構文：`center(Circle)`

円の中心をプロットします。円は、円コマンドで定義するか、名前（**[GC]**など）で指定できます。

例：

`center(circle(x^2+y2-x-y))` は、`point(1/2,1/2)` をプロットします。

[交点]

構文：`single_inter(Curve1, Curve2, [Point])`

点 Point に最も近い位置にある、曲線 Curve1 と曲線 Curve2 の交点をプロットします。

例：

`single_inter(line(y=x), circle(x^2+y^2=1), point(1,1))`は、`point((1+i)*√2/2)`をプロットします。

[交点]

2 つの曲線の交点をベクトルとして返します。

`inter(Curve1, Curve2)`

例：

`inter(8-x^2/6, x/2-1)`は、`[[6 2],[-9 -11/2]]`を返します。

注記：このコマンドでは点が作成されます。この点の位置に基づいて交点が判断されます。点を移動して、移動先近くにある別の交点を選択できます。

[線]

[線分]

端点によって定義される線分を描画します。

`segment(point1, point2)`

例：

`segment(1+2i, 4)`は、座標が(1, 2)および(4, 0)の点によって定義される線分を描画します。

`segment(GA, GB)`は、線分 AB を描画します。

[半直線]

2 点を指定して、1 つ目の点から始まって 2 つ目の点を通る半直線を描画します。

`half_line((point1, point2)`

[直線]

直線を描画します。引数は、2 つの点、a*x+b*y+c 形式の線の式、または点と傾き（例を参照）で指定できます。

`line(point1, point2)`または`line(a*x+b*y+c)`または`line(point1, slope=realm)`

例：

`line(2+i, 3+2i)`は、点(2,1)および点(3,2)を通る直線、つまり式 y=x－1 の直線を描画します。

`line(2x-3y-8)`は、式 2x－3y=8 の直線を描画します。

`line(3-2i,slope=1/2)`は、点(3, －2)を通り、傾きが m=1/2 の直線、つまり式 x－2y=7 の直線を描画します。

[平行]

指定した点を通り、指定した線に平行な直線を描画します。

`parallel(point,line)`

例：

`parallel(A, B)`は、点 A を通り、線 B に平行な直線を描画します。

`parallel(3-2i, x+y-5)`は、点(3, − 2)を通り、式 x+y=5 の直線に平行な直線、つまり式 y= − x+1 の直線を描画します。

[垂直]

指定した点を通り、指定した線に垂直な直線を描画します。線は、名前、2 つの点、または x と y の式で指定できます。

`perpendicular(point, line)`または`perpendicular(point1, point2, point3)`

例：

`perpendicular(GA, GD)`は、点 A を通り、線 D に垂直な直線を描画します。

`perpendicular(3+2i, GB, GC)`は、座標が(3, 2)の点を通り、線 BC に垂直な直線を描画します。

`perpendicular(3+2i,line(x-y=1))`は、座標が(3, 2)の点を通り、式 x − y = 1 の直線に垂直な直線、つまり式 y = − x + 5 の直線を描画します。

[正接]

指定された点を通る、指定された曲線の接線を描画します。点は曲線上になくてもかまいません。

`tangent(curve, point)`

例：

`tangent(plotfunc(x^2), GA)`は、点 A を通る、y=x^2 のグラフの接線を描画します。

`tangent(circle(GB, GC-GB), GA)`は、中心が点 B で半径が線分 BC で定義される円に対して、点 A を通る 1 本以上の接線を描画します。

[中線]

三角形を定義する 3 つの点を指定して、1 つ目の点と、他の 2 つの点で定義される線分の中点を通る、三角形の中線を描画します。

`median_line(point1, point2, point3)`

例：

`median_line(0, 8i, 4)`は、式 y=2x の直線を描画します。この直線は、(0,0)と、端点が(0, 8)および(4, 0)である線分の中点(2,4)を通ります。

[垂線])

同一線上にない 3 つの点を指定して、3 つの点によって定義される三角形の、1 つ目の点を通る垂線を描画します。三角形を描画する必要はありません。

`altitude(point1, point2, point3)`

例：

`altitude(A, B, C)`は、点 A を通り、線 BC に垂直な直線を描画します。

[二等分線]

3 つの点を指定して、1 つ目の点を頂点とし、3 つの点によって定義される角度の二等分線を作成します。角度を[プロット表示]に描画する必要はありません。

bisector(point1, point2, point3)

例：

bisector(A,B,C)は、∡BACの二等分線を描画します。

bisector(0,-4i,4)は、y=−xによって指定される直線を描画します。

[多角形]

[三角形]

3つの頂点を指定して三角形を描画します。

triangle(point1, point2, point3)

例：

triangle(GA, GB, GC)は、ΔABCを描画します。

[Δ三角形]

2つの頂点および1つの角度を指定して二等辺三角形を描画します。2つの頂点によって2等辺の一方が決まり、角度によって2等辺間の角度が決まります。equilateral_triangleと同様に、3つ目の点の座標をCAS変数に保存するオプションがあります。

isosceles_triangle(point1, point2, angle)

例：

isosceles_triangle(GA, GB, angle(GC, GA, GB)は、2等辺の一方がABで、2等辺間の角度が∡ACBである二等辺三角形を描画します。

[◿三角形]

2つの点およびスケール係数を指定して直角三角形を描画します。2つの点によって直角三角形の隣辺の一方が決まり、1つ目の点が直角の頂点になり、スケール係数と1つ目の隣辺の長さを掛けることによってもう一方の隣辺の長さが決まります。

right_triangle(point1, point2, realk)

例：

right_triangle(GA, GB, 1)は、点Aが直角になり、直角の隣辺がどちらも線分ABの長さと等しい直角二等辺三角形を描画します。

[四辺形]

4つの点のセットから四辺形を描画します。

quadrilateral(point1, point2, point3, point4)

例：

quadrilateral(GA, GB, GC, GD)は、四辺形ABCDを描画します。

[平行四辺形]

3つの頂点を指定して平行四辺形を描画します。4つ目の点は自動的に計算されますが、記号によって定義されません。他の多くの多角形コマンドと同様に、4つ目の点の座標をCAS変数に保存できます。平行四辺形の向きは、1つ目の点を始点として反時計回りになります。

parallelogram(point1, point2, point3)

例：

parallelogram(0,6,9+5i)は、(0, 0)、(6, 0)、(9, 5)、および(3,5)を頂点とする平行四辺形を描画します。最後の点の座標は自動的に計算されます。

[菱形]

2 つの点および 1 つの角度を指定して菱形を描画します。他の多くの多角形コマンドと同様に、オプションで、指定した以外の 2 つの頂点の座標を点として保存するための CAS 変数の名前を指定できます。

rhombus(point1, point2, angle)

例：

rhombus(GA, GB, angle(GC, GD, GE))は、線分 AB を辺とし、頂点 A の角度が∡DCE に等しい菱形を描画します。

[長方形]

同一辺上の 2 つの頂点、および 2 つの頂点によって定義される辺の対辺上の 1 点、または 2 つの頂点によって定義される辺に垂直な辺のスケール係数を指定して、長方形を描画します。他の多くの多角形コマンドと同様に、オプションで、指定した以外の 2 つの頂点の座標を点として保存するための CAS 変数の名前を指定できます。

rectangle(point1, point2, point3)または rectangle(point1, point2, realk)

例：

rectangle(GA, GB, GE)は、最初の 2 つの頂点を点 A および点 B、つまり 1 辺を線分 AB とする長方形を描画します。線分 AB の対辺は点 E を通る直線上に作成されます。

rectangle(GA, GB, 3, p, q)は、最初の 2 つの頂点を点 A および点 B、つまり 1 辺を線分 AB とする長方形を描画します。線分 AB に垂直な辺の長さは 3*AB になります。3 つ目の点および 4 つ目の点は、それぞれ p、q という名前の CAS 変数に保存されます。

[多角形]

すべての頂点を指定して多角形を描画します。

polygon(point1, point2, …, pointn)

例：

polygon(GA, GB, GD)は、ΔABD を描画します。

[正多角形]

最初の 2 つの頂点および辺の数を指定して正多角形を描画します。辺の数は 1 より大きくなります。辺の数が 2 の場合は、線分が描画されます。計算された点の座標を、点が作成された順に保存するための CAS 変数名を指定できます。多角形の向きは反時計回りになります。

isopolygon(point1, point2, realn)。ここで、realn は 1 より大きい整数です。

例：

isopolygon(GA, GB, 6)は、最初の 2 つの頂点が点 A および点 B である正六角形を描画します。

[正方形]

2 つの連続する頂点を点として指定した正方形を描画します。

`square(point1, point2)`

例：

`square(0, 3+2i, p, q)` は、(0, 0)、(3, 2)、(1, 5)、および(-2, 3)を頂点とする正方形を描画します。最後の 2 つの頂点は自動的に計算されて、CAS 変数 p および変数 q に保存されます。

[曲線]

[円]

直径の両端、中心と半径、または x と y の式を与え、円を描画します。

`circle(point1, point2)` または `circle(point1, point 2-point1)` または `circle(equation)`

例：

`circle(GA, GB)` は、AB を直径とする円を描画します。

`circle(GA, GB-GA)` は、中心 A、半径 AB の円を描画します。

`circle(x^2+y^2=1)` は、単位円を描画します。

このコマンドを使用して、弧を描画することもできます。

`circle(GA, GB, 0, π/2)` は、AB を直径とする 4 分の 1 の円弧を描画します。

[外接円]

三角形の外接円を描画します。これは三角形に外接する円です。

`circumcircle(point1, point2, point3)`

例：

`circumcircle(GA, GB, GC)` は、ΔABC に外接する円を描画します。

[傍接円]

三角形を定義する 3 点を与えると、最後の 2 点で定義される辺に接し、最初の点を共通の頂点とする 2 辺の延長線にも接する三角形の傍接円を描画します。

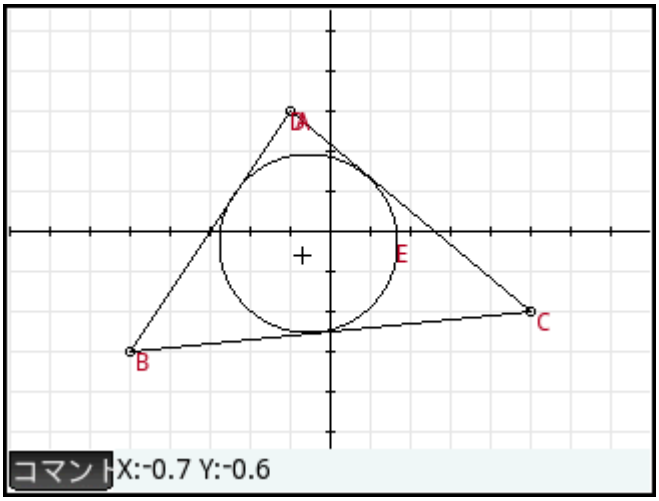
例：

`excircle(GA, GB, GC)` は、線分 BC と、半直線 AB および半直線 AC に接する円を描画します。

[内接円]

内接円とは、多角形の各辺に接する円です。HP Prime では、三角形の各辺に接する内接円を描画できます。

三角形の各頂点をタップして、タップするごとに Enter ≈ を押します。



[楕円]

両方の焦点と、楕円上の点か、楕円上の点から両方の焦点までの距離の一定となる和の半分のスカラー量を指定し、楕円を描画します。

`ellipse(point1, point2, point3)` または `ellipse(point1, point2, realk)`

例：

`ellipse(GA, GB, GC)` は、点 A および点 B を焦点として点 C を通る楕円を描画します。

`ellipse(GA, GB, 3)` は、点 A および点 B を焦点とする楕円を描画します。楕円上の点 P は、AP+BP=6 を満たします。

[双曲線]

両方の焦点と、双曲線上の点か、双曲線上の点から両方の焦点までの距離の一定となる差の半分のスカラー量を指定し、双曲線を描画します。

`hyperbola(point1, point2, point3)` または `hyperbola(point1, point2, realk)`

例：

`hyperbola(GA, GB, GC)` は、点 A および点 B を焦点として点 C を通る双曲線を描画します。

`hyperbola(GA, GB, 3)` は、点 A および点 B を焦点とする双曲線を描画します。双曲線上の点 P は、|AP-BP|=6 を満たします。

[放物線]

焦点と準線を与えるか、放物線の頂点および焦点距離を表す実数を与えると、放物線を描画します。

`parabola(point,line)` または `parabola(vertex,real)`

例：

`parabola(GA, GB)_` は、焦点が点 A で準線が直線 B の放物線を描画します。

`parabola(GA, 1)` は、頂点が点 A で焦点距離が 1 の放物線を描画します。

[円錐]

x と y の式で定義される円錐断面のグラフをプロットします。

conic(expr)

例：

conic(x^2+y^2-81)は、中心(0,0)、半径 9 の円を描画します。

[軌跡]

最初の点と、幾何オブジェクトの要素（その上の点）である 2 番目の点を与えると、2 番目の点がオブジェクト上を移動するときの最初の点の軌跡を描画します。

locus(point,element)

[プロット]

[関数]

独立変数 x の式を与えると、関数のプロットを描画します。小文字の x を使用してください。

構文：plotfunc(Expr)

例：

plotfunc(3*sin(x))は y=3*sin(x)のグラフを描画します

[パラメトリック]

1 変数の複素数の式と、その変数の区間を引数として指定します。複素数の式 f(t)+i*g(t)を x=f(t)および y=g(t)として解釈し、2 番目の引数で指定した区間にわたってパラメトリック方程式をプロットします。

構文：plotparam(f(Var)+i*g(Var), Var= Start..Stop, [tstep=Value])

例：

plotparam(cos(t)+ i*sin(t), t=0..2*π)は単位円をプロットします

plotparam(cos(t)+ i*sin(t), t=0..2*π, tstep=π/3)は、単位円に内接する正六角形をプロットします（tstep 値に注意してください）

[極座標]

極座標プロットを描画します。

構文：plotpolar(Expr,Var=Interval, [Step])または plotpolar(Expr, Var, Min, Max, [Step])

例：

plotpolar(f(x),x,a,b)は、x が[a,b]のときの極座標曲線 r=f(x)を描画します

[数列]

x の式と、3 つの値を含むリストを与え、直線 y=x と、最後の 2 つの値の間の区間として定義される定義域にわたって式によって定義される関数のプロットを描画し、式によって再帰的に定義される（最初の値で開始される）数列の最初の n 項についてのクモの巣状のプロットを描画します。

構文：plotseq(f(Var), Var={Start, Xmin, Xmax}, Integern)

例：

plotseq(1-x/2, x={3 -1 6}, 5)は、y=x および y=1-x/2（x=-1 から x=6）をプロットし、u(0)=3 で始まる u(n)=1-(u(n-1)/2)のクモの巣状のプロットの最初の 5 項を描画します

[陰]

式（x および y）によって陰関数で定義された曲線をプロットします。特に、式=0 についてプロットします。小文字の x および y を使用することに注意してください。オプションの x 区間および y 区間を指定すると、その区間内でのみプロットします。

構文：plotimplicit(Expr, [XIntrvl, YIntrvl])

例：

plotimplicit((x+5)^2+(y+4)^2-1)は、点(-5, -4)を中心とする半径 1 の円をプロットします

[勾配場]

式に含まれる f(x,y)について、微分方程式 y’=f(x,y)の勾配場のグラフをプロットします。ベクトル変数は、変数を含むベクトルです。ベクトル変数が[x=Interval, y=Interval]の形式の場合、勾配場は指定された x 範囲および y 範囲内でプロットされます。x ステップ値と y ステップ値を与えると、これらのステップを使用して勾配場の線分をプロットします。オプションが normalize の場合、描画される勾配場の線分の長さが同じになります。

構文：plotfield(Expr, VectorVar, [xstep=Val, ystep=Val, Option])

例：

plotfield(x*sin(y), [x=-6..6, y=-6..6],normalize)は、y'=x*sin(y)の勾配場を-6 から 6 まで両方向に描画します。線分はすべて同じ長さになります。

[ODE]

変数 Val1、Val2,...の初期条件を含む微分方程式 y’=f(Va1, Var2, ...)の解を描画します。最初の引数は、式 f(Var1, Var2,...)で、2 番目の引数は変数のベクトルで、3 番目の引数は初期条件のベクトルです。

構文：plotode(Expr, [Var1, Var2, ...], [Val1, Val2. ...])

例：

plotode(x*sin(y), [x,y], [-2, 2])は、点(-2, 2)を初期条件として渡す y’=x*sin(y)の解のグラフを描画します

[リスト]

n 個の点のセットをプロットし、これらを線分で結びます。点は 2xn 行列で定義され、最初の行が横座標、2 番目の行が縦座標となります。

構文：plotlist(Matrix 2xn)

例：

plotlist([[0,3],[2,1],[4,4],[0,3]])は三角形を描画します

[スライダー]

パラメーターの値の制御に使用できるスライダー バーを作成します。ダイアログ ボックスに、スライダー バーの定義およびスライダーのアニメーションが表示されます。完了すると、[プロット表示]の左上付近にスライダー バーが表示されます。その後、別の場所に移動できます。

[変換]

[平行移動]

与えられたベクトルに沿って幾何オブジェクトを平行移動します。ベクトルは 2 つの点の差として与えられます（先頭-末尾）。

`translation(vector, object)`

例：

`translation(0-i, GA)` はオブジェクト A を 1 単位だけ下に平行移動します。

`translation(GB-GA, GC)` はオブジェクト C をベクトル AB に沿って平行移動します。

[鏡映]

直線または点に対して幾何オブジェクトを鏡映変換します。後者は半回転と呼ばれることもあります。

`reflection(line, object)` または `reflection(point, object)`

例：

`reflection(line(x=3),point(1,1))` は、垂直線 x=3 に対して点(1, 1)を鏡映変換し、点(5,1)を作成します。

`reflection(1+i, 3-2i)` は、点(1, 1)に対して点(3, -2)を鏡映変換し、点(-1, 4)を作成します。

[回転]

与えられた中心点の周りを与えられた角度で幾何オブジェクトを回転させます。

`rotate(point, angle, object)`

例：

`rotate(GA, angle(GB, GC, GD),GK)` は、K のラベルが付いた幾何オブジェクトについて、点 A の周りを∡CBD に等しい角度だけ回転させます。

[拡大縮小]

ある中心点について、幾何オブジェクトを所定のスケール係数で拡大します。

`homothety(point, realk, object)`

例：

`homothety(GA, 2, GB)` は、点 A を中心としたスケール係数 2 の拡大を作成します。幾何オブジェクト B 上の任意の点 P は、半直線 AP 上に、AP’=2AP を満たす像 P’ を持ちます。

[相似]

幾何オブジェクトを同一の中心点について拡大および回転します。

`similarity(point, realk, angle, object)`

例：

`similarity(0, 3, angle(0,1,i),point(2,0))` は点(2,0)を相似比 3 で拡大し（点(6,0)）、結果を反時計回りに 90°回転させて点(0, 6)を作成します。

[投影]

曲線への点の直交投影を描画します。

`projection(curve, point)`

[反転]

別の点を基準として、スケール係数の大きさで点の反転を描画します。

`inversion(point1, realk, point2)`

例：

`inversion(GA, 3, GB)`は、直線 AB 上に AB*AC=3 を満たす点 C を描画します。この場合、点 A は反転の中心で、スケール係数は 3 です。点 B は、反転が作成される元の点です。

一般的に、点 A をスケール係数 k で中心 C の向こうに反転すると、A の写像が A’に作成されます。A’は直線 CA 上にあり、CA*CA’=k（CA および CA’は対応する線分の長さを示す）を満たします。k=1 の場合、CA の長さは CA’の長さの逆数です。

[逆数変換]

円と、点または直線のオブジェクトのベクトルを与え、その円に関して、それぞれの点をその極線に置き換え、それぞれの直線をその極に置き換えるベクトルを返します。

`reciprocation(Circle, [Obj1, Obj2,...Objn])`

例：

`reciprocation(circle(0,1),[line(1+i,2),point(1+i*2)])`は、`[point(1/2, 1/2) line(y=-x/2+1/2)]`を返します

[数値表示]：[コマンド]メニュー

[デカルト]

[横座標]

点の x 座標またはベクトルの x 方向の長さを返します。

`abscissa(point) or abscissa(vector)`

例：

`abscissa(GA)`は点 A の x 座標を返します。

[縦座標]

点の y 座標またはベクトルの y の長さを返します。

`ordinate(point)`または`ordinate(vector)`

例：

`ordinate(GA)`は点 A の y 座標を返します。

[座標]

複数の点のベクトルを与えると、それらの点の x 座標および y 座標を含む行列を返します。行列の各行は 1 つの点を定義し、最初の列は x 座標、2 番目の列は y 座標を含みます。

coordinates([point1, point2, …, pointn]))

[方程式]

x と y による曲線のデカルト方程式か、点のデカルト座標を返します。

equation(curve)または equation(point)

例：

GA が(0, 0)の点、GB が(1, 0)の点、GC が circle(GA, GB-GA)と定義された場合、equation(GC)は x2 + y2=1 を返します。

[パラメトリック]

[equation]コマンドと同様の機能を持ちますが、複素形式のパラメトリック方程式の結果を返します。

parameq(GeoObj)

[極座標]

点または複素数の極座標を含むベクトルを返します。

polar_coordinates(point)または polar_coordinates(complex)

例：

polar_coordinates(√2, √2)は[2, π/4]を返します

[測定]

[距離]

2 点間または点と曲線との距離を返します。

distance(point1, point2)または distance(point, curve)

例：

distance(1+i, 3+3i)は 2.828…または 2√2 を返します。

GA が点(0, 0)で、GB を plotfunc(4-x^2/4)と定義した場合、distance(GA, GB)は 3.464…または 2√3 を返します。

[半径]

円の半径を返します。

radius(circle)

例：

GA が(0, 0)の点、GB が(1, 0)の点、GC が circle(GA, GB-GA)と定義された場合、radius(GC)は 1 を返します。

[周長]

多角形の外周または円の円周を返します。

perimeter(polygon)または perimeter(circle)

例：

GA が(0, 0)の点、GB が(1, 0)の点、GC が circle(GA, GB-GA)と定義された場合、`perimeter(GC)` は 2p を返します。

GA が(0, 0)の点、GB が(1, 0)の点、GC が square(GA, GB-GA)と定義された場合、`perimeter(GC)` は 4 を返します。

[傾き]

直線オブジェクト（線分、半直線、または直線）の傾きを返します。

`slope(Object)`

例：

`slope(line(point(1, 1), point(2, 2)))` は 1 を返します。

[面積]

円または多角形の面積を返します。

`area(circle)` または `area(polygon)`

このコマンドは、2 点間の曲線の下の面積を返すこともできます。

`area(expr, value1, value2)`

例：

GA が単位円として定義された場合、`area(GA)` は p を返します。

`area(4-x^2/4, -4,4)` は 14.666…を返します

[角度]

指示された角の角度を返します。最初の点は角の頂点として見なされ、次の 2 つの点の順によって、角度と符号を与えます。

`angle(vertex, point2, point3)`

例：

`angle(GA, GB, GC)` は、∡BAC の角度を返します。

[弧長]

曲線上の 2 点間の曲線の弧長を返します。曲線は式で、独立変数が宣言され、2 つの点は独立変数の値によって定義されます。

このコマンドは、曲線のパラメーター定義も受け入れることができます。この場合、式は 3 番目の独立変数についての 2 つの式のリスト（1 番目が x について、2 番目が y について）です。

`arcLen(expr, real1, real2)`

例：

`arcLen(x^2, x, -2, 2)` は 9.29…を返します。

`arcLen({sin(t), cos(t)}, t, 0, π/2)` は 1.57…を返します

[テスト]

[共線]

点のセットを引数として指定し、これらの点が共線であるかどうかを検査します。点が共線であれば 1、そうでない場合は 0 を返します。

```
is_collinear(point1, point2, …, pointn)
```

例：

`is_collinear(point(0,0), point(5,0), point(6,1))` は 0 を返します

[円周上]

点のセットを引数として指定し、これらがすべて同一円上にあるかどうかを検査します。すべての点が同一円上にある場合は 1、そうでない場合は 0 を返します。

```
is_concyclic(point1, point2, …, pointn)
```

例：

`is_concyclic(point(-4,-2), point(-4,2), point(4,-2), point(4,2))` は 1 を返します

[オブジェクト上]

点が幾何オブジェクト上にあるか検査します。その場合、点が含まれている部分を表す数を返します (側面の 1 ~ n の数)。それ以外の場合は 0 を返します。

```
is_element(point, object)
```

例：

`is_element(point(2/√2,2/√2), circle(0,1))` は 1 を返します。

`is_element(point(0,-5), square(point(3,3),point(-5,3))` は 3 を返します。

[平行]

2 つの直線が平行かどうかを検査します。そうであれば 1、そうでない場合は 0 を返します。

```
is_parallel(line1, line2)
```

例：

`is_parallel(line(2x+3y=7),line(2x+3y=9))` は 1 を返します。

[垂直]

[is_orthogonal]に類似します。2 つの直線が垂直に交わるかどうかを検査します。

```
is_perpendicular(line1, line2)
```

[二等辺]

3 つの点を指定し、これらが単一の二等辺三角形の頂点であるかどうかを検査します。そうでない場合は 0 を返します。そうである場合、長さの等しい 2 辺が共有する点の番号（1、2、または 3）を返します。3 点が正三角形をなす場合は 4 を返します。

```
is_isosceles(point1, point2, point3)
```

例：

`is_isoscelesl(point(0,0), point(4,0), point(2,4))`.は 3 を返します。

[等辺形]

3 つの点を指定し、これらが単一の正三角形の頂点であるかどうかを検査します。そうであれば 1、そうでない場合は 0 を返します。

`is_equilateral(point1, point2, point3)`

例：

`is_equilateral(point(0,0), point(4,0), point(2,4))`は 0 を返します。

[平行四辺形]

4 つの点が平行四辺形の頂点かどうかを検査します。そうでない場合は 0 を返します。そうであるとき、平行四辺形の場合は 1、菱形の場合は 2、長方形の場合は 3、正方形の場合は 4 を返します。

`is_parallelogram(point1, point2, point3, point4)`

例：

`is_parallelogram(point(0,0), point(2,4), point(0,8), point(-2,4))`は 2 を返します。

[共役]

与えられた円について、2 つの点または 2 つの直線が共役であるかどうかを検査します。そうであれば 1、そうでない場合は 0 を返します。

`is_conjugate(circle, point1, point2)`または `is_conjugate(circle, line1, line2)`

その他の幾何アプリケーション関数

以下の関数は、幾何アプリケーションのメニューからは利用できませんが、[カタログ]メニューから利用できます。

affix

点またはベクトルの x と y の長さを複素数として返します。

`affix(point)`または `affix(vector)`

例：

GA が点(1, – 2)の場合、`affix(GA)`は 1 – 2i を返します。

barycenter

点の重み（実数）をそれぞれ指定した場合の、点のセットについての仮想的な質量の中心を計算します。個々の点と重みのペアは、ベクトルとして角括弧で囲まれます。

`barycenter([[point1, weight1], [point2, weight2],…,[pointn, weightn]])`

例：

barycenter $\left(\begin{bmatrix} \text{point}(1) & 1 \\ \text{point}(1+i) & 2 \\ \text{point}(1-i) & 1 \end{bmatrix}\right)$ は、点(1, 1/4)を返します。

convexhull

与えられた点のセットの凸包となる点を含むベクトルを返します。

convexhull(point1, point2, …, pointn)

例：

convexhull(0,1,1+i,1+2i,-1-i,1-3i,-2+i) は、[1-3*i 1+2*i -2+ i -1- i]を返します

distance2

2 点間または点と曲線との距離の 2 乗を返します。

distance2(point1, point2) または distance2(point, curve)

例：

distance2(1+i, 3+3i) は 8 を返します。

GA が点(0, 0)で、GB を plotfunc(4-x^2/4)と定義した場合、distance2(GA, GB) は 12 を返します。

division_point

2 つの点 A および点 B と数値係数 k に対し、C-B=k*(C-A)を満たす点 C を返します。

division_point(point1, point2, realk)

例：

division_point(0,6+6*i,4) は点(8,8)を返します

equilateral_triangle

1 辺つまり 2 つの連続する頂点によって定義された正三角形を描画します。3 番目の点は自動的に計算されますが、記号によって定義されません。3 番目の引数として小文字の変数が追加された場合、3 番目の点の座標がその変数に保存されます。三角形の向きは、最初の点から反時計回りになります。

equilateral_triangle(point1, point2) または equilateral_triangle(point1, point2, var)

例：

equilateral_triangle(0,6) は、最初の 2 つの頂点が(0, 0)および(6, 0)である正三角形を描画します。3 番目の頂点は(3,3*√3)として計算されます。

equilateral_triangle(0,6, v) は、最初の 2 つの頂点が(0, 0)および(6, 0)である正三角形を描画します。3 番目の頂点は(3,3*√3)として計算され、これらの座標が CAS 変数 v に保存されます。[CAS 表示]では、v を入力すると、(3,3*√3)に等しい point(3*(√3*i+1))が返されます。

exbisector

三角形を定義する 3 つの点を与えると、最初の点を共通の頂点とする三角形の外角の二等分線を作成します。三角形を[プロット表示]に描画する必要はありません。

`exbisector(point1, point2, point3)`

例：

`exbisector(A,B,C)` は、点 A を共通の頂点とする ΔABC の外角の二等分線を描画します。

`exbisector(0,-4i,4)` は、y=x によって指定される直線を描画します。

extract_measure

幾何オブジェクトの定義を返します。点の場合、定義は点の座標で構成されます。他のオブジェクトの場合、定義は[記号表示]の定義が反映され、定義する点の座標が指定されます。

`extract_measure(Var)`

harmonic_conjugate

3 点の共役調和関数を返します。特に、point1 および point2 に関する point3 の共役調和関数を返します。また、3 つの平行直線または 1 点に集まる直線を受け入れます。この場合、共役調和関数直線の等式を返します。

`harmonic_conjugate(point1, point2, point3)` または `harmonic_conjugate(line1, line2, line3)`

例：

`harmonic_conjugate(point(0, 0), point(3, 0), point(4, 0))` は、point(12/5, 0)を返します

harmonic_division

3 点の共役調和関数を返します。特に、point1 および point2 に関する point3 の共役調和関数を返し、結果を変数 var に保存します。また、3 つの平行直線または 1 点に集まる直線を受け入れます。この場合、共役調和関数直線の等式を返します。

`harmonic_division(point1, point2, point3, var)` または `harmonic_division(line1, line2, line3, var)`

例：

`harmonic_division(point(0, 0), point(3, 0), point(4, 0), p)` は、point(12/5, 0)を返し、それを変数 p に保存します

isobarycenter

点のセットについての仮想的な質量の中心を返します。barycenter と同様の機能を持ちますが、すべての点の重みが同じであることを前提とします。

`isobarycenter(point1, point2, …,pointn)`

例：

`isobarycenter(-3,3,3*√3*i)` は、(0,√3)に等しい point(3*√3*i/3)を返します。

is_harmonic

4 つの点が調和分割または調和列点かどうかを検査します。そうであれば 1、そうでない場合は 0 を返します。

```
is_harmonic(point1, point2, point3, point4)
```

例：

`is_harmonic(point(0, 0), point(3, 0), point(4, 0), point(12/5, 0))` は 1 を返します

is_harmonic_circle_bundle

指定された円が共通外接線を持つ場合は 1、これらの中心がすべて同じである場合は 2、これらがすべて同じ円である場合は 3、それ以外の場合は 0 を返します。

```
is_harmonic_circle_bundle({circle1, circle2, …, circlen})
```

is_harmonic_line_bundle

指定された直線が 1 点に集まる場合は 1、すべて平行な場合は 2、すべて同じ直線の場合は 3、それ以外の場合は 0 を返します。

```
is_harmonic_line_bundle({line1, line2, …, linen}))
```

is_orthogonal

2 つの直線または 2 つの円が直交する（垂直に交わる）かどうかを検査します。2 つの円の場合、交点の接線が直交するかどうかを検査します。そうであれば 1、そうでない場合は 0 を返します。

`is_orthogonal(line1, line2)` または `is_orthogonal(circle1, circle2)`

例：

`is_orthogonal(line(y=x),line(y=-x))` は 1 を返します。

is_rectangle

4 つの点のセットが長方形の頂点であるかどうかを検査します。そうでない場合は 0、そうであれば 1、正方形の頂点である場合は 2 を返します。

```
is_rectangle(point1, point2, point3, point4)
```

例：

`is_rectangle(point(0,0), point(4,2), point(2,6), point(-2,4))` は 2 を返します。

3 つの点のみが引数として指定された場合、これらが直角三角形の頂点かどうかを検査します。そうでない場合は 0 を返します。そうである場合、2 つの直交する辺が共有する点の番号（1、2、または 3）を返します。

`is_rectangle(point(0,0), point(4,2), point(2,6))` は 2 を返します。

is_rhombus

4 つの点が菱形の頂点であるかどうかを検査します。そうでない場合は 0、そうであれば 1、正方形の頂点である場合は 2 を返します。

```
is_rhombus(point1, point2, point3, point4)
```

例：

`is_rhombus(point(0,0), point(-2,2), point(0,4), point(2,2))` は 2 を返します

is_square

4 つの点が正方形の頂点であるかどうかを検査します。そうであれば 1、そうでない場合は 0 を返します。

`is_square(point1, point2, point3, point4)`

例：

`is_square(point(0,0), point(4,2), point(2,6), point(-2,4))` は 1 を返します。

LineHorz

水平線 y=a を描画します。

`LineHorz(a)`

例：

`LineHorz(-2)` は、式が y = −2 の水平線を描画します。

LineVert

垂直線 x=a を描画します。

`LineVert(a)`

例：

`LineVert(-3)` は、式が x = −3 の垂直線を描画します。

open_polygon

点のセットを与えられた順に線分で接続し、多角形を生成します。最後の点が最初の点と同じ場合、多角形は閉じた多角形になり、そうでない場合は開いた多角形になります。

`open_polygon(point1, point2, …, point1)` または `open_polygon(point1, point2, …, pointn)`

orthocenter

三角形の重心を返します。重心は三角形の 3 つの垂線の交点です。引数は、三角形の名前とするか、三角形を定義する共線上に存在しない 3 つの点とすることができます。後者の場合、三角形を描画する必要はありません。

`orthocenter(triangle)` または `orthocenter(point1, point2, point3)`

例：

`orthocenter(0,4i,4)` は(0,0)を返します

perpendicular_bisector

線分の垂直二等分線を描画します。線分は、名前または 2 つの端点によって定義されます。

`perpen_bisector(segment)` または `perpen_bisector(point1, point2)`

例：

`perpen_bisector(GC)`は、線分 C の垂直二等分線を描画します。

`perpen_bisector(GA, GB)`は、線分 AB の垂直二等分線を描画します。

`perpen_bisector(3+2i, i)`は、座標(3, 2)および座標(0, 1)を両端とする線分の垂直二等分線を描画します。つまり、式が y=x/3+1 の直線になります。

point2d

それぞれの点について、x ∈ [-5, 5]および y ∈ [-5, 5]となるように点のセットをランダムに再分散させます。どれかの点が移動すると、タップするか方向キーを押すたびに、すべての点がランダムに再分散します。

`point2d(point1, point2, ⋯, pointn)`

polar

与えられた円の極として与えられた点の極線を返します。

`polar(circle, point)`

例：

`polar(circle(x^2+y^2=1),point(1/3,0))`は x=3 を返します

pole

与えられた円について、与えられた直線の極を返します。

`pole(circle, line)`

例：

`pole(circle(x^2+y^2=1), line(x=3))`は、point(1/3, 0)を返します

powerpc

円と点を与え、点から円の中心までの距離の 2 乗と円の半径の 2 乗との差を返します。

`powerpc(circle, point)`

例：

`powerpc(circle(point(0,0), point(1,1)-point(0,0)), point(3,1))`は 8 を返します

radical_axis

2 つの円に対してすべて同じ powerpc 値を持つ点で構成される直線を返します。

`radical_axis(circle1, circle2)`

例：

`radical_axis(circle(((x+2)²+y²) = 8),circle(((x-2)²+y²) = 8))`は line(x=0)を返します

vector

point1 から point2 へのベクトルを作成します。1 つの点を引数として指定すると、原点がベクトルの末尾として使用されます。

`vector(point1, point2)` または `vector(point)`

例：

`vector(point(1,1), point(3,0))` は(1, 1)から(3, 0)へのベクトルを作成します。

vertices

多角形の頂点のリストを返します。

`vertices(polygon)`

vertices_abca

多角形の頂点のリストを返します。

`vertices_abca(polygon)`

10 スプレッドシート

スプレッドシート アプリケーションは、コンテンツ（数値、テキスト、式など）を入力したり、入力内容に対して特定の操作を実行したりするためのセルをグリッド形式で表示します。

スプレッドシート アプリケーションを開くには、 Apps Info を押して、**[スプレッドシート]**を選択します。

アプリケーションの作成と同様に、それぞれが独自の名前を持つカスタマイズしたスプレッドシートをいくつでも作成できます。カスタマイズしたスプレッドシートも、同じ方法で開きます。 Apps Info を押して、特定のスプレッドシートを選択してください。

1 つのスプレッドシートの最大サイズは 10,000 行 x676 列です。

[数値表示]でアプリケーションが開きます。[プロット表示]や[記号表示]はありません。特定のシステム全体の設定を上書きできる、[記号のセットアップ]表示（ Shift Symb Setup ）があります（これは、[記号のセットアップ]表示での一般的な操作です）。

スプレッドシート アプリケーションでの作業の開始

週末に市場で露店を出すとします。持ち主から委託された家具を販売し、10%の手数料を取ります。露店を出すには土地所有者に 1 日 100 ドルを支払う必要があり、自分の取り分が 250 ドルになるまで露店の営業を続ける予定です。

1. Spreadsheet（スプレッドシート）アプリケーションを開きます。

   Apps Info を押して、**[スプレッドシート]**を選択します。

2. 列 A を選択します。**[A]**をタップするか、カーソル キーを使用してセル A（列 A の見出し）を強調表示します。

3. 「PRICE」と入力し、 名前 をタップして、1 列目全体に「PRICE」という名前を付けます。

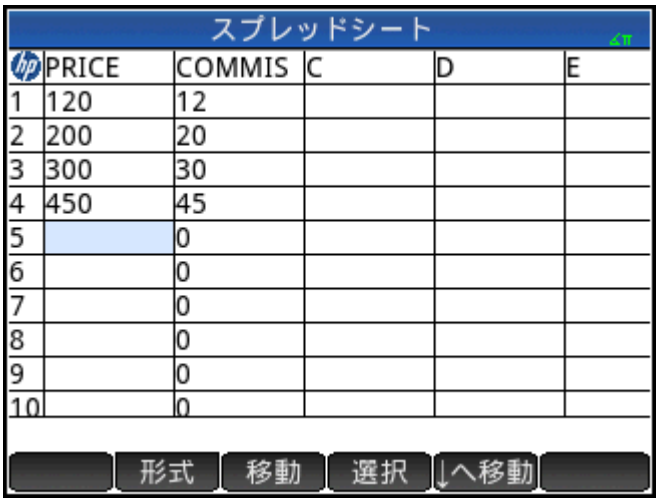
4. 列 B を選択します。**[B]**をタップするか、カーソル キーを使用してセル B を強調表示します。
5. 手数料（各販売品目の価格の 10%）を計算するため、以下の式を入力します。

   Shift • PRICE × 0.1 Enter

   列の見出しに式を入力したので、式はその列にあるすべてのセルに自動的にコピーされます。「PRICE」列には値がまだないため、現時点では 0 のみが表示されています。

6. 列 B を選択します。
7. 形式 をタップして、**[名前]**を選択します。
8. 「COMMIS」と入力し、 OK をタップします。列 B の見出しが「COMMIS」になります。
9. いくつかのダミー値を入力して結果が予想どおりかどうかを確認する方法で、常に式を確認することをおすすめします。セル A1 を選択し、メニューに →へ移動 ではなく ↓へ移動 が表示されていることを確認します（そうなっていない場合は、このボタンをタップします）。このオプションは、コンテンツを入力したセルのすぐ下のセルがカーソルによって自動的に選択されることを意味します。

10. **[PRICE]**列にいくつかの値を追加して、**[COMMIS]**列の結果を確認します。結果が正しく表示されない場合は、**[COMMIS]**見出し→ 編集 の順にタップして、式を修正できます。



11. ダミー値を削除するには、セル**[A1]**を選択し、 選択 をタップして、すべてのダミー値が選択されるまで ▼ を押してから、 Del を押します。

12. セル**[C1]**を選択します。

13. 売上のラベルを入力するために、以下の操作を行います。

Shift • ALPHA 0 TAKINGS Enter

注記：テキスト文字列（名前を除く）は、引用符で囲む必要があります。

14. セル**[C1]**を選択します。

15. 売上を合計するため、以下の式を入力します。

Shift • SUM ( ) PRICE Enter

「A1:A100」のように範囲を指定できますが、列の名前を指定することで、列内のすべての入力内容を合計に確実に含めることができます。

16. セル**[C3]**を選択します。

17. 手数料合計のラベルを入力するため、以下の操作を行います。

Shift • ALPHA 0 TOTAL COMMIS Enter

18. 列 C を広くして C3 のラベル全体を表示するには、列 C の見出しセルを選択し、 形式 をタップして、[列 ↔]を選択します。

その列の必要な幅を指定するための入力フォームが表示されます。

19. 「100」と入力し、 ↔ をタップします。

場合によっては、思いどおりの列の幅になるまで試す必要があります。入力する値は、ピクセル単位の列の幅です。

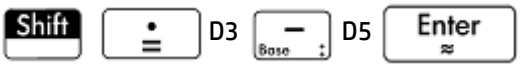
20. セル**[D3]**を選択します。
21. 手数料を合計するため、以下の式を入力します。

> **ヒント：**「SUM」を手動で入力する代わりに、**[アプリケーション]**メニュー（ツールボックス メニューの 1 つ）から選択できます。

22. セル**[C5]**を選択します。
23. 固定費のラベルを入力するために、以下の操作を行います。

24. セル**[D5]**に「100」と入力します。これは、露店用のスペースを借りる際に土地所有者に支払う必要がある金額です。

25. セル**[C7]**にラベル「PROFIT」を入力します。
26. 利益を計算するため、セル**[D7]**に以下の式を入力します。



また、D3 および D5 に名前を付けることもできます。たとえば、各セルを「TOTCOM」および「COSTS」という名前にします。その場合、D7 の式は「=TOTCOM – COSTS」になります。

27. セル**[E1]**にラベル「GOAL」を入力します。

指で画面をスワイプするか、カーソル キーを繰り返し押すと、**[E1]**が表示されます。

28. セル**[F1]**に「250」と入力します。

これは、当日に上げたい最低限の利益です。

29. セル**[C9]**に、ラベル「GO HOME」を入力します。

30. セル**[D9]**に、次の式を入力します。

Shift ≐ D7≥F1 Enter ≈

≥は関係パレット（ Shift 6 ≤,≥,≠ W ）から選択できます。

この式は、**[D9]**に、目標の利益を達成した場合は**[0]**を、達成しなかった場合は**[1]**を表示します。こうすれば、十分な利益を上げたので帰宅できることがすばやく確認できます。

31. **[C9]**および**[D9]**を選択します。

両方のセルを選択するには、指でドラッグするか、**[C9]**を強調表示し、 選択 を選択して、 ▶ を押します。

32. 形式 をタップして、**[色]**を選択します。

33. 選択したセルのコンテンツの色を選択します。

34. 形式 をタップして、**[塗りつぶし]**を選択します。

35. 選択したセルの背景の色を選択します。

これで、スプレッドシート内で最も重要なセルがその他のセルよりも目立つようになります。

これでスプレッドシートは完成ですが、**[PRICE]**列にいくつかのダミー データを追加して、すべての式を確認することをおすすめします。利益が 250 に達すると、**[D9]**に表示される値が**[0]**から**[1]**に変化します。

基本操作

ナビゲーション、選択、およびジェスチャ

スプレッドシートを移動するには、カーソル キーを使用する、スワイプする、または 移動 をタップして移動先のセルを指定するという方法があります。

セルを選択するには、そのセルまで移動するだけです。列の文字をタップして列全体を選択することも、行の番号をタップして行全体を選択することもできます。また、スプレッドシート全体を選択することもできます。スプレッドシートの左上隅にある番号の記されていないセルをタップするだけです （HP のロゴが表示されています）。

セルのブロックを選択するには、選択するブロックの隅のセルを押し、1 秒後に対角線上の反対側のセルまで指をドラッグします。また、隅のセルに移動して 選択 をタップし、カーソル キーを使用して対角線上の反対側のセルに移動することでセルのブロックを選択することもできます。 選択• または別のセルをタップすると選択は解除されます。

セル参照

式の中のあるセルの値を、変数のように参照できます。セルはその列および行の座標で参照され、絶対参照も相対参照も使用できます。絶対参照は、CR と記述します（ここで、C は列番号で、R は行番号です）。したがって、B7 は絶対参照です。式またはそのコピーがどこに配置されていても、式の中では常にセル B7 のデータを参照します。これに対し、B7 は相対参照です。これはセルの相対的位置に基づきます。したがって、たとえば B7 を参照する B8 内の式を C8 にコピーした場合、この式は B7 ではなく C7 を参照します。

セルの範囲は、C6:E12 のように指定することも、列全体（E:E）または行全体（$3:$5）を指定することもできます。列名のアルファベット要素には大文字でも小文字でも使用できますが、g、l、m、および z の列は除きます。G、L、M、および Z は、グラフィック オブジェクト、リスト、行列、および複素数用に予約されている名前です。）これらは前に$が置かれない場合、小文字にする必要がありま

す。したがって、セル B1 は、B1、b1、B1、またはb1 のように参照できますが、M1 は、m1、m1、またはM1 のようにしか参照できません。

セルの名前設定

セル、行、および列には名前を付けられます。この名前は式で使用できます。名前付きのセルでは枠線が青色になります。

方法 1

空のセル、行、または列に名前を付けるには、そのセル、行見出し、または列見出しに移動し、名前を入力して 名前 をタップします。

方法 2

セル、行、または列に名前を付けるには（空かどうかを問いません）、以下の操作を行います。

1. セル、行、または列を選択します。
2. 形式 をタップして、**[名前]**を選択します。
3. 名前を入力し、 OK をタップします。

計算での名前の使用

セル、行、または列に指定する名前は数式で使用できます。たとえば、セルの名前を**[TOTAL]**とすると、別のセルに式「`=TOTAL*1.1`」を入力できます。

以下は、列全体の名前設定に関する比較的複雑な例です。

1. セル**[A]**（列 A の見出しセル）を選択します。
2. 「`COST`」と入力し、 名前 をタップします。
3. セル**[B]**（列 B の見出しセル）を選択します。
4. Shift = COST*0.33 を入力し、 OK をタップします。
5. 列**[A]**に何らかの値を入力し、列**[B]**で計算結果を確認します。

コンテンツの入力

スプレッドシート内にコンテンツを直接入力するか、統計アプリケーションからデータをインポートできます。

直接入力

セルには、実数（3.14）、複素数(a + ib)、整数（#1Ah）、リスト（{1, 2}）、行列またはベクトル（[1, 2]）、文字列（"text"）、単位（2_m）、計算式（つまり式）などの有効な電卓のオブジェクトを入力できます。コンテンツを追加するセルに移動して、ホーム ビューと同様にコンテンツの入力を開始します。入力し終えたら、 Enter を押します。一度の入力で多数のセルにコンテンツを入力することもできます。セルを選択し、コンテンツ（たとえば、「`=Row*3`」）を入力して、 Enter を押すだけです。

入力行に入力した内容は、 Enter を押すとすぐに評価され、結果が 1 つ以上のセルに配置されます。ただし、元になった式を保持する場合は、その式の前に Shift を付けます。たとえば、セル A1（7 を含む）をセル B2（12 を含む）に追加するとします。A4 に「A1 + B2 Enter 」と入力すると 19 になります。A5 に「 Shift A1 + B2」と入力した場合も同様です。ただし、A1（または B2）の値を変更した場合、A5 の値は変更されますが、A4 の値は変更されません。これは、A5 で計算式（式）が保持されたためです。表示された値だけがセルに含まれているか、値を生成した元になっている式も含まれているかを調べるには、セルまでカーソルを移動します。式がある場合は入力行に表示されます。

単一の式で、列内または行内のすべてのセルに内容を追加できます。たとえば、C（列 C の見出しセル）に移動し、「 Shift SIN(Row)」と入力し、 Enter を押します。この列の各セルに、そのセルの行番号の正弦が入力されます。同様のプロセスで、同じ式を使用して行内のすべてのセルに入力することもできます。また、式を一度追加して、それをスプレッドシート内のすべてのセルに適用させることもできます。これを行うには、左上のセル（HP のロゴが表示されたセル）に式を入力します。この機能を確認するため、2 乗から始まる累乗（2 乗、3 乗など）の表を作成するとします。

1. HP のロゴが入ったセル（左上隅）をタップします。または、カーソル キーを使用してそのセルまで移動することもできます（列または行の見出しの選択と同様）。

2. 入力行で、「Shift [=] Row [x^y] Col [+] 1」と入力します。

「Row」および「Col」は組み込み変数です。これらは、これらの変数を含む式が入ったセルの行番号および列番号のプレースホルダーです。

	A	B	C	D	E
1	1	1	1	1	1
2	4	8	16	32	64
3	9	27	81	243	729
4	16	64	256	1,024	4,096
5	25	125	625	3,125	15,625
6	36	216	1,296	7,776	46,656
7	49	343	2,401	16,807	117,64
8	64	512	4,096	32,768	262,14
9	81	729	6,561	59,049	531,44
10	100	1,000	10,000	100,000	1,000,(

3. OK をタップするか Enter を押します。

各列には、行番号の n 乗の値が、二乗を先頭に示されます。したがって、9^5 は 59,049 になります。

データのインポート

Statistics 1Var（1 変数統計）アプリケーションおよび Statistics 2Var（2 変数統計）アプリケーションから（および統計アプリケーションからカスタマイズされた任意のアプリケーションから）データをインポートできます。すぐ下の手順では、Statistics 1Var（1 変数統計）アプリケーションのデータセット D1 がインポートされています。

1. セルを選択します。
2. 「`Statistics_1Var.D1`」と入力します。
3. Enter を押します。

手順 1 で選択したセルから、列に統計アプリケーションからのデータが入力されます。その列の任意のデータはインポート中のデータによって上書きされます。

統計データの入力および編集手順によって、スプレッドシート アプリケーションから統計アプリケーションにデータをエクスポートすることもできます。この手順は、[1 変数統計]アプリケーションと[2 変数統計]アプリケーションの両方でも使用できます。

外部関数

式では、[数学]、[CAS]、[アプリ]、[ユーザー]、または[カタログ]メニューで利用可能なすべての関数を使用できます。たとえば、x = 2 に最も近い 3 − x^2 の根を求めるには、次のようにセルに入力しま

す。 Shift [=] ALPHA alpha ALPHA alpha ROOT ALPHA alpha [( )] 3 [−] ALPHA alpha [×] [x^2] [, Eval] 2 Enter ≈ 。結果は 1.732 になります。

また、メニューから関数を選択することもできます。たとえば、以下の操作を行います。

1. Shift [=] を押します。
2. [Mem] を押して、 CAS をタップします。
3. **[多項式]→[根を求める]**の順に選択します。

   入力行には次のように表示されます。**[=CAS.proot()]**
4. 多項式の係数を、降べきの順にコンマで区切って入力します。

   [+/−] 1 [, Eval] 0 [, Eval] 3
5. Enter ≈ を押して結果を確認します。セルを選択し、 表示 をタップすると、両方の根を含むベクトル[1.732… −1.732…]が表示されます。
6. OK をタップするとスプレッドシートに戻ります。

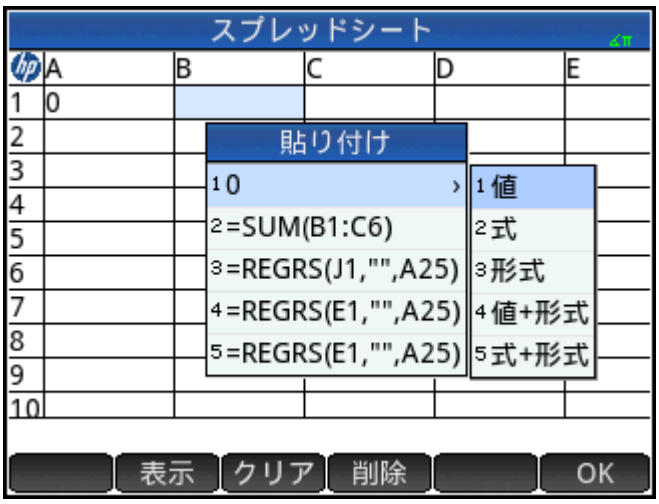
関数に付けられるプレフィックス CAS は、計算が CAS によって実行されることを示すものです（そのため、可能な場合は記号的な結果が返されます）。スプレッドシートの CAS をタップすることで、計算が CAS によって処理されるように強制することもできます。

使用可能なスプレッドシート関数は他にもあります（主に金融および統計の計算に関連する関数）。

コピーおよび貼り付け

1. 1 つ以上のセルをコピーするには、それらを選択して、 Shift [View Copy] を押します。

2. 目的の位置まで移動し、 Shift Menu Paste を押します。



値、式、形式、値と形式の両方、または式と形式の両方を貼り付けすることを選択できます。

また、[スプレッドシート]アプリケーションからデータをコピーし、[統計]アプリケーション、[リスト]エディター、または[行列]エディターに貼り付けることもできます。あるいは、これらのいずれかのアプリケーションからコピーして、[スプレッドシート]アプリケーションに貼り付けることができます。これらの場合、貼り付けが可能なのは値のみです。

CHOOSE コマンドの使用

CHOOSE コマンドは、スプレッドシートのセルにドロップダウン ボックスを定義します。セルの名前は変数名として使用されます。

たとえば、セル A1 にコマンド`=CHOOSE($B$1, "Favorite Color", {"Red", "Green", "Yellow", "Blue"})`を入力した場合、セル A1 はドロップダウン ボックスになります。このセルをタップすると、Favorite Color というタイトルで、Red、Green、Yellow、および Blue のエントリーがあるリストが開きます。Blue をタップすると、Blue は 4 番目のエントリーなので、セル B1 の値は 4 になります。セル B1 に `2` と入力した場合、2 番目のエントリーは Green なので、セル A1 で選択されている値は Green に変わります。

外部参照

参照**[SpreadsheetName.CR]**を使用すると、スプレッドシート アプリケーションの外部からスプレッドシート内のデータを参照できます。たとえば、ホーム表示では、`Spreadsheet.A6` を入力することによって、組み込みのスプレッドシート内のセル A6 を参照できます。したがって、式 6*Spreadsheet.A6 は、組み込みアプリケーションの現在のセル A6 にある値に 6 をかけます。

たとえば、「Savings」という名前のカスタマイズしたスプレッドシートを作成した場合は、5*Savings.A6 のように、そのスプレッドシートを名前で参照します。

外部参照は、5*Savings.TOTAL など、名前付きのセルに対して行えます。

同じ方法で、CAS 内のスプレッドシートのセルへの参照を入力することもできます。

スプレッドシートの外部で作業している場合は、絶対参照によってセルを参照できます。たとえば、`Spreadsheet.$A$6` と入力すると、[スプレッドシート]アプリのセル A6 の内容が返されます。

注記：スプレッドシート名への参照は大文字と小文字を区別します。

変数の参照

セルには任意の変数を挿入できます。これには、ホーム変数、アプリケーション変数、CAS 変数、およびユーザー変数が含まれます。

変数は、参照することも入力することもできます。たとえば、ホーム表示で P に 10 を代入した場合、スプレッドシートのセルに「`=P*5`」と入力して Enter を押すと、50 が表示されます。後で P の値を変更すると、そのセル内の値が新しい値を反映するように自動的に変更されます。これは、参照される変数の例です。

P の現在の値が必要なだけで、P が変化してもその値を変更しない場合は、単に `P` と入力して Enter を押します。これは、入力される変数の例です。

他のアプリケーションで値を指定された変数も、スプレッドシートで参照できます。解くアプリケーションを使用して、方程式を解くことができます。例として、$V^2 = U^2 + 2AD$ を使用します。スプレッドシートに、式として「=V」、「=U」、「=A」、および「=D」を設定した 4 つのセルを作成できます。解くアプリケーションでこれらの変数についてさまざまな値を試すと、入力された値および計算された値がスプレッドシートにコピーされます（これらの値をスプレッドシートでさらに操作できます）。

他のアプリケーションの変数には、特定の計算の結果が含まれています。たとえば、関数アプリケーションで関数をプロットし、2 つの x 値の間の符号付き面積を計算した場合、Vars を押し、アプリ をタップしてから**[関数]→[結果]→[符号付き面積]**の順に選択すると、スプレッドシートでその値を参照できます。

また、多数のシステム変数も利用できます。たとえば、ホーム表示で計算された最後の解答を取得するには、Shift + Ans Enter を入力できます。また、ホーム表示で計算された最後の解答を取得し、ホーム表示で新しい計算が行われたときにその値が自動的に更新されるようにするには、Shift Shift + Ans Enter を入力できます（これは、ホーム表示の Ans でのみ機能し、[CAS 表示]の Ans では機能しないことに注意してください）。

利用可能なすべての変数は、 Vars Chars A を押すと表示される変数メニューに一覧表示されます。

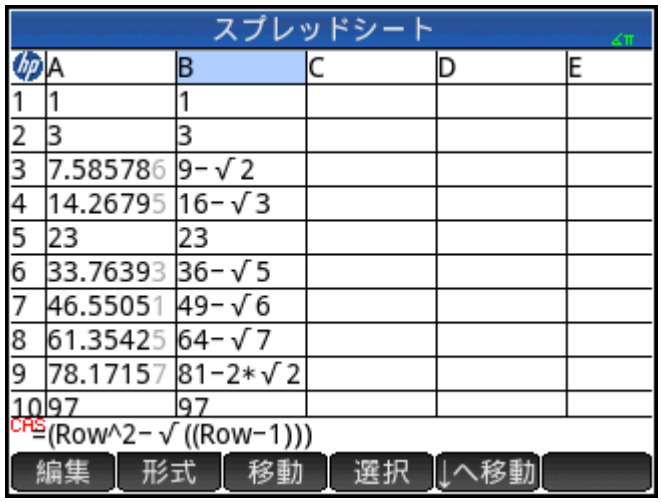
スプレッドシート計算での CAS の使用

スプレッドシート計算が CAS によって実行されるように強制し、結果が必ず記号で表示されるようにすることができます（そのため、正確な結果が表示されます）。たとえば、行 5 に式=√Row を入力すると、CAS によって計算されない場合は 2.2360679775 が、計算された場合は√5 が得られます。

計算エンジンの選択は、式の入力時に行います。式の入力を開始するとすぐに、 形式 ボタンが CAS または CAS• （最後の選択内容によって異なります）に変化します。これはトグル キーです。このキーをタップすると、ボタンが切り替わります。

CAS が表示されている場合、計算は数値で表示されます（有効桁数は電卓の精度によって制限されます）。 CAS• が表示されている場合、計算は CAS によって実行されるため、正確になります。

次の図では、セル A に入力されている式は、セル B に入力されている式= Row2 – √(Row – 1)とまったく同じです。唯一の違いは、B に式を入力しているときには CAS• が表示（または選択）されていたため、計算が強制的に CAS によって実行されたことです。選択したセルに CAS によって計算される式が含まれている場合、入力行に赤色で「CAS」と表示されます。



ボタンおよびキー

ボタンまたはキー	用途
編集	選択されたセルのオブジェクトを編集するために、入力行を有効にします。このボタンは選択したセルに内容がある場合にのみ表示されます
名前	入力行に入力したテキストを名前に変換します。このボタンは入力行が有効である場合にのみ表示されます
CAS / CAS•	式を CAS によって強制的に処理するかどうかを切り替えます。ただし、 CAS のみが式を評価します。このボタンは入力行が有効である場合にのみ表示されます
$	$記号を入力します。このボタンは絶対参照を入力するときのショートカットで、入力行が有効である場合にのみ表示されます

ボタンまたはキー	用途
形式	選択したセル、ブロック、列、行、またはスプレッドシート全体の形式オプションを表示します。217 ページの形式オプションを参照してください
移動	ジャンプ先のセルを指定するための入力フォームを表示します
選択	カーソル キーを使用してセルのブロックを簡単に選択できるように、電卓を選択モードに設定します。これは 選択• に変わり、これをタップするとセルを選択解除できます。押したままドラッグすることでセルのブロックを選択することもできます
↓へ移動 または →へ移動	コンテンツがセルに入力された後にカーソルを動かす方向を設定します
表示	選択したセルの結果を全画面モードで表示します(垂直および水平スクロールを有効にします)。選択したセルに内容がある場合にのみ表示されます
並替	並べ替える列を選択し、昇順または降順で並べ替えることができます。セルが選択されている場合にのみ表示されます
キャンセ	入力をキャンセルし、入力行をクリアします
OK	入力を受け入れて評価します
Shift Esc Clear	スプレッドシートをクリアします

形式オプション

形式オプションは、 形式 をタップすると表示されます。それらは、セル、ブロック、列、行、スプレッドシート全体など、現在選択している項目に適用されます。

オプションは、次のとおりです。

- **[名前]**：選択した項目に名前を付ける入力フォームを表示します。
- **[番号書式]**：[自動]、[標準]、[固定]、[関数]、または[工学]（これは、[ホーム設定]での設定に似ています)。

- **[フォント サイズ]**：自動または 10 から 22 ポイント。
- **[色]**：選択したセルの内容（テキスト、数字など）の色。灰色の点の付いたオプションは[自動]を表します。
- **[塗りつぶし]**：選択したセルを塗りつぶす背景色。灰色の点の付いたオプションは[自動]を表します。
- **[↔ で整列]**：[自動]、[左]、[中央]、[右]の水平調整。
- **[↕ で整列]**：[自動]、[最上段]、[中央]、[最下段]の垂直調整。
- **[列 ↔]**：選択した列の必要な幅を指定するための入力フォームを表示します。スプレッドシート全体または 1 列以上の列全体を選択した場合にのみ利用できます。

  水平方向のピンチ アウトまたはピンチ インのジェスチャによって、選択された列の幅を変更することもできます。
- **[行 ↕]**:選択した行の必要な高さを指定するための入力フォームを表示します。スプレッドシート全体または 1 行以上の行全体を選択した場合にのみ利用できます。

  垂直方向のピンチ アウトまたはピンチ インのジェスチャによって、選択された行の高さを変更することもできます。
- **[" "を表示]**：スプレッドシートの本体で文字列を囲む引用符を表示します。オプションは、[自動]、[はい]、または[いいえ]です。
- **[テキストブック]**：テキストブック形式で式を表示します。オプションは、[自動]、[はい]、または[いいえ]です。
- **[キャッシュ中]**：多くの式を含むスプレッドシートで計算を迅速化する場合にこのオプションをオンにします。スプレッドシート全体を選択した場合にのみ利用できます。

形式パラメーター

各形式属性は、式で参照できるパラメーターによって表されます。たとえば、「=D1(1)」と入力すると、セル D1 の式が返されます（または D1 に式がない場合は、何も返されません）。対応するパラメーターを参照することによって式で取得できる属性の一覧を以下に示します。

パラメーター	属性	結果
0	内容	内容（または空）
1	式	式
2	名前	名前（または空）
3	番号書式	[標準]：0 [固定]：1 [関数]：2 [工学]：3
4	小数点以下の桁数	1 から 11、または指定なし（-1）
5	フォント	0 から 6、または指定なし（-1） 0：10 ポイント、6：22 ポイント

パラメーター	属性	結果
6	背景色	セルを塗りつぶす色（指定しない場合は 32768）
7	前景色	セルの内容の色（指定しない場合は 32768）
8	水平調整	左：0 中央：1 右：2 指定なし：-1
9	垂直調整	最上段：0 中央：1 最下段：2 指定なし：-1
10	文字列を囲む引用符を表示	はい：0 いいえ：1 指定なし：-1
11	テキストブック モード（代数モードにしない）	はい：0 いいえ：1 指定なし：-1

形式属性を取得するだけでなく、該当するセルの式で指定することで、形式属性（またはセルの内容）を設定することもできます。たとえば、どこに G5(1):=6543 を配置しても、セル G5 に 6543 が入力されます。それまで G5 にあったすべての内容が置き換えられます。同様に、「`B3(5):=2`」と入力すると、B3 の内容が強制的に中程度のフォント サイズで表示されます。

スプレッドシート関数

[数学]、**[CAS]**、および**[カタログ]**メニューの関数に加えて、特別なスプレッドシート関数を使用できます。これらは、ツールボックス メニューの 1 つである**[アプリケーション]**メニューにあります。

を押し、 アプリ をタップして、**[スプレッドシート]**を選択します。

関数が依存している値が変化したときに結果が自動的に更新されるようにする場合は、関数の前に必ず等号（

）を付けてください。等号を付けないと、単に現在の値を入力することになります。

11　1 変数統計アプリケーション

1 変数統計アプリケーションは、同時に最大 10 個のデータ セットを格納できます。1 つ以上のデータ セットの 1 変数統計を実行できます。

1 変数統計アプリケーションは、データ入力に使用される[数値表示]から開始できます。[記号表示]は、データを含める列および頻度を含める列の指定に使用されます。

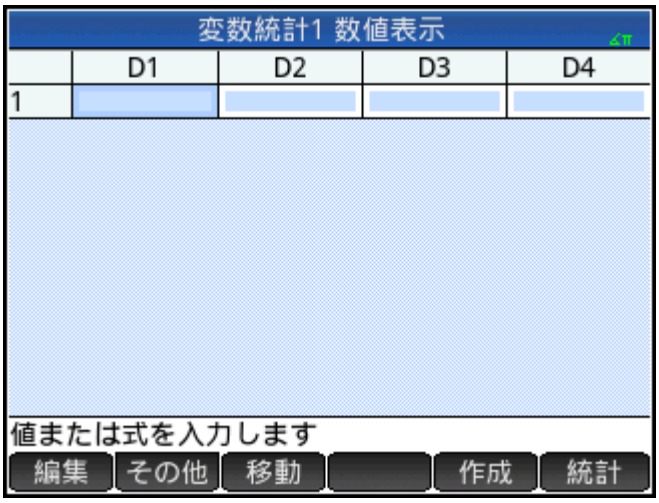
また、ホームで統計を計算したり、特定の統計変数の値を再呼び出したりすることもできます。

1 変数統計アプリケーションで計算された値は変数に保存され、ホーム表示およびその他のアプリケーションで再利用できます。

1 変数統計アプリケーションでの作業の開始

クラス内の学生の身長を測定して、平均身長を求めているとします。最初の 5 人の学生の測定値は、160 cm、165 cm、170 cm、175 cm、および 180 cm です。

1. Apps Info を押して[1 変数統計]を選択し、1 変数統計アプリケーションを開きます。



2. 列 D1 に、次のように測定データを入力します。

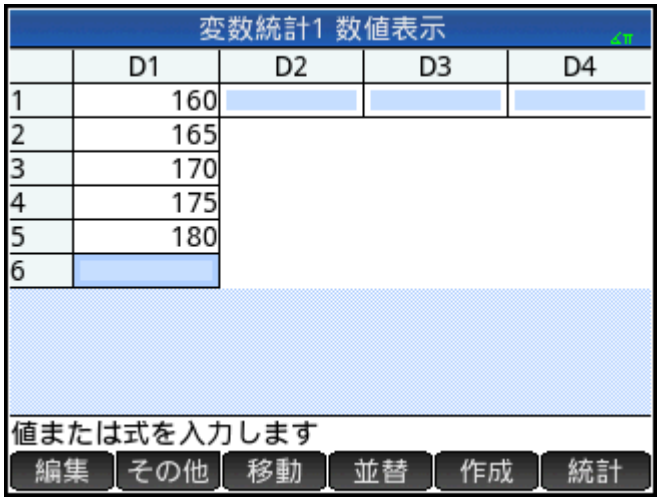
160 Enter

165 Enter

170 Enter

175 Enter

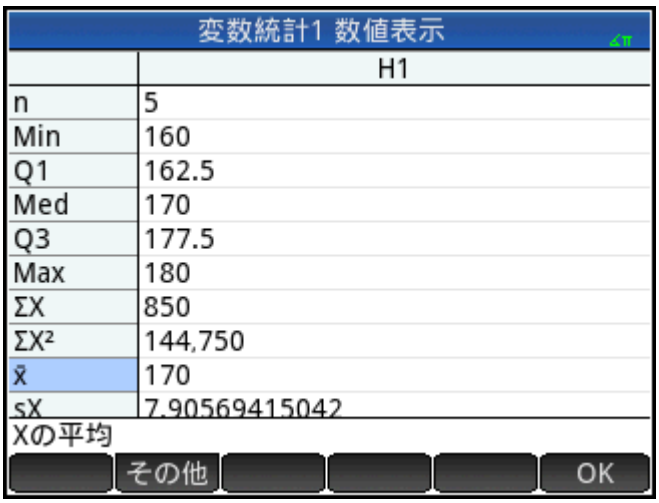
180 Enter



3. 標本の平均値を求めます。

統計 をタップして、D1 の標本データから計算された統計を確認します。平均値（$\bar{x}$）は 170 です。1 つの画面に他の統計が表示される場合もあります。そのため、場合によっては、求めている統計を表示するためにスクロールする必要があります。

統計が表示される列のタイトルが H1 であることに注意してください。1 変数統計では、H1 から H5 の 5 つのデータ セット定義が利用できます。データが D1 に入力されると、H1 はデータとして D1 を使用するように自動的に設定され、各標本の頻度は 1 に設定されます。アプリケーションの[記号表示]から、その他のデータ列を選択できます。



4. OK をタップし、統計ウィンドウを閉じます。

5. Symb Setup を押してデータ セット定義を表示します。

各定義セットの最初のフィールドでは、分析対象のデータの列を指定します。2 番目のフィールドでは、各標本の頻度が入った列を指定します。そして 3 番目のフィールド[プロット n]（n は数字）では、[プロット表示]でデータを表すプロットのタイプ（ヒストグラム、箱ひげ図、正規確率、線、棒、またはパレート）を選択します。

[記号表示]：メニュー項目

[記号表示]でタップできるメニュー項目は次のとおりです。

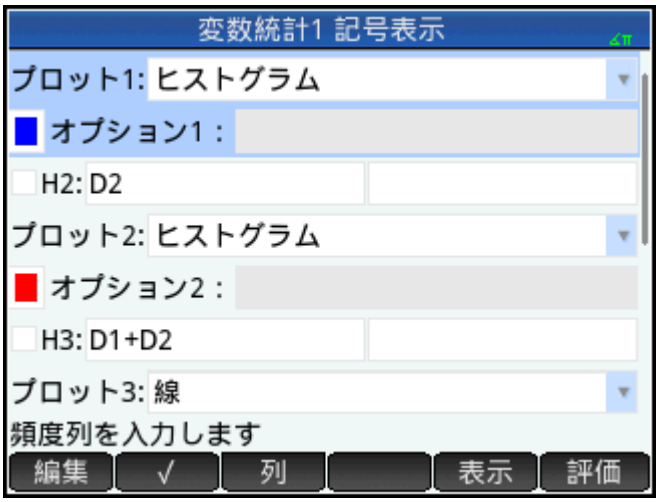
メニュー項目	用途
編集	編集するために、列の変数（または変数式）を入力行にコピーします。完了したら OK をタップします
√	対象となる統計分析（H1 から H5）を選択（または選択解除）します
列	[数値表示]での列名を選択します
表示	現在の式を、テキストブック形式で全画面表示します。完了したら OK をタップします
評価	強調表示された式を評価し、他の定義への参照をすべて解決します

ここでの例を続けるために、クラスの残りの学生の身長を測定し、それぞれの測定値を、最初に記録した 5 つの値のうち最も近い値に丸めるとします。D1 に新しいデータをすべて入力する代わりに、D1 にある 5 つの標本の頻度が入った列 D2 を追加します。

身長（cm）	頻度
160	5
165	3
170	8
175	2
180	1

1. H1 の右にある**[統計]**をタップします（または、 ▶ を押して H1 の 2 つ目のフィールドを強調表示します）。

2. 列 をタップして使用可能な D*n* リストを表示し、**[D2]**を選択します。



3. オプションで、グラフの色を選択します。
4. [記号表示]で複数の分析が定義されている場合は、現在対象ではない分析の選択をすべて解除します。
5. [数値表示]に戻ります。

6. 列 D2 に、前の表で示した頻度データを入力します。

▶ 5 Enter

3 Enter

8 Enter

2 Enter

1 Enter

変数統計1 数値表示

	D1	D2	D3	D4
1	160	5		
2	165	3		
3	170	8		
4	175	2		
5	180	1		
6				

値または式を入力します

編集 その他 移動 並替 作成 統計

7. 統計を再計算するには、 統計 をタップします。

平均身長は約 167.631 cm になりました。

変数統計1 数値表示

	H1
n	19
Min	160
Q1	160
Med	170
Q3	170
Max	180
ΣX	3,185
ΣX²	534,525
x̄	167.631578947
sX	5.86146100782

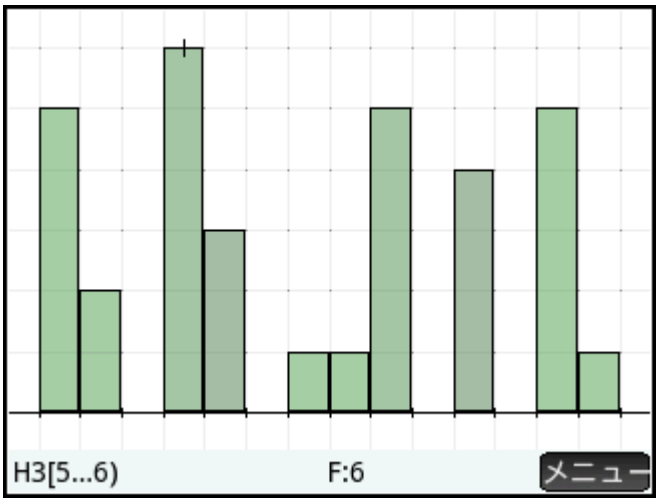
Xの平均

その他 OK

8. データのヒストグラム図を構成します。 OK をタップしてから、 Shift Plot Setup を押します。

データに適したパラメーターを入力します。次の図に示されているパラメーターを入力すると、この例のすべてのデータが[プロット表示]に表示されるようになります。

9. データのヒストグラムをプロットするには、 Plot Setup を押します。



トレーサーを移動して各ビンの区間および頻度を確認するには、 ◀ および ▶ を押します。タップしてビンを選択することもできます。[プロット表示]をスクロールするには、タップしてドラッグします。 + Ans ; または − Base : を押して、カーソルの位置でそれぞれ拡大または縮小することもできます。最後に、2 本指ピンチ ズーム ジェスチャを垂直、水平、または斜め方向に実行して、拡大または縮小を実行できます。

統計データの入力および編集

[数値表示]の各列はデータセットで、D0 ~ D9 までの名前を持つ変数によって表されます。データを列に入力するには、次の 3 つの方法があります。

- [数値表示]に移動し、データを直接入力します。例については、220 ページの 1 変数統計アプリケーションでの作業の開始を参照してください。
- ホーム表示に移動し、リストからデータをコピーします。たとえば、ホーム表示で「L1 Sto ▸ D1」と入力すると、リスト L1 内の項目が 1 変数統計アプリケーションの列 D1 にコピーされます。
- ホーム表示に移動し、スプレッドシート アプリケーションからデータをコピーします。たとえば、目的のデータがスプレッドシート アプリケーションの A1:A10 にあり、そのデータを列 D7 にコピーするとします。1 変数統計アプリケーションが開いている状態で、ホーム表示に戻り、「Spreadsheet.A1:A10 Sto ▸ D7 Enter ≈ 」と入力します。

どの方法でも、入力したデータは自動的に保存されます。いったんこのアプリケーションを離れてから戻っても、最後に入力したデータは引き続き利用できます。

データの入力後は、[記号表示]でデータ セット（およびそれらのデータ セットをプロットする方法）を定義する必要があります。

[数値表示]：メニュー項目

[数値表示]でタップできるメニュー項目は次のとおりです。

編集	選択した項目を編集可能にするために入力行にコピーします
その他	編集オプションのメニューを表示します。227 ページの[その他]メニューを参照してください
移動	指定したリスト項目にカーソルを移動します
並替	さまざまな方法でデータを並べ替えます。229 ページのデータ値の並べ替えを参照してください
作成	指定した列の値のリストを生成する式を入力するための入力フォームを表示します。229 ページのデータの生成を参照してください
統計	[記号表示]で選択されている各データ セットの統計を計算します。229 ページの計算される統計を参照してください

[その他]メニュー

[その他]メニューにはデータのリストを編集するためのオプションが含まれています。オプションの説明を次の表に示します。

オプション	サブオプション	用途
[挿入]	[行]	選択されたリストに新しい行を挿入します。新しい行には要素として 0 が含まれます
[削除]	[列]	選択されたリストの内容を削除します 特定の項目を削除するには、その項目を選択して Del を押します
[選択]	[行]	現在選択されているセルを含む行を選択します。その後、行全体をコピーできます

オプション	サブオプション	用途
	[ボックス]	開始位置および終了位置を指定して矩形選択するためのダイアログ ボックスが開きます。開始位置となるセルをタップしたまま選択し、指でドラッグして要素を矩形選択することもできます。選択後、その要素をコピーできます
	[列]	現在のリストを選択します。選択後、そのリストをコピーできます
[選択]		選択モードのオン/オフを切り替えます 選択モードがオフの場合、セルをタップしたまま指でドラッグして、複数のセルを矩形選択することもできます
[スワップ]	[列]	2 つの列（リスト）の内容を入れ替えます

データ セットの編集

[数値表示]で、変更するデータを強調表示し、新しい値を入力して、 Enter を押します。データを強調表示し、 編集 をタップして入力行にデータをコピーしてから、変更を行って Enter を押すこともできます。

データの削除

- データ項目を削除するには、項目を強調表示して Del を押します。削除されたセルの下にある値が 1 行分上に移動します。
- データ列を削除するには、その列のエントリを強調表示して Shift Esc Clear を押します。列を選択して OK をタップします。
- すべての列にあるすべてのデータを削除するには、 Shift Esc Clear を押し、**[すべての列]**を選択して、 OK をタップします。

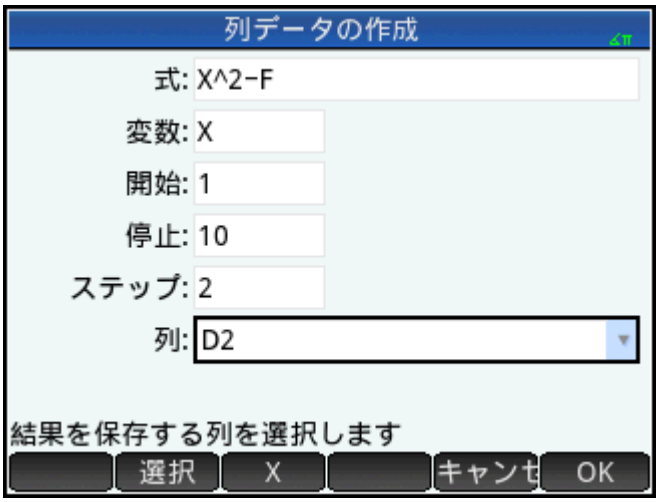
データの挿入

1. 値を挿入する場所の下にあるセルを強調表示します。
2. その他 をタップし、**[挿入]**を選択して、**[行]**を選択します。
3. 値または式を入力して、 Enter を押します。

データ セットにデータを追加するだけで、追加する場所が重要ではない場合には、データ セットの最後のセルを選択して、新しいデータの入力を開始します。

データの生成

作成 をタップすることによって、指定した列の標本のリストを生成する式を入力できます。次の例では、5 つの標本が列 D2 に入力されます。これらは、式 $X^2 - F$ によって生成されます。ここで、X は{1, 3, 5, 7, 9}の集合からもたらされます。これらは 1 から 10 の間の 1 つおきの値です。F は、どこか他の場所（ホーム表示など）でそれに割り当てられている値です。F がたまたま 5 であった場合、列 D2 には{−4, 4, 20, 44, 76}が入力されます。



データ値の並べ替え

選択した独立列に基づいて、最大 3 列のデータを一度に並べ替えることができます。

1. [数値表示]で、並べ替える列に強調表示バーを配置し、 並替 をタップします。
2. **[昇順]**または**[降順]**の並び順を指定します。
3. 独立データ列および従属データ列を指定します。並べ替えは独立列を基準として行われます。たとえば、年齢が C1 にあって収入が C2 にあり、収入によって列を並べ替える場合は、C2 を独立列に、C1 を従属列にします。
4. 任意の頻度データ列を指定します。
5. OK をタップします。

独立列が指定されているとおりに並べ替えられ、その他すべての列は独立列に合わせて並べ替えられます。1 列のみを並べ替えるには、**[従属]**列および**[頻度]**列として**[なし]**を選択します。

計算される統計

統計 をタップすると、[記号表示]で選択された各データセットについて次の結果が表示されます。

統計	定義
n	標本の数
Min（最小）	最小値
Q1	第 1 四分位数：中央値よりも小さい値の中央値

統計	定義
Med（中央）	中央値
Q3	第 3 四分位数：中央値よりも大きい値の中央値
Max（最大）	最大値
ΣX	データ値（その頻度を考慮）の和
ΣX^2	データ値の二乗の和
$\bar{x}$	平均
sX	標本標準偏差
σX	母標準偏差
serrX	標準誤差
ssX	X の偏差平方和

データ セットに含まれている値の数が奇数の場合、Q1 および Q3 の計算の際に中央値は使用されません。たとえば、データ セット{3,5,7,8,15,16,17}の場合、Q1 の計算には最初の 3 項目（3、5、および 7）のみが使用され、Q3 の計算には最後の 3 項目（15、16、および 17）のみが使用されます。

プロット

次をプロットできます。

- ヒストグラム
- 箱ひげ図（異常値あり/なし）
- 正規確率プロット
- 線プロット
- 棒グラフ
- パレート図
- コントロール図
- ドット図
- 茎葉図
- 円グラフ

データを入力し、データ セットを定義したら、データをプロットできます。最大 5 つのグラフを同時にプロットできます。複数のグラフをプロットする場合、 を押して**[自動スケール]**を選択することで、初期ウィンドウをセットアップします。次に、指でパンおよびズームを行って、グラフの表示が最適になるようにします。

統計データのプロット

1. [記号表示]で、プロットするデータ セットを選択します。
2. **[プロット n]**（n は数字）メニューからプロット タイプを選択します。

3. すべてのプロット（特にヒストグラム）について、[プロットのセットアップ]表示でプロットのスケールおよび範囲を調整します。ヒストグラムの棒が太すぎるか細すぎる場合は、**[H 幅]**の設定を変更することで調整できます（236 ページのプロットのセットアップを参照してください）。

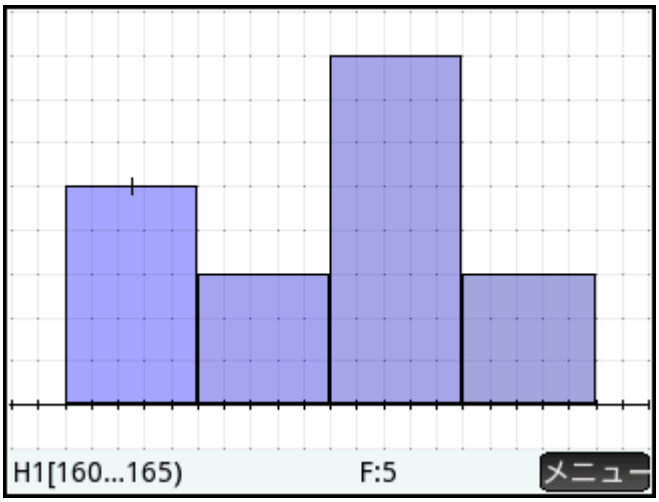
4. Plot Setup を押します。スケールが適切でない場合は、View Copy を押して**[自動スケール]**を選択します。

   自動スケールは、スケールの出発点として利用できます。後でこのスケールを[プロットのセットアップ]表示で直接調整するか、[プロットのセットアップ]表示で調整することができます。

プロット タイプ

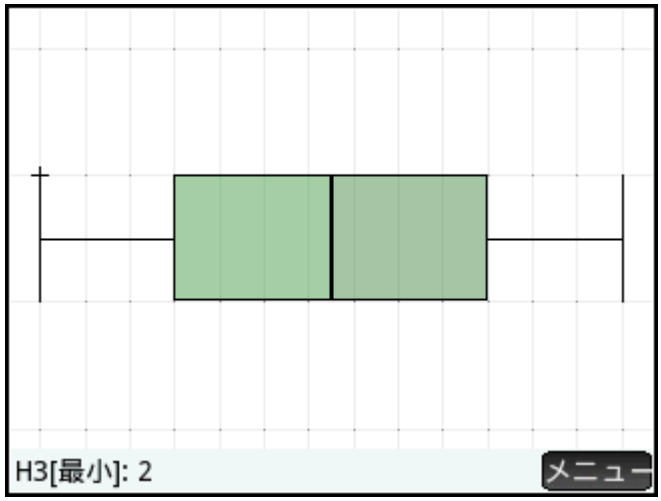
ヒストグラム

プロットの下にある最初の一連の数値は、カーソルの位置を示しています。次の例では、カーソルは 5 と 6 の間（ただし 6 を除きます）のデータに対応するビンにあり、そのビンの頻度は 6 です。このデータ セットは、[記号表示]の H3 によって定義されています。その他のビンに関する情報を表示するには、◀ または ▶ を押します。



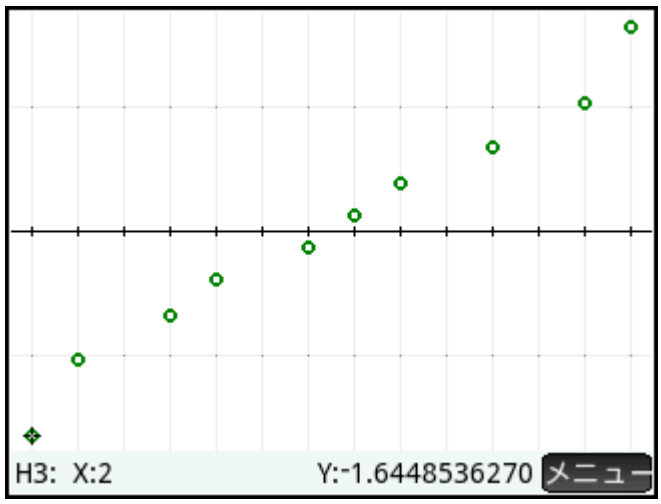
箱ひげ図

左側のひげは、最小のデータ値を示します。箱は、第 1 四分位数、中央値、および第 3 四分位数を示します。右側のひげは、最大のデータ値を示します。プロットの次にある数値によって、カーソルの位置の統計を確認できます。その他の統計を表示するには、◀ または ▶ を押します。[記号表示]では、異常値を含めるか除外することができます。**[選択肢]**フィールドで、**[異常値を表示します]**を選択してプロットの外側の異常値を表示するか、**[異常値なし]**を選択してすべての異常値をデータセットに含めます。



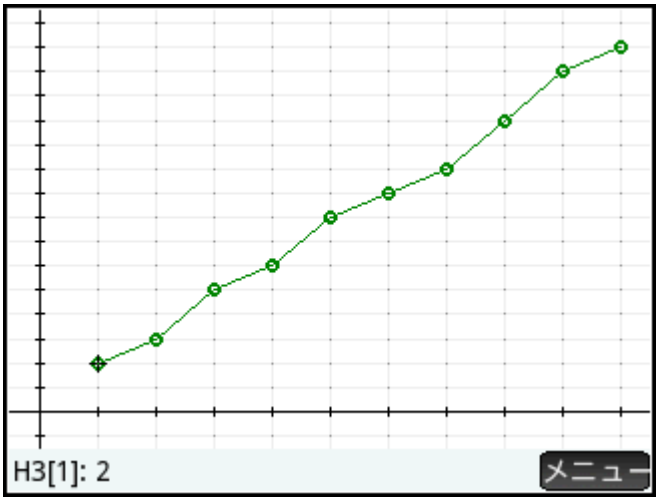
正規確率プロット

正規確率プロットは、標本データがほぼ正規分布しているかどうかの判別に使用されます。データの表示が線形になるほど、データが正規分布している可能性が高くなります。



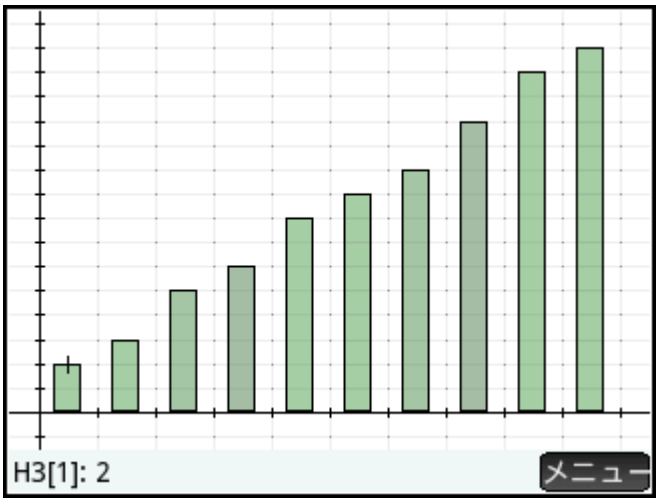
線プロット

線プロットは、(x, y)という形式の点を接続します。ここで x は標本の行番号で、y はその値です。



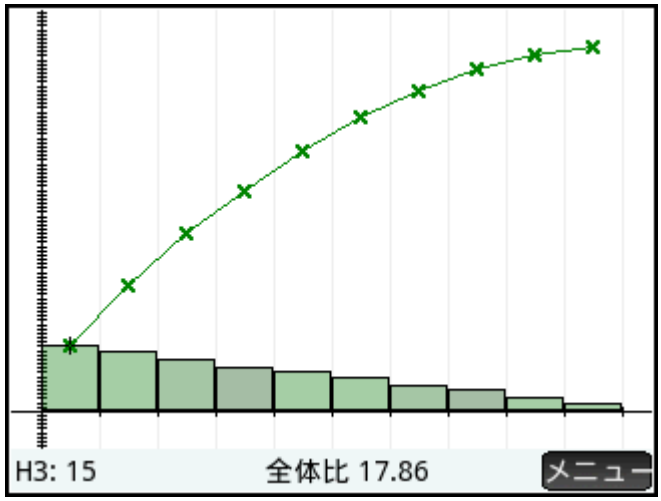
棒グラフ

棒グラフは、x 軸に沿って標本の行番号の位置に配置された垂直の棒として、標本の値を示します。



パレート図

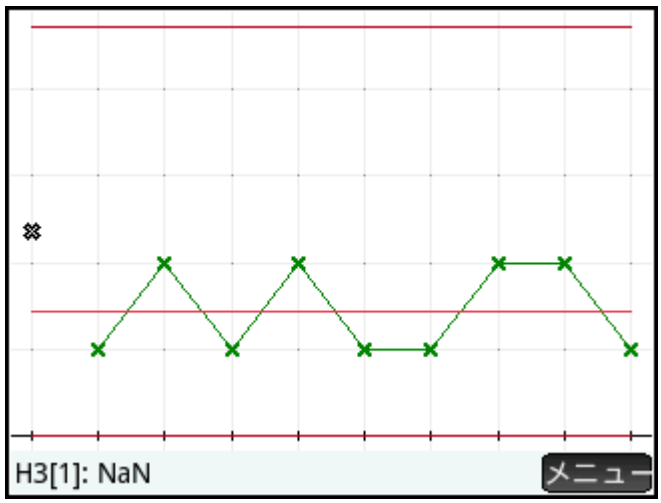
パレート図は、データを降順に配置し、それぞれのデータを全体に占める割合とともに表示します。



コントロール図

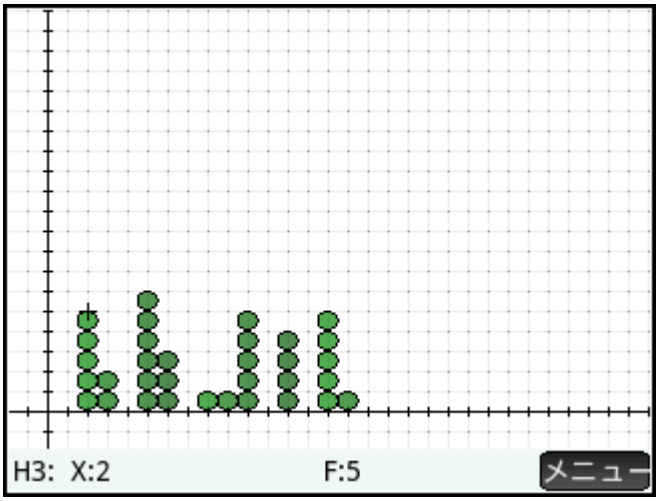
コントロール図では、平均と上下両方の信頼水準に水平線が引かれます。次にデータを順次プロットし、標本を直線で結びます。このプロット タイプには、個々の標本の代わりに移動範囲（標本のペアの差）をプロットするオプションがあります。

[オプション]ボックスでは、**[個体]**または**[移動範囲]**のどちらかを選択できます。



ドット図

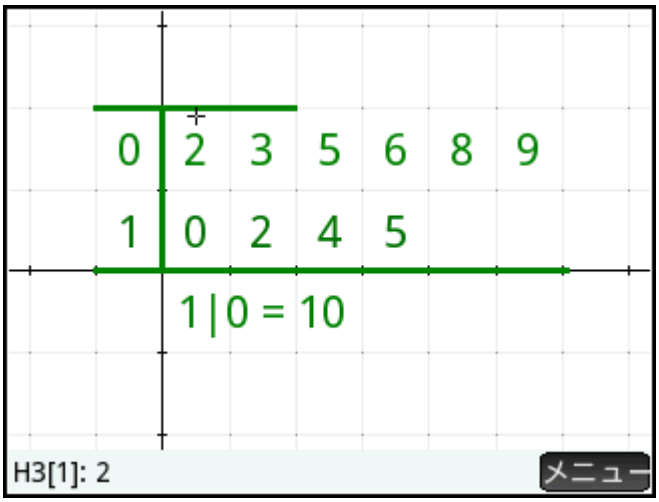
ドット図は各標本に対してドットを描画し、同一の標本を垂直に積み重ねます。



茎葉図

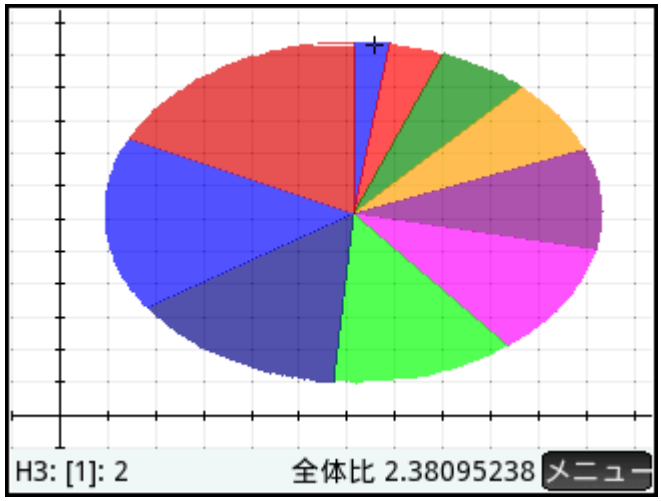
茎葉図は、10 のべき乗ごとに値を分離し、各標本について、茎は最も大きい 10 のべき乗を示し、葉は次に低い 10 のべき乗を表示します。凡例はプロットの下部に表示されます。

[オプション]ボックスでは、**[茎の分割]**または初期設定の**[単一の茎]**のどちらかを選択できます。[茎の分割]オプションでは、それぞれの茎が 5、50 などで 2 つの部品に分割されます。



円グラフ

円グラフは各標本を円の扇形として表示し、扇形の面積は、個々の標本が表すデータ セット全体に占める割合に対応します。



プロットのセットアップ

[プロットのセットアップ]表示（ Shift Plot Setup ）では、その他のアプリケーションと同じプロットパラメーターの多くを指定できます（[X 範囲]や[Y 範囲]など）。1 変数統計アプリケーションに固有の設定としては、次の 2 つがあります。

- **[H 幅]** : **[H 幅]**では、ヒストグラムのビンの幅を指定できます。これによって、ディスプレイに表示されるビンの数およびデータの分布方法（つまり、各ビンに含まれる標本の数）が決まります。
- **[H 範囲]** : **[H 範囲]**では、一連のヒストグラムのビンに対応する値の範囲を指定できます。範囲とは、最も左のビンの左端から最も右のビンの右端までのことです。

グラフの操作

[プロット表示]（ Plot Setup ）には、座標表示だけでなく、ズームおよびトレースのオプションもあります。[自動スケール]オプションは、[表示]メニュー（ View Copy ）および ズーム メニューから利用できます。[表示]メニューでは、グラフを分割画面モードで表示することもできます。

すべてのプロット タイプについて、タップしてドラッグすると[プロット表示]をスクロールできます。2 本指のピンチ ズームを水平方向に実行して x 軸を拡大および縮小したり、垂直方向に実行して y 軸を拡大および縮小したり、斜め方向に実行して両方の軸を拡大および縮小したりできます。 + Ans : または − Base : を押して、カーソルの位置でそれぞれ拡大または縮小することもできます。

[プロット表示] : メニュー項目

[プロット表示]でタップできるメニュー項目は次のとおりです。

ボタン	用途
ズーム	[ズーム]メニューを表示します
トレース	トレース モードをオンまたはオフにします

ボタン	用途
定義	現在の統計プロットの定義を表示します
メニュー	メニューの表示または非表示を切り替えます

12　2 変数統計アプリケーション

2 変数統計アプリケーションは、同時に最大 10 個のデータ セットを格納できます。1 つ以上のデータ セットの 2 変数統計を実行できます。

2 変数統計アプリケーションは、データ入力に使用される[数値表示]から開始できます。[記号表示]は、データを含める列および頻度を含める列の指定に使用されます。

また、ホームおよびスプレッドシート アプリケーションで統計を計算することもできます。

2 変数統計アプリケーションで計算された値は、変数に保存されます。これらは、ホーム表示およびその他のアプリケーションで参照できます。

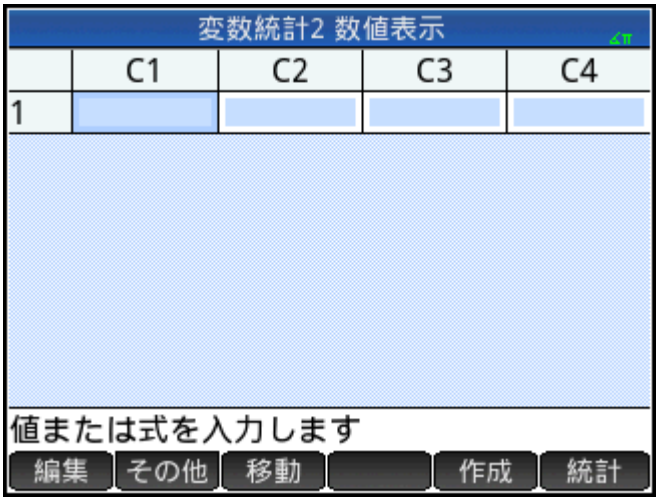
2 変数統計アプリケーションでの作業の開始

次の例では、以下の表の広告および売り上げデータを使用します。この例では、データを入力し、要約統計を計算して、曲線をデータにフィットさせ、広告の増加が売り上げにもたらす影響を予測します。

広告の時間（分） （独立、x）	売り上げ結果（ドル） （従属、y）
2	1400
1	920
3	1100
5	2265
5	2890
4	2200

2 変数統計アプリケーションを開く

▲ Apps Info を押して**[2 変数統計]**を選択し、2 変数統計アプリケーションを開きます。



データの入力

1. 列 C1 に、次のように広告の時間データ（分）を入力します。

   2 Enter 1 Enter 3 Enter 5 Enter 5 Enter 4 Enter

2. 列 C2 に、次のように売り上げ結果データを入力します。

   1400 Enter

   920 Enter

   1100 Enter

   2265 Enter

   2890 Enter

   2200 Enter

データ列およびフィットの選択

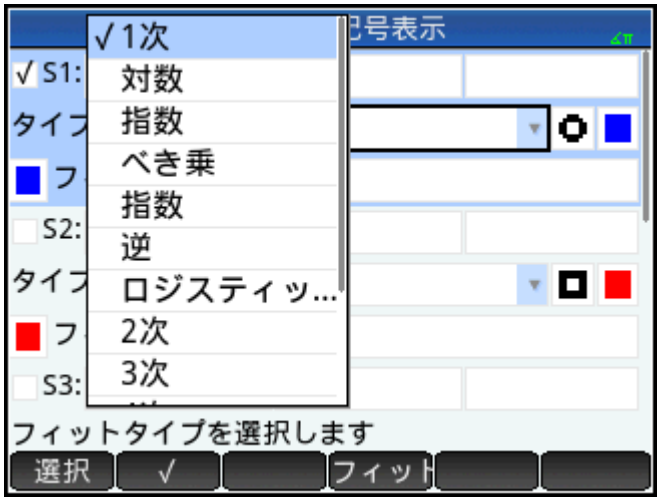
[記号表示]では、S1 ~ S5 までの名前を持つ、最大 5 つの 2 変数データの分析を定義できます。この例では、1 つの分析（S1）のみを定義します。このプロセスには、データ セットおよびフィット タイプの選択が含まれます。

1. Symb Setup を押して、分析するデータが含まれている列を指定します。

   この例では、初期設定で C1 および C2 が表示されます。ただし、C1 および C2 以外の列にデータが入力されている可能性もあります。

2. フィットを選択します。

   [タイプ 1]ボックスからフィットを選択します。この例では、**[1 次]**を選択します。



3. オプションで、散布図の点のタイプと色を選択します。
4. オプションで、**[フィット]**の左にある色メニューを使用して、フィットのグラフの色を選択します。
5. [記号表示]で複数の分析が定義されている場合は、現在対象ではない分析の選択をすべて解除します。

統計の操作

1. 以下の操作を行って、広告の時間と売り上げの相関 r を求めます。

   Num Setup 統計

   相関は r=0.8995… です。

2. 以下の操作を行って、広告の時間の平均値（$\bar{x}$.）を求めます。

X

広告の時間の平均値（$\bar{x}$）は 3.33333…分です。

3. 売り上げの平均値（$\bar{y}$）を求めます。

Y

売り上げの平均値（$\bar{y}$）は約 1,796 ドルです。

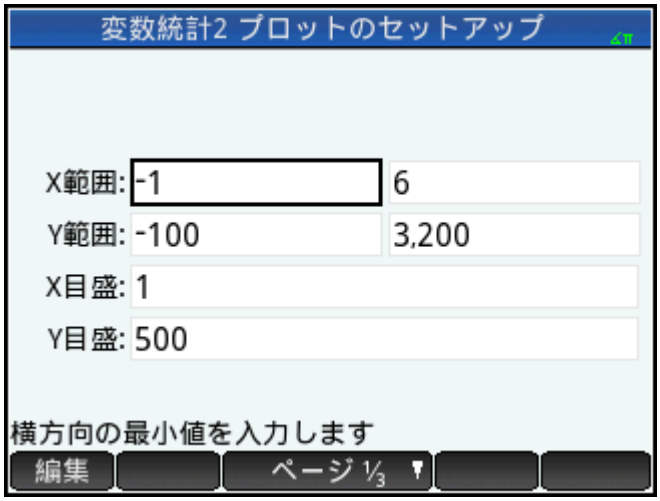
OK を押して[数値表示]に戻ります。

プロットのセットアップ

▲ プロット範囲を変更して、すべての標本がプロットされるようにします。

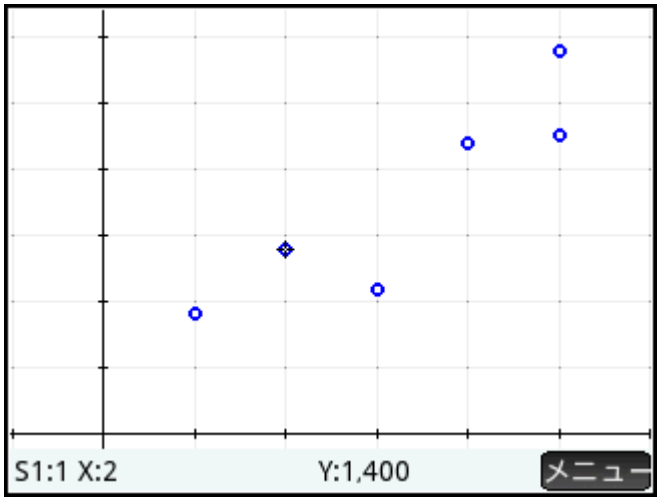
Shift Plot Setup +/− 1 Enter ≈ 6 Enter ≈ +/− 100 Enter ≈ 3200 Enter ≈

▼ 500 Enter ≈



グラフのプロット

1. Plot Setup を押して、グラフをプロットします。

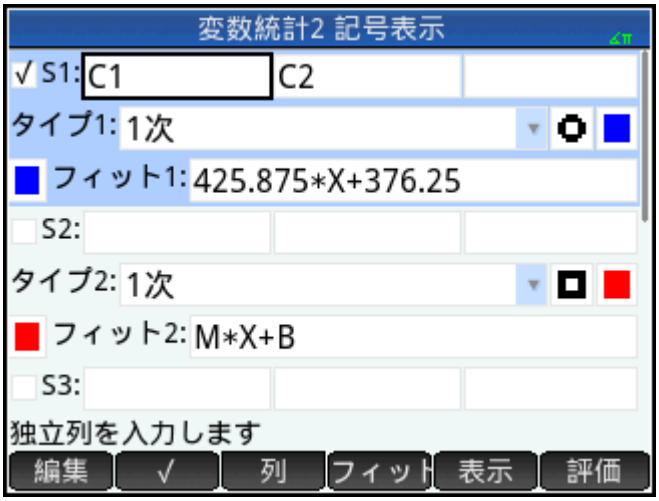


2. メニュー、フィット の順にタップして、フィットをプロットします。

方程式の表示

▲ Symb Setup を押して、[数値表示]に戻ります。

[フィット 1]フィールドの式に注目してください。この式は、回帰直線の傾き（m）が 425.875 で、y 切片（b）が 376.25 であることを示しています。



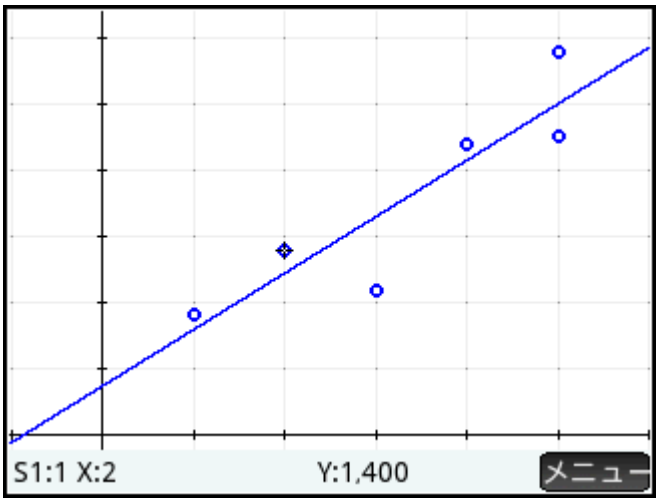
値の予測

ここでは、広告の時間を 6 分まで延ばした場合の売上高を予測しましょう。

1. Plot Setup を押して[プロット表示]に戻ります。

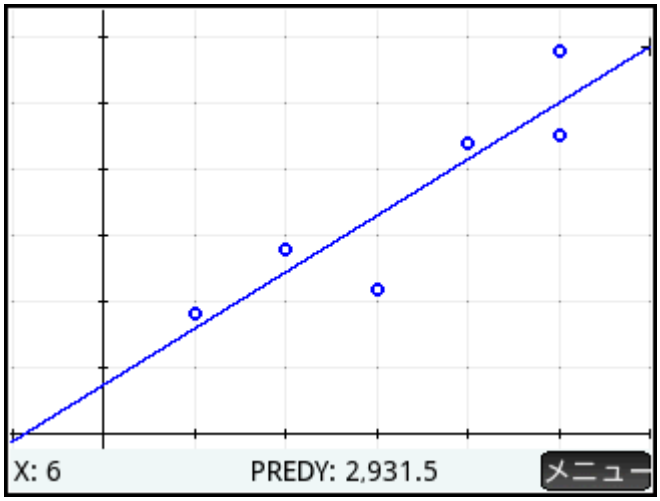
   トレース オプションは初期設定で有効です。このオプションによって、◀ または ▶ を押すとカーソルが標本間を移動します。標本間を移動すると、対応する x 値および y 値が画面の下部に表示されます。この例では、x 軸が広告の時間（分）を表し、y 軸が売り上げを表します。

   しかし、6 分に対応する標本は存在しません。そのため、カーソルを x = 6 に移動することはできません。その代わりに、存在するデータに基づいて、x = 6 の場合の y の値を予測する必要があります。これを行うには、存在する標本ではなく、回帰曲線をトレースする必要があります。



2. ▲ または ▼ を押して、標本ではなく回帰直線をトレースするようにカーソルを設定します。

標本上にあったカーソルが回帰曲線上にジャンプします。



3. x = 6 付近（ディスプレイの右端付近）の回帰直線をタップします。次に、x = 6 になるまで ▶ を押します。画面の左下隅に x 値が表示されない場合は、Menu Paste をタップします。x = 6 に達すると、**[PREDY]**値（この値も画面の下部に表示されます）が 2931.5 を示すことがわかります。したがって、このモデルによれば、広告を 6 分に延ばすと売り上げが 2,931.50 ドルまで伸びることが予測されます。

ヒント：同じトレース手法を利用することで、指定した売上高の達成に必要なおおよその広告時間（分）を予測できます。ただし、より正確な方法が利用できます。具体的には、ホーム表示に戻り、`PredX(s)` と入力します（ここで、s は売上高です）。PredY および PredX はアプリケーション関数です。

統計データの入力および編集

[数値表示]の各列はデータセットで、C0 ~ C9 までの名前を持つ変数によって表されます。データを列に入力するには、次の 3 つの方法があります。

- [数値表示]に移動し、データを直接入力します。例については、238 ページの 2 変数統計アプリケーションでの作業の開始を参照してください。
- ホーム表示に移動し、リストからデータをコピーします。たとえば、ホーム表示で「`L1`」と入力し、Sto ▸ をタップしてから「`C1`」と入力すると、リスト L1 内の項目が 2 変数統計アプリケーションの列 C1 にコピーされます。
- ホーム表示に移動し、[スプレッドシート]アプリケーションからデータをコピーします。たとえば、目的のデータがスプレッドシート アプリケーションの A1:A10 にあり、そのデータを列 C7 にコピーするとします。2 変数統計アプリケーションが開いている状態で、ホーム表示に戻って「`Spreadsheet.A1:A10`」と入力し、Sto ▸ をタップして「`C7`」と入力してから、Enter ≈ を押します。

注記：有効な 2 変数統計を得るには、1 つのデータ列に少なくとも 4 つの標本が入力されている必要があります。

どの方法でも、入力したデータは自動的に保存されます。いったんこのアプリケーションを離れてから戻っても、最後に入力したデータは引き続き利用できます。

データの入力後は、[記号表示]でデータ セット（およびそれらのデータ セットをプロットする方法）を定義する必要があります。

[数値表示]：メニュー項目

[数値表示]でタップできるメニュー項目は次のとおりです。

編集	選択した項目を編集可能にするために入力行にコピーします
その他	編集オプションのメニューを表示します。246 ページの[その他]メニューを参照してください
移動	指定したリスト項目にカーソルを移動します
並替	さまざまな方法でデータを並べ替えます
作成	指定した列の値のリストを生成する式を入力するための入力フォームを表示します
統計	[記号表示]で選択されている各データ セットの統計を計算します。

[その他]メニュー

[その他]メニューにはデータのリストを編集するためのオプションが含まれています。オプションの説明を次の表に示します。

オプション	サブオプション	用途
[挿入]	[行]	選択されたリストに新しい行を挿入します。新しい行には要素として 0 が含まれます
[削除]	[列]	選択されたリストの内容を削除します 特定の項目を削除するには、その項目を選択して を押します
[選択]	[行]	現在選択されているセルを含む行を選択します。その後、行全体をコピーできます
	[ボックス]	開始位置および終了位置を指定して矩形選択するためのダイアログ ボックスが開きます。開始位置となるセルをタップしたまま選択し、指でドラッグして要素を矩形選択することもできます。選択後、その要素をコピーできます
	[列]	現在のリストを選択します。選択後、そのリストをコピーできます
[選択]		選択モードのオン/オフを切り替えます

オプション	サブオプション	用途
		選択モードがオフの場合、セルをタップしたまま指でドラッグして、複数のセルを矩形選択することもできます
[スワップ]	[列]	2 つの列（リスト）の内容を入れ替えます

回帰モデルの定義

回帰モデルは[記号表示]で定義します。次の 3 つの方法があります。

- 初期設定のオプションを受け入れて、データを直線にフィットする。
- あらかじめ定義されたフィット タイプ（対数や指数など）を選択する。
- 独自の数式を入力する。式がプロットされると、標本にどの程度フィットするかがわかります。

フィットの選択

1. Symb Setup を押して[記号表示]を表示します。
2. 目的の分析（S1 ~ S5）について、**[タイプ]**フィールドを選択します。
3. フィールドをもう一度タップして、フィット タイプのメニューを表示します。
4. メニューから、目的のフィット タイプを選択します（247 ページのフィット タイプを参照してください）。

フィット タイプ

次の 12 種類のフィット タイプが利用できます。

フィット タイプ	意味
Linear（1 次）	（初期値）データを次の直線にフィットします：y = mx + b。最小 2 乗フィットを使用します
Logarithmic（対数）	データを次の対数曲線にフィットします：y = m lnx + b
Exponential（指数）	データを e を底とする指数曲線にフィットします：$y = b * e^{mx}$
Power（べき乗）	データを次のべき乗曲線にフィットします：$y = b * x^{m}$
Exponent（指数）	データを次の指数曲線にフィットします：$y = b * m^{x}$
Inverse（逆）	データを次の逆比例にフィットします：y = m lnx + b
Logistic（ロジスティック）	データを次のロジスティック曲線にフィットします：$y = \frac{L}{1 + ae^{(-bx)}}$（ここで、L は増加の飽和値です）。L には正の実数を保存できます。また、L = 0 の場合は、L を自動的に計算できます
Quadratic（2 次）	データを次の 2 次曲線にフィットします：$y = ax^2 + bx + c$。3 つ以上の点が必要です
Cubic（3 次）	データを次の 3 次多項式にフィットします：$y = ax^3 + bx^2 + cx + d$
Quartic（4 次）	次の 4 次多項式にフィットします：$y = ax^4 + bx^3 + cx^2 + dx + e$

フィットタイプ	意味
Trigonometric（三角）	データを次の三角曲線にフィットします : y = a * sin(bx + c) + d。3 つ以上の点が必要です
User Defined（ユーザー定義）	独自のフィットを定義します（以下を参照してください）

独自のフィットの定義

1. Symb Setup を押して[記号表示]を表示します。
2. 目的の分析（S1 ~ S5）について、**[タイプ]**フィールドを選択します。
3. フィールドをもう一度タップして、フィットタイプのメニューを表示します。
4. メニューから、**[ユーザー定義]**を選択します。
5. 対応するフィットのフィールドを選択します。
6. 式を入力して Enter ≈ を押します。独立変数を X にし、式に未知の変数を含める必要があります。たとえば、1.5 * cos(x) + 0.3 * sin(x)です。このアプリケーションでは、変数を大文字で入力する必要があることに注意してください。

計算される統計

統計 をタップすると、3 セットの統計が利用できるようになります。初期設定では、独立列と従属列の両方に関する統計が表示されます。独立列のみに関する統計を表示するには X を、従属列から得られる統計を表示するには Y をタップします。 統計 をタップすると初期設定の表示に戻ります。次の表で、各表示で表示される統計について説明します。

統計 をタップした場合に計算される統計は次のとおりです。

統計	定義
n	標本の数
r	（選択したフィットタイプにかかわらず）1 次フィットのみに基づく、独立データ列および従属データ列の相関係数。 –1 から 1 の間の値を返します。ここで、1 および –1 は最良適合を示します
R^2	決定係数、つまり相関係数の 2 乗。この統計の値は、選択したフィットタイプに依存します。値が 1 の場合、完全適合を示します
sCOV	独立データ列および従属データ列の標本共分散
σCOV	独立データ列および従属データ列の母共分散
ΣXY	x および y の個々の積すべての和

X をタップした場合に表示される統計は次のとおりです。

統計	定義
$\dot{x}$	x（独立）値の平均値
ΣX	x 値の和
ΣX^2	x^2 値の和
sX	独立列の標本標準偏差
σX	独立列の母標準偏差
serrX	独立列の標準誤差
ssX	X の偏差平方和

Y をタップした場合に表示される統計は次のとおりです。

統計	定義
$\dot{y}$	y（従属）値の平均値
ΣY	y 値の和
ΣY^2	y^2 値の和
sY	従属列の標本標準偏差
σY	従属列の母標準偏差
serrY	従属列の標準誤差
ssY	Y の偏差平方和

統計データのプロット

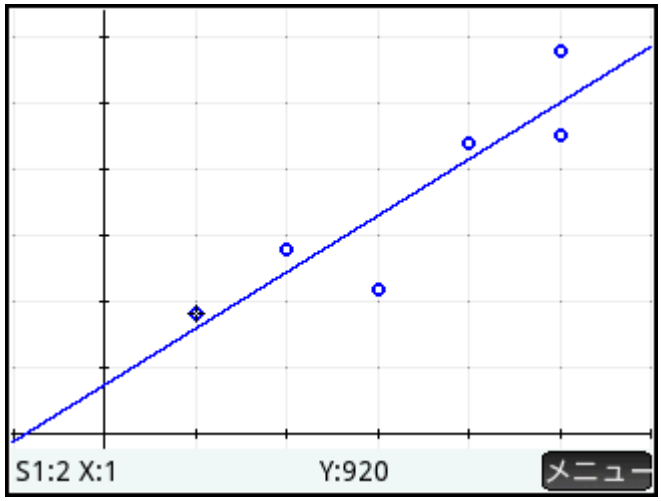
データを入力し、分析するデータ セットを選択して、フィット モデルを指定したら、データをプロットできます。一度に 5 つまでの散布図をプロットできます。

1. [記号表示]で、プロットするデータ セットを選択します。
2. すべてのデータがプロットされることを確認します。そのためには、**[X 範囲]**フィールドおよび**[Y 範囲]**フィールドを [プロットのセットアップ]表示（ Shift Plot Setup ）で確認（および必要に応じて調整）します。
3. Plot Setup を押します。

   データ セットおよび回帰直線が最適な方法で配置されていない場合は、 View Copy を押して**[自動スケール]**を選択します。自動スケールは、スケールの出発点として利用できます。後でこのスケールを[プロットのセットアップ]表示で調整できます。

散布図のトレース

プロットの下にある数値は、カーソルが S1 の 2 番目の標本(1, 920)にあることを示しています。次の標本に移動してその標本に関する情報を表示するには、 ▶ を押します。



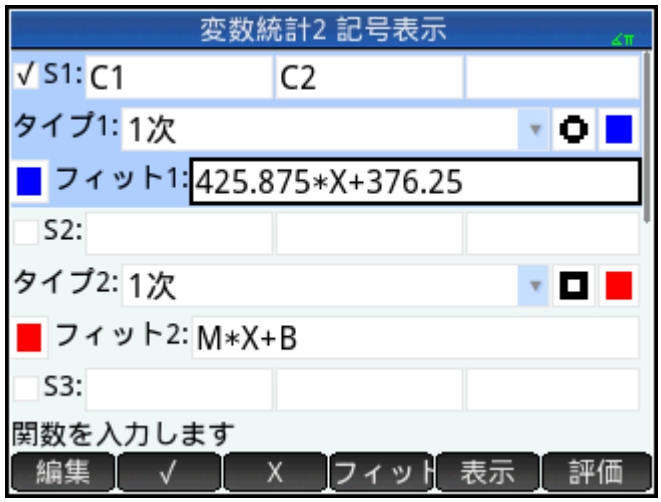
曲線のトレース

回帰直線が表示されない場合は、フィットをタップします。トレースカーソルの座標は画面の下部に表示されます（表示されていない場合は、Menu Paste をタップしてください）。

[記号表示]で回帰直線の方程式を表示するには、Symb Setup を押します。

方程式が長すぎて画面に収まらない場合は、その方程式を選択して表示をタップします。

以下の例は、回帰直線の傾き（m）が 425.875 であり、y 切片（b）が 376.25 であることを示しています。



トレースの順序

および を使用するとカーソルが散布図のフィットに沿って移動するか、または散布図の点の間を移動します。トレースする散布図またはフィットを選択するには、 および を使用します。アクティブな分析（S1 から S5）のそれぞれについて、トレースの順序は最初が散布図で次がフィットになります。そのため、S1 と S2 の両方がアクティブである場合、 Plot Setup を押すと、トレーサーは初期設定で S1 の散布図に配置されます。S1 のフィットをトレースするには を押します。この時点で、S1 の散布図に戻るには を、S2 の散布図をトレースするにはもう一度 を押します。S2 のフィットをトレースするには、 を 3 回押します。 を 4 回押すと、S1 の散布図に戻ります。現在のトレース対象がわからなくなった場合は、 定義 をタップして、現在トレース中のオブジェクト（散布図またはフィット）の定義を表示します。

[プロット表示]：メニュー項目

[プロット表示]のメニュー項目は次のとおりです。

ボタン	用途
ズーム	[ズーム]メニューを表示します
トレース	トレース モードをオンまたはオフにします
移動	ジャンプ先の最も適合する曲線上の x 値（または、カーソルが最も適合する曲線上ではなく標本上にある場合はジャンプ先の標本）を指定できます。
フィット	[記号表示]で有効になっている分析に最も適合する曲線の表示または非表示を切り替えます。
関数	[関数]メニューを開きます。251 ページの[関数]メニューを参照してください。
メニュー	メニュー ボタンの表示または非表示を切り替えます

[関数]メニュー

[関数]メニューのメニュー項目は以下のとおりです。

オプション	用途
フィット	[記号表示]で有効になっている分析に最も適合する曲線の表示または非表示を切り替えます。[プロット表示]メニューのボタンをタップすることもできます。
スケッチ	散布図の関数フィット曲線を指でスケッチできます。
定義	現在の散布図またはフィット曲線の定義を表示します。散布図とフィット曲線を切り替え、[記号表示]で有効になっているプロットをスクロールするには、 または を押します。

スケッチ

スケッチ オプションは指で関数をスケッチする[プロット表示]を開きます。画面の下部にはメッセージが表示されます。前のスケッチが不要な場合には、新しい関数をスケッチできます。関数のスケッチが完了したら、 OK をタップします。[記号表示]で使用できる最初のデータセットのフィット種類（S1 ~ S5）が**[ユーザー定義]**に変更され、フィットの（X の）式はユーザー定義のフィット定義として保存されます。

[プロットのセットアップ]表示

プロット機能を備えたすべてのアプリケーションと同様に、[プロットのセットアップ]表示（ Shift Plot Setup ）では、[プロット表示]の範囲および外観を設定できます。設定は、その他の[プロットのセットアップ]表示での操作と共通です。[プロットのセットアップ]表示の 2 ページ目には、**[接続]**フィールドがあります。このオプションを選択すると、[プロット表示]で標本が直線で接続されます。

値の予測

PredX は、Y の値が与えられた場合に X の値を予測する関数です。同様に、PredY は、X の値が与えられた場合に Y の値を予測する関数です。どちらの場合も、選択されたフィット タイプに応じて、データに最も適合する方程式に基づいた予測が行われます。

値の予測は、2 変数統計アプリケーションの[プロット表示]で行うことができます。また、ホーム表示で行うことも可能です。

[プロット表示]

1. [プロット表示]で、 フィット をタップしてデータ セットの回帰曲線を表示します（まだ表示されていない場合）。
2. トレース カーソルが回帰曲線上に配置されていることを確認します（配置されていない場合、 ▲ または ▼ を押します）。
3. ▶ または ◀ を押すと、カーソルが回帰曲線に沿って移動し、対応する X 値および Y 値が画面の下部に表示されます（これらの値が表示されていない場合は、 Menu Paste をタップしてください）。

カーソルを特定の X 値に強制的に移動するには、 移動 をタップし、値を入力して OK をタップします。カーソルが、曲線上の指定された点にジャンプします。

ホーム表示

2 変数統計アプリケーションがアクティブなアプリケーションである場合は、ホーム表示でも X 値および Y 値を予測できます。

- 指定された Y 値に対応する X 値を予測するには、`PredX(Y)` と入力して Enter ≈ を押します。
- 指定された X 値に対応する Y 値を予測するには、`PredY(X)` と入力して Enter ≈ を押します。

注記 : 複数のフィット曲線が表示されている場合は、PredX 関数および PredY 関数は[記号表示]で定義されている最初のアクティブなフィットを使用します。

PredX および PredY は、入力行に直接入力することも、([2 変数統計]カテゴリにある) [アプリ]関数メニューから選択することもできます。[アプリ]関数メニューはツールボックス メニュー（ Mem ）の 1 つです。

プロットのトラブルシューティング

プロットで問題が生じる場合は、以下を確認してください。

- 目的のフィット（回帰モデル）が選択されていること。
- [記号表示]で、分析またはプロットするデータ セットのみが選択されていること。
- プロットの範囲が適切であること。 View Copy を押して**[自動スケール]**を選択するか、[プロットのセットアップ]表示でプロット パラメーターを調整します。
- 列のペアの両方にデータが含まれており、その長さが同じであること。

13 推論アプリケーション

推論アプリケーションは、線形回帰の推論に基づく検定と信頼区間の両方に加えて、仮説検証、信頼区間、およびカイ 2 乗検定を計算します。推論アプリケーションだけでなく、[数学]メニューにも、さまざまな分布（カイ 2 乗、F、二項、ポワソンなど）に基づく確率関数のセットが豊富に用意されています。

1 つまたは 2 つの標本から得られた統計に基づいて、次の数量について仮説を検証したり、信頼区間を求めたりすることができます。

- 平均
- 比率
- 2 つの平均の差
- 2 つの比率の差

カイ 2 乗分布に基づいて、適合度検定および 2 元表に基づく検定を実行することができます。また、線形回帰の推論に基づいて次の計算を実行できます。

- 線形 t 検定
- 傾きについての信頼区間
- 切片についての信頼区間
- 平均応答についての信頼区間
- 将来の応答についての予測区間

さらに、データのリストに対して一元配置分散分析（ANOVA)を実行することもできます。

サンプル データ

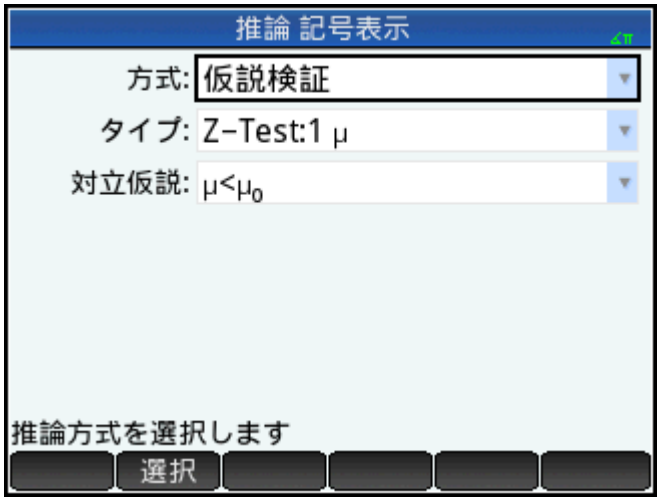
多くの計算について、推論アプリケーションの[数値表示]にはサンプル データ（アプリケーションのリセットによって復元可能）が付属しています。このサンプル データは、アプリケーションを理解する上で役立ちます。

推論アプリケーションでの作業の開始

以降のセクションで、サンプル データを使用した 1 つの平均に関する Z 検定を行います。

推論アプリケーションを開く

▲ Apps Info を押して、**[推論]**を選択します。



推論アプリケーションが[記号表示]で開きます。

[記号表示]のオプション

以降の表で、[記号表示]で利用できるオプションについて説明します。

表 13-1 仮説検証

検定	説明
Z-Test: 1 μ	1 つの平均に関する Z 検定
Z-Test: $\mu_1 - \mu_2$	2 つの平均の差に関する Z 検定
Z-Test: 1 π	1 つの比率に関する Z 検定
Z-Test: $\pi_1 - \pi_2$	2 つの比率の差に関する Z 検定
T-Test: 1 μ	1 つの平均に関する T 検定
T-Test: $\mu_1 - \mu_2$	2 つの平均の差に関する T 検定

表 13-2 信頼区間

検定	説明
Z-Int: 1 μ	正規分布に基づく、1 つの平均についての信頼区間
Z-Int: $\mu_1 - \mu_2$	正規分布に基づく、2 つの平均の差についての信頼区間
Z-Int: 1 π	正規分布に基づく、1 つの比率についての信頼区間
Z-Int: $\pi_1 - \pi_2$	正規分布に基づく、2 つの比率の差についての信頼区間
T-Int: 1 μ	スチューデントの t 分布に基づく、1 つの平均についての信頼区間
T-Int: $\mu_1 - \mu_2$	スチューデントの t 分布に基づく、2 つの平均の差についての信頼区間

表 13-3 X^2 検定

検定	説明
Goodness of fit（適合度検定）	カテゴリカル データに基づくカイ 2 乗適合度検定
2 要因検定	2 元表のカテゴリカル データに基づくカイ 2 乗検定

表 13-4 回帰

検定	説明
線形 t 検定	線形回帰の t 検定
区間: 傾き	t 分布に基づく、真の線形回帰直線の傾きについての信頼区間
区間: 切片	t 分布に基づく、真の線形回帰直線の y 切片についての信頼区間
区間: 平均応答	t 分布に基づく、平均応答についての信頼区間
Prediction interval（予測区間）	t 分布に基づく、将来の応答についての予測区間

表 13-5 ANOVA

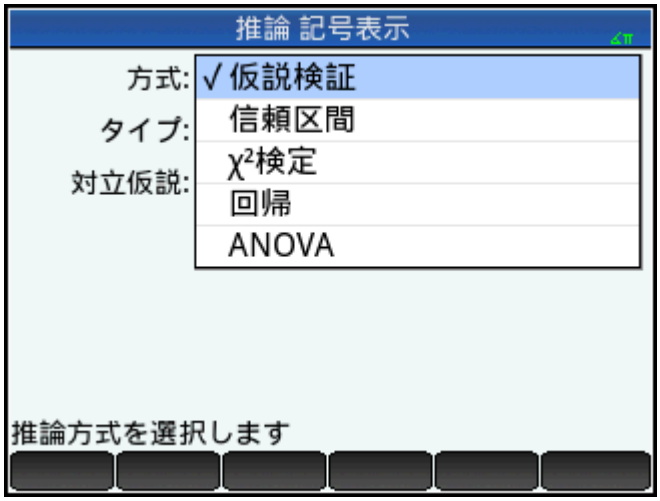
検定	説明
1-way ANOVA（一元配置分散分析）	F 分布に基づく一元配置分散分析

仮説検証のどれかを選択した場合、帰無仮説に対立する仮説として検証する対立仮説を選択できます。各検定について、2 つの数量の定量比較に基づく対立仮説を 3 種類から選択できます。帰無仮説とは、2 つの数量が常に等しいとする仮説です。そのため、対立仮説は、2 つの数量が等しくならない複数のケース（<、>、および≠）を対象とします。

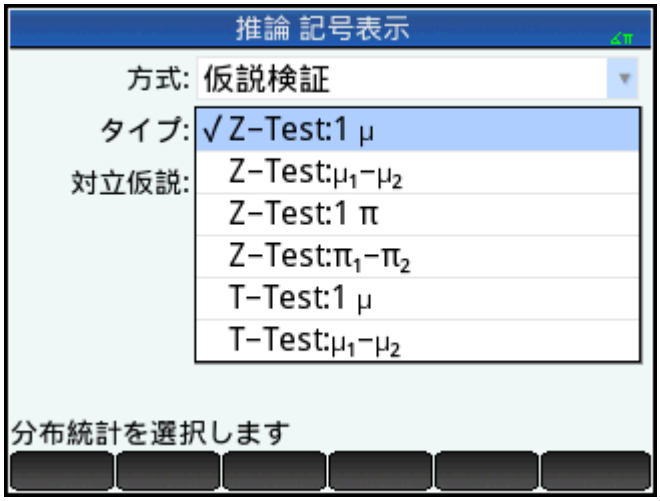
このセクションでは、アプリケーションの機能を説明するために、サンプル データで 1 つの平均に関する Z 検定を行います。

推論方式の選択

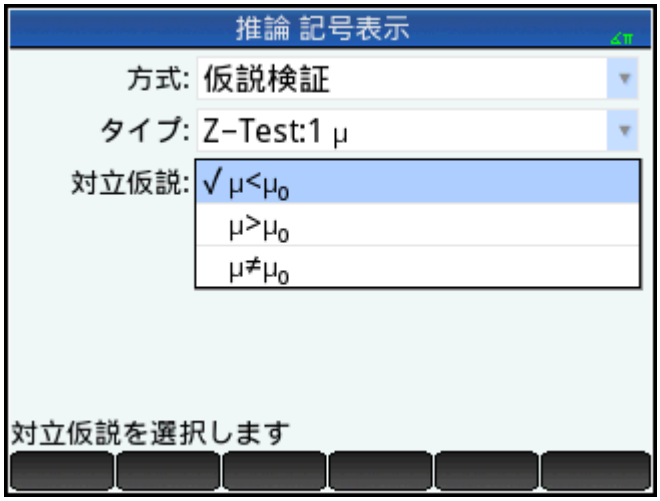
1. 初期設定の推論方式は、**[仮説検証]**です。これが選択されていない場合は、**[方式]**をタップして選択します。



2. 検定のタイプを選択します。この場合は、**[Z-Test:: 1 μ]**を**[タイプ]**メニューから選択します。



3. 対立仮説を選択します。この場合は、**[μ <]** μ_0 を**[対立仮説]**メニューから選択します。



データの入力

▲ [数値表示]に移動し、サンプル データを確認します。

次の表で、サンプル データに関するこの表示内のフィールドについて説明します。

フィールド名	説明
$\dot{x}$	標本平均
n	t 分布に基づく、真の線形回帰直線の傾きについての信頼区間
μ_0	母集団平均の想定値
σ	母標準偏差
α	検定の α 水準

[数値表示]では、検査する状況に適した標本統計および母集団パラメーターを入力します。ここで入力されているサンプル データは、ある学生が自分のグラフ電卓で 50 個の疑似乱数を生成したケースの一部です。アルゴリズムが適切に機能していれば、平均は 0.5 近くになり、母標準偏差は約 0.2887 になります。しかし、標本平均（0.461368）がやや低いように思われるため、帰無仮説に対して不適切な対立仮説を検定していることをこの学生は懸念しています。

検定結果の表示

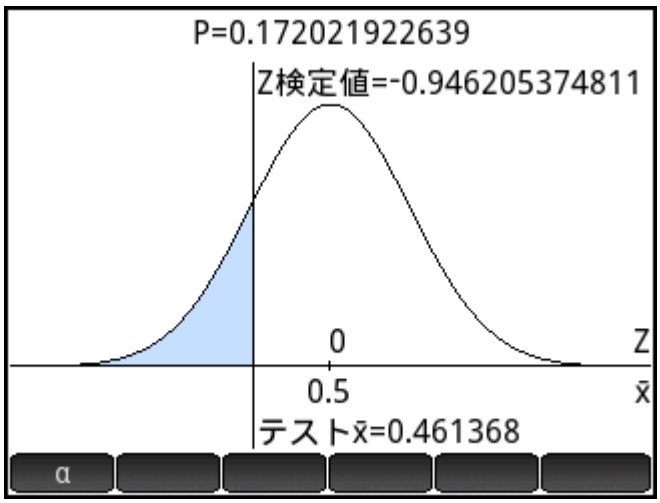
▲ 計算 をタップします。

検定の分布値およびそれに関連付けられた確率が表示されます。また、検定の臨界値および統計の関連付けられた臨界値も表示されます。この場合は、帰無仮説を棄却すべきでないことがこの検定によって示されています。

OK をタップして[数値表示]に戻ります。

検定結果のプロット

▲ Plot Setup を押します。



分布のグラフが表示され、Z 検定値が示されます。対応する X 値も示されます。

Z 臨界値を表示するには、 α をタップします。α 水準が表示されている状態で、 ▲ または ▼ を押して α 水準を増減できます。

統計のインポート

多くの計算について、推論アプリケーションでは、1 変数統計アプリケーションおよび 2 変数統計アプリケーションのデータから要約統計をインポートできます。その他の計算については、手動でデータをインポートできます。次の例で、その手順を示します。

一連の 6 回の実験によって、ある液体の沸点として次の値が得られたとします。

82.5、83.1、82.6、83.7、82.4、および 83.0

この標本に基づいて、真の沸点を 90%の信頼水準で推定します。

1 変数統計アプリケーションを開く

▲ Apps Info を押して、**[1 変数統計]**を選択します。

不要なデータのクリア

▲ アプリケーション内に不要なデータがある場合は、以下の操作を行ってそれをクリアします。

Shift Esc Clear を押してから、**[すべての列]**を選択します。

データの入力

▲ 列 D1 に、実験中に得られた沸点を入力します。

82 5 Enter

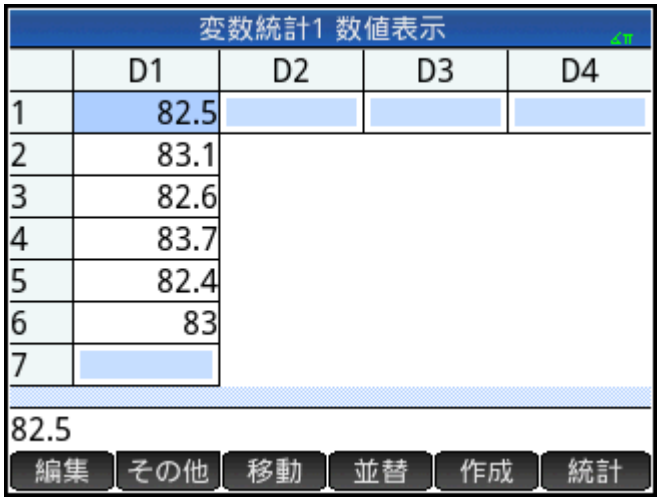
83 1 Enter

82 6 Enter

83 7 Enter

82 [•] 4 [Enter]

83 [Enter]



統計の計算

1. 統計 をタップします。

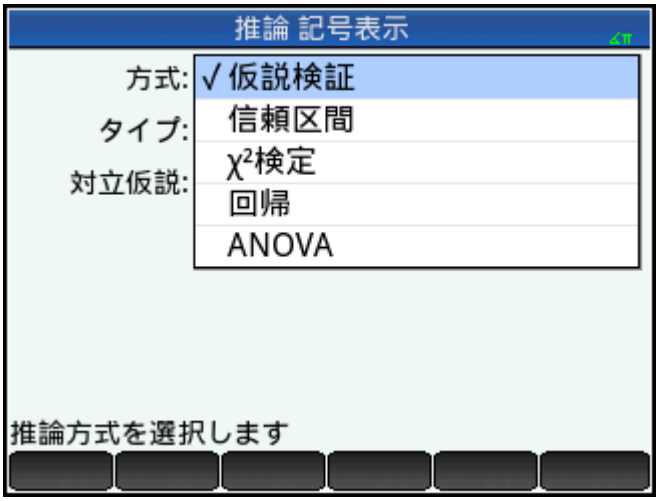
   これで、計算された統計が推論アプリケーションにインポートされるようになります。

2. OK をタップし、統計ウィンドウを閉じます。

推論アプリケーションを開く

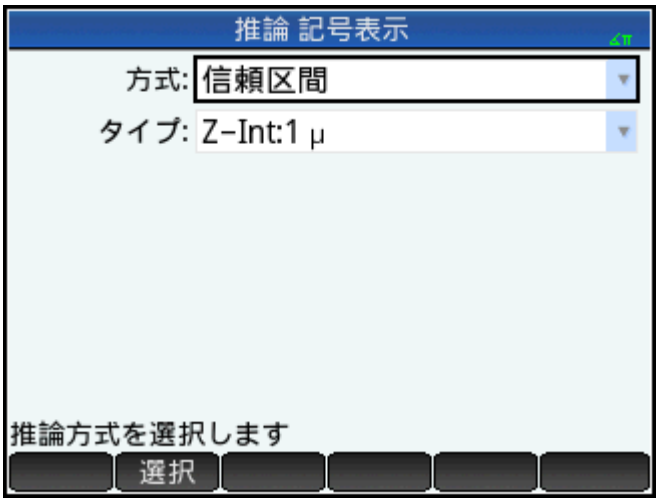
▲ 推論アプリケーションを開き、現在の設定をクリアします。

Apps Info を押して、**[推論]**を選択し、 Shift Esc Clear を押します。

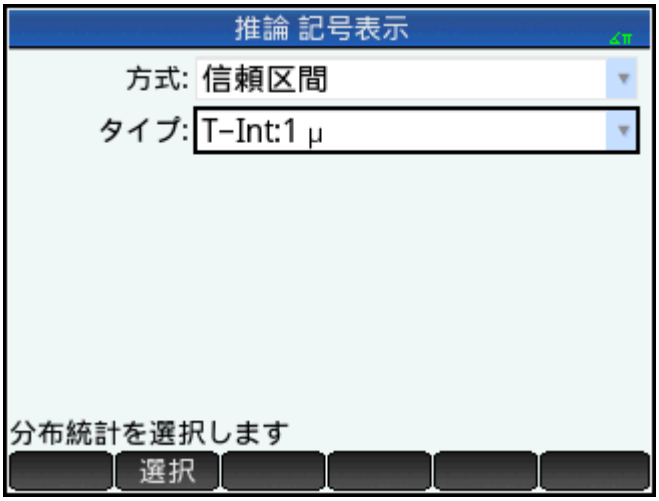


推論の方式およびタイプの選択

1. **[方式]**を選択してから**[信頼区間]**を選択します。



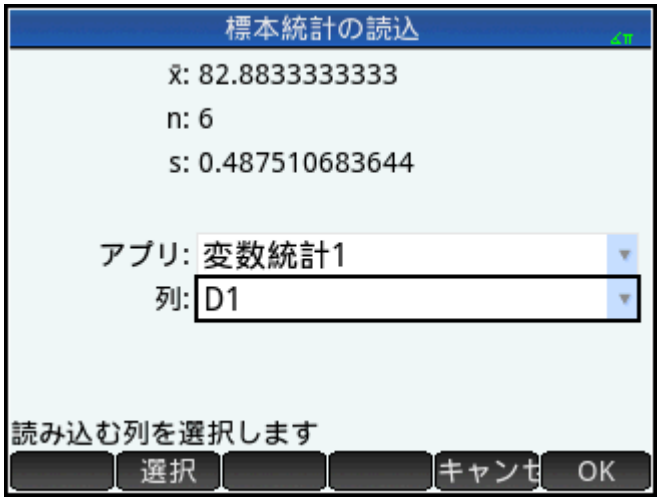
2. **[タイプ]**を選択してから**[T-Int: 1 μ]**を選択します。



データのインポート

1. Num Setup を押します。
2. 以下の操作を行って、インポートするデータを指定します。

   読込 をタップします。
3. **[アプリ]**フィールドで、インポートするデータがある統計アプリケーションを選択します。
4. **[列]**フィールドで、そのアプリケーション内でデータが格納されている列を指定します（D1 が初期値です）。



5. OK をタップします。
6. **[C]**フィールドで、90%信頼区間を指定します。

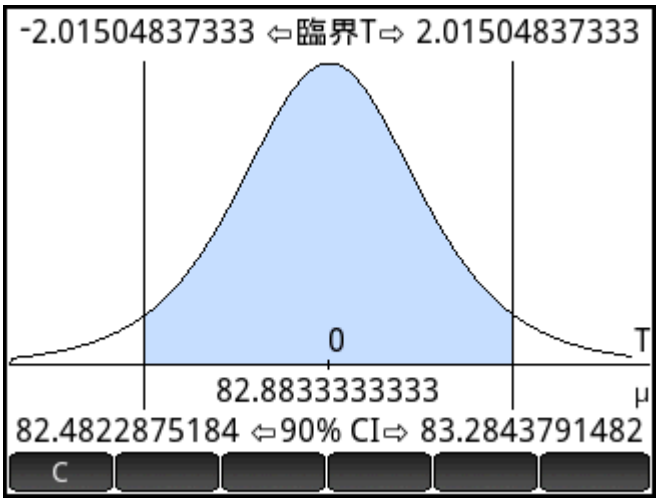
数値での結果の表示

1. [数値表示]で信頼区間を表示するには、 計算 をタップします。

2. OK をタップして[数値表示]に戻ります。

グラフでの結果の表示

▲ [プロット表示]で信頼区間を表示するには、 Plot Setup を押します。



90%信頼区間は[82.48…, 83.28…]です。

仮説検証

仮説検証は、1 つまたは 2 つの母集団の統計パラメーターに関する仮説の妥当性を検定するために使用します。この検定は、母集団の標本の統計に基づいて行われます。

HP Prime の仮説検証では、確率の計算に正規 Z 分布またはスチューデントの t 分布を使用します。他の分布を使用する場合は、ホーム表示で、[数学]メニューの[確率]カテゴリ内にある分布を使用してください。

1 標本 Z 検定

メニュー名

Z-Test: 1 μ

この検定では、1 つの標本による統計に基づいて、帰無仮説に対立するものとして選択された仮説の証拠の強度を測定します。帰無仮説は、母集団平均が指定された値に等しい、つまり $H_0: \mu = \mu_0$ というものです。

帰無仮説に対立するものとして検証する対立仮説を次の中から選択します。

- $H_0: \mu < \mu_0$
- $H_0: \mu > \mu_0$
- $H_0: \mu \neq \mu_0$

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}$	標本平均
n	標本サイズ
μ_0	仮説の母集団平均
σ	母標準偏差
α	有意水準

結果

結果は、次のとおりです。

結果	説明
テスト Z	Z 検定の統計
テスト $\dot{x}$	Z 検定値に関連付けられた $\dot{x}$ の値
P	Z 検定の統計に関連付けられた確率
臨界 Z	指定した α 水準に関連付けられた Z の境界値
臨界 $\dot{x}$	指定した α 水準によって必要になる $\dot{x}$ の境界値

2 標本 Z 検定

メニュー名

Z-Test: $\mu_1 - \mu_2$

この検定では、それぞれ母集団が異なる 2 つの標本に基づいて、帰無仮説に対立するものとして選択された仮説の証拠の強度を測定します。帰無仮説は、2 つの母集団の平均が等しい、つまり $H_0: \mu_1 = \mu_2$ というものです。

帰無仮説に対立するものとして検証する対立仮説を次の中から選択できます。

- $H_0: \mu_1 < \mu_2$
- $H_0: \mu_1 > \mu_2$
- $H_0: \mu_1 \neq \mu_2$

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}_1$	標本 1 の平均
$\dot{x}_2$	標本 2 の平均
n_1	標本 1 のサイズ
n_2	標本 2 のサイズ
σ_1	母標準偏差 1
σ_2	母標準偏差 2
α	有意水準

結果

結果は、次のとおりです。

結果	説明
テスト Z	Z 検定の統計
テスト $\Delta\dot{x}$	Z 検定値に関連付けられた平均の差
P	Z 検定の統計に関連付けられた確率
臨界 Z	指定した α 水準に関連付けられた Z の境界値
臨界 $\Delta\dot{x}$	指定した α 水準に関連付けられた平均の差

1 比率の Z 検定

メニュー名

Z-Test: 1 π

この検定では、1 つの標本による統計に基づいて、帰無仮説に対立するものとして選択された仮説の証拠の強度を測定します。帰無仮説は、成功比率が想定値である、つまり $H_0: \pi = \pi_0$ というものです。

帰無仮説に対立するものとして検証する対立仮説を次の中から選択します。

- $H_0: \pi < \pi_0$
- $H_0: \pi > \pi_0$
- $H_0: \pi \neq \pi_0$

入力

入力は、次のとおりです。

フィールド名	説明
x	標本の成功数
n	標本サイズ
π_0	母集団の成功比率
α	有意水準

結果

結果は、次のとおりです。

結果	説明
テスト z	z 検定の統計
テスト $\hat{p}$	標本の成功比率
P	z 検定の統計に関連付けられた確率
臨界 z	指定した α 水準に関連付けられた z の境界値
臨界 $\hat{p}$	指定した水準に関連付けられた成功比率

2 比率の z 検定

メニュー名

Z-Test: $\pi_1 - \pi_2$

この検定では、それぞれ母集団が異なる 2 つの標本による統計に基づいて、帰無仮説に対立するものとして選択された仮説の証拠の強度を測定します。帰無仮説は、2 つの母集団の成功比率が等しい、つまり $H_0: \pi_1 = \pi_2$ というものです。

帰無仮説に対立するものとして検証する対立仮説を次の中から選択します。

- $H_0: \pi_1 < \pi_2$
- $H_0: \pi_1 > \pi_2$
- $H_0: \pi_1 \neq \pi_2$

入力

入力は、次のとおりです。

フィールド名	説明
x_1	標本 1 の成功数
x_2	標本 2 の成功数
n_1	標本 1 のサイズ
n_2	標本 2 のサイズ
α	有意水準

結果

結果は、次のとおりです。

結果	説明
テスト Z	Z 検定の統計
テスト $\hat{p}$	Z 検定値に関連付けられた 2 つの標本の成功比率の差
P	Z 検定の統計に関連付けられた確率
臨界 Z	指定した α 水準に関連付けられた Z の境界値
臨界 Δ $\hat{p}$	指定した α 水準に関連付けられた 2 つの標本の成功比率の差

1 標本 T 検定

メニュー名

T-Test: 1 μ

この検定は、母標準偏差が不明な場合に使用されます。この検定では、1 つの標本による統計に基づいて、帰無仮説に対立するものとして選択された仮説の証拠の強度を測定します。帰無仮説は、標本平均に何らかの想定値が含まれる、つまり $H_0: \mu = \mu_0$ というものです。

帰無仮説に対立するものとして検証する対立仮説を次の中から選択します。

- $H_0: \mu < \mu_0$
- $H_0: \mu > \mu_0$
- $H_0: \mu \neq \mu_0$

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}$	標本平均
s	標本標準偏差
n	標本サイズ
μ_0	仮説の母集団平均
α	有意水準

結果

結果は、次のとおりです。

結果	説明
T 検定	T 検定の統計
T テスト $\dot{x}$	t 検定値に関連付けられた $\dot{x}$ の値
P	T 検定の統計に関連付けられた確率
DF	自由度
臨界 T	指定した α 水準に関連付けられた T の境界値
臨界 $\dot{x}$	指定した α 水準によって必要になる $\dot{x}$ の境界値

2 標本 T 検定

メニュー名

T-Test: $\mu_1 - \mu_2$

この検定は、母標準偏差が不明な場合に使用されます。この検定では、それぞれ母集団が異なる 2 つの標本による統計に基づいて、帰無仮説に対立するものとして選択された仮説の証拠の強度を測定します。帰無仮説は、2 つの母集団の平均が等しい、つまり $H_0: \mu_1 = \mu_2$ というものです。

帰無仮説に対立するものとして検証する対立仮説を次の中から選択します。

- $H_0: \mu_1 < \mu_2$
- $H_0: \mu_1 > \mu_2$
- $H_0: \mu_1 \neq \mu_2$

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}_1$	標本 1 の平均
$\dot{x}_2$	標本 2 の平均

フィールド名	説明
s_1	標本 1 標準偏差
s_2	標本 2 標準偏差
n_1	標本 1 のサイズ
n_2	標本 2 のサイズ
α	有意水準
プール済み	その標準偏差に基づいて標本をプールするには、このオプションを選択します

結果

結果は、次のとおりです。

結果	説明
T 検定	T 検定の統計
テスト $\Delta\dot{x}$	t 検定値に関連付けられた平均の差
P	T 検定の統計に関連付けられた確率
DF	自由度
臨界 T	指定した α 水準に関連付けられた T の境界値
臨界 $\Delta\dot{x}$	指定した α 水準に関連付けられた平均の差

信頼区間

HP Prime で実行できる信頼区間の計算は、正規 Z 分布またはスチューデントの t 分布に基づいて行われます。

1 標本 Z 区間

メニュー名

Z-Int: 1 μ

このオプションでは、真の母標準偏差 σ が既知の場合に、正規 Z 分布を使用して母集団の真の平均 μ についての信頼区間を計算します。

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}$	標本平均
n	標本サイズ

フィールド名	説明
σ	母標準偏差
C	信頼水準

結果

結果は、次のとおりです。

結果	説明
C	信頼水準
臨界 Z	Z の臨界値
下側	μ の下側限界
上側	μ の上側限界

2 標本 Z 区間

メニュー名

Z-Int: $\mu_1 - \mu_2$

このオプションでは、母標準偏差 σ_1 および σ_2 が既知の場合に、正規 Z 分布を使用して 2 つの母集団の平均の差 $\mu_1 - \mu_2$ についての信頼区間を計算します。

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}_1$	標本 1 の平均
$\dot{x}_2$	標本 2 の平均
n_1	標本 1 のサイズ
n_2	標本 2 のサイズ
σ_1	母標準偏差 1
σ_2	母標準偏差 2
C	有意水準

結果

結果は、次のとおりです。

結果	説明
C	信頼水準
臨界 Z	Z の臨界値
下側	Δμ の下側限界
上側	Δμ の上側限界

1 比率の Z 区間

メニュー名

Z-Int: 1 π

このオプションでは、正規 Z 分布を使用して、サイズ n の標本にいくつかの成功数 x が含まれる場合の、母集団の成功比率についての信頼区間を計算します。

入力

入力は、次のとおりです。

フィールド名	説明
x	標本の成功数
n	標本サイズ
C	信頼水準

結果

結果は、次のとおりです。

結果	説明
C	信頼水準
臨界 Z	Z の臨界値
下側	π の下側限界
上側	π の上側限界

2 比率の Z 区間

メニュー名

Z-Int: $\pi_1 - \pi_2$

このオプションでは、正規 Z 分布を使用して、2 つの母集団の成功比率の差についての信頼区間を計算します。

入力

入力は、次のとおりです。

フィールド名	説明
x_1	標本 1 の成功数
x_2	標本 2 の成功数
n_1	標本 1 のサイズ
n_2	標本 2 のサイズ
C	信頼水準

結果

結果は、次のとおりです。

結果	説明
C	信頼水準
臨界 Z	Z の臨界値
下側	Δπ の下側限界
上側	Δπ の上側限界

1 標本 T 区間

メニュー名

T-Int: 1 μ

このオプションでは、スチューデントの t 分布を使用して、真の母標準偏差 σ が不明な場合の、母集団の真の平均 μ についての信頼区間を計算します。

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}$	標本平均
s	標本標準偏差
n	標本サイズ
C	信頼水準

結果

結果は、次のとおりです。

結果	説明
C	信頼水準
DF	自由度
臨界 T	T の臨界値
下側	μ の下側限界
上側	μ の上側限界

2 標本 T 区間

メニュー名

T-Int: $\mu_1 - \mu_2$

このオプションでは、母標準偏差 σ_1 および σ_2 が不明な場合に、スチューデントの t 分布を使用して 2 つの母集団の平均の差 $\mu_1 - \mu_2$ についての信頼区間を計算します。

入力

入力は、次のとおりです。

フィールド名	説明
$\dot{x}_1$	標本 1 の平均
$\dot{x}_2$	標本 2 の平均
s_1	標本 1 標準偏差
s_2	標本 2 標準偏差
n_1	標本 1 のサイズ
n_2	標本 2 のサイズ
C	信頼水準
プール済み	その標準偏差に基づいて標本をプールするかどうかの選択

結果

結果は、次のとおりです。

結果	説明
C	信頼水準
DF	自由度
臨界 T	T の臨界値
下側	Δμ の下側限界
上側	Δμ の上側限界

カイ 2 乗検定

HP Prime 電卓は、カイ 2 乗分布に基づいてカテゴリカル データに関する検定を実行できます。具体的には、HP Prime 電卓では、適合度検定と 2 元表に基づく検定の両方がサポートされています。

適合度検定

メニュー名

[適合度検定]

このオプションでは、カイ 2 乗分布を使用して、期待される確率または期待されるカウントに対する、求められたカウントのカテゴリカル データの適合度を検定します。[記号表示]の**[期待]**ボックスで次のどちらかを選択します：**[確率]**（初期設定）または**[カウント]**

入力

[期待]→[確率]が選択されている場合、[数値表示]の入力は次のとおりです。

フィールド名	説明
ObsList	求められたカウント データのリスト
ProbList	期待される確率のリスト

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
x^2	カイ 2 乗検定の統計の値
P	カイ 2 乗値に関連付けられた確率
DF	自由度

メニューキー

メニュー キーのオプションは、次のとおりです。

メニューキー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
統計	前に述べたように、初期設定の検定結果を表示します
期待	期待されるカウントを表示します
寄与率	各カテゴリによるカイ 2 乗値への寄与率のリストを表示します
OK	[数値表示]に戻ります

[期待]→[カウント]が選択されている場合、[数値表示]の入力には、期待されるカウントを指定する[ExpList]が[ProbList]の代わりに含まれ、[結果]画面のメニュー キーのラベルに[期待]は含まれません。

2 元表に基づく検定

メニュー名

[2 要因検定]

このオプションでは、カイ 2 乗分布を使用して、2 元表に含まれる求められたカウントのカテゴリカル データの適合度を検定します。

入力

[数値表示]の入力は、次のとおりです。

フィールド名	説明
ObsMat	2 元表の求められたカウント データの行列

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
x^2	カイ 2 乗検定の統計の値
P	カイ 2 乗値に関連付けられた確率
DF	自由度

メニューキー

メニュー キーのオプションは、次のとおりです。

メニューキー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
期待	期待されるカウントの行列を表示します。終了するには OK を押します
寄与率	各カテゴリによるカイ 2 乗値への寄与率の行列を表示します。終了するには OK を押します
OK	[数値表示]に戻ります

回帰の推論

HP Prime 電卓は、線形回帰の推論に基づいて検定を実行したり区間を計算したりすることができます。こうした計算は、t 分布に基づいて行われます。

線形 t 検定

メニュー名

[線形 t 検定]

このオプションでは、説明データのリストおよび応答データのリストに基づいて、真の線形回帰式に関する t 検定を実行します。[記号表示]の**[対立仮説]**フィールドで、対立仮説を選択する必要があります。

入力

[数値表示]の入力は、次のとおりです。

フィールド名	説明
Xlist	説明データのリスト
Ylist	応答データのリスト

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
T 検定	t 検定の統計の値
P	T 検定の統計に関連付けられた確率
DF	自由度
β_0	計算された回帰直線の切片
β_1	計算された回帰直線の傾き
serrLine	計算された回帰直線の標準誤差
serrSlope	計算された回帰直線の傾きの標準誤差
serrInter	計算された回帰直線の切片の標準誤差
r	データの相関係数
R^2	データの決定係数

メニューキー

メニュー キーのオプションは、次のとおりです。

メニューキー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
OK	[数値表示]に戻ります

傾きについての信頼区間

メニュー名

区間: 傾き

このオプションでは、説明データのリスト、応答データのリスト、および信頼水準に基づいて、真の線形回帰式の傾きについての信頼区間を計算します。[数値表示]でデータを入力して 計算 をタップした後、表示されるプロンプトで信頼水準を入力します。

入力

[数値表示]の入力は、次のとおりです。

フィールド名	説明
Xlist	説明データのリスト
Ylist	応答データのリスト
C	信頼水準（0 < C < 1）

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
C	入力の信頼水準
臨界 T	t の臨界値
DF	自由度
β_1	計算された回帰直線の傾き
serrSlope	計算された回帰直線の傾きの標準誤差
下側	傾きについての信頼区間の下側限界
上側	傾きについての信頼区間の上側限界

メニューキー

メニュー キーのオプションは、次のとおりです。

メニューキー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
OK	[数値表示]に戻ります

切片についての信頼区間

メニュー名

区間: 切片

このオプションでは、説明データのリスト、応答データのリスト、および信頼水準に基づいて、真の線形回帰式の切片についての信頼区間を計算します。[数値表示]でデータを入力して 計算 をタップした後、表示されるプロンプトで信頼水準を入力します。

入力

[数値表示]の入力は、次のとおりです。

フィールド名	説明
Xlist	説明データのリスト
Ylist	応答データのリスト
C	信頼水準（0 < C < 1）

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
C	入力の信頼水準
臨界 T	t の臨界値
DF	自由度
β_0	計算された回帰直線の切片
serrInter	回帰直線の y 切片の標準誤差
下側	切片についての信頼区間の下側限界
上側	切片についての信頼区間の上側限界

メニューキー

メニュー キーのオプションは、次のとおりです。

メニューキー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
OK	[数値表示]に戻ります

平均応答についての信頼区間

メニュー名

区間: 平均応答

このオプションでは、説明データのリスト、応答データのリスト、説明変数の値（X）、および信頼水準に基づいて、平均応答（ŷ）についての信頼区間を計算します。[数値表示]でデータを入力して 計算 をタップした後、表示されるプロンプトで信頼水準および説明変数の値（X）を入力します。

入力

[数値表示]の入力は、次のとおりです。

フィールド名	説明
Xlist	説明データのリスト
Ylist	応答データのリスト
X	平均応答および信頼区間を求める説明変数の値
C	信頼水準（0 < C < 1）

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
C	入力の信頼水準
臨界 T	t の臨界値
DF	自由度
ŷ	入力 X 値についての平均応答
serrŷ	ŷ の標準誤差
下側	平均応答についての信頼区間の下側限界
上側	平均応答についての信頼区間の上側限界

メニューキー

メニュー キーのオプションは、次のとおりです。

メニューキー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
OK	[数値表示]に戻ります

予測区間

メニュー名

[予測区間]

このオプションでは、説明データのリスト、応答データのリスト、説明変数の値（X）、および信頼水準に基づいて、将来の応答についての予測区間を計算します。[数値表示]でデータを入力して 計算 をタップした後、表示されるプロンプトで信頼水準および説明変数の値（X）を入力します。

入力

[数値表示]の入力は、次のとおりです。

フィールド名	説明
Xlist	説明データのリスト
Ylist	応答データのリスト
X	将来の応答および信頼区間を求める説明変数の値
C	信頼水準（0 < C < 1）

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
C	入力の信頼水準
臨界 T	t の臨界値
DF	自由度
ŷ	入力 X 値についての将来の応答
serrŷ	ŷ の標準誤差
下側	平均応答についての信頼区間の下側限界
上側	平均応答についての信頼区間の上側限界

メニュー キー

メニュー キーのオプションは、次のとおりです。

メニュー キー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
OK	[数値表示]に戻ります

ANOVA

メニュー名

ANOVA

このオプションは、数値データのリストに基づいて F 検定を使用して一元配置分散分析（ANOVA）を実行します。

入力

一元配置分散分析のための入力は、l1 ~ l4 にあるデータのリストです。l5 などに追加のリストを追加できます。

結果

計算 をタップした場合の結果は、次のとおりです。

結果	説明
F	検定の F 値
P	検定の F 値に関連付けられた確率
DF	検定の自由度
SS	処理平方和
MS	処理平均平方
DFerr	誤差自由度
SSerr	誤差平方和
MSerr	誤差平均平方

メニュー キー

メニュー キーのオプションは、次のとおりです。

メニュー キー	説明
その他	コピーして貼り付ける複数のセルを選択するためのメニューを開きます
OK	[数値表示]に戻ります

表を移動するには、カーソル キーを使用するかタップします。 その他 をタップするだけでなく、セルをタップしたまま指でドラッグして、コピーして貼り付けるための複数のセルを矩形選択することもできます。

14 Solve（解く）アプリケーション

Solve（解く）アプリケーションでは、最大 10 個の方程式または式を定義でき、それぞれには変数をいくつでも設定できます。シード値に基づいて、変数の 1 つに対する単一の方程式または式を解くことができます。また、再びシード値を使用して、連立方程式（1 次または非 1 次）を解くこともできます。

以下に示す、方程式と式の違いに注意してください。

- 方程式には等号が含まれます。その解は、方程式の両辺の値が同じになるような未知の変数の値です。
- 式には等号は含まれません。その解は、式がゼロの値になるような未知の変数の値である根です。

簡潔にするために、この章での用語「方程式」には、方程式と式の両方が含まれます。

解くは実数の場合のみ機能します。

解くアプリケーションでの作業の開始

解くアプリケーションでは、主要なアプリケーション表示をすべて使用できます。[記号表示]、[プロット表示]、[数値表示]の 3 つです。ただし、[数値表示]は、値の表を表示するのではなく数値として解くことのみを目的としているため、他のアプリケーションとは大きく異なります。

このアプリケーションでは、[記号表示]および[プロット表示]のメニュー ボタンはすべて使用できます。

1 つの方程式

自動車の速度を 100 m の距離で 16.67 m/s（60 kph）から 27.78 m/s（100 kph）まで上げるために必要な加速度を求めるとします。

解くべき方程式は以下のとおりです。

$V^2 = U^2 + 2AD$

この方程式で、V = 最終的な速度、U = 最初の速度、A = 必要な加速度、および D = 距離です。

Solve（解く）アプリケーションを開く

▲ Apps Info を押して、**[解く]**を選択します。

Solve（解く）アプリケーションは[記号表示]で開きます。この表示で、解くべき方程式を指定できます。

注記：組み込み変数に加えて、（ホーム表示または CAS のどちらかで）ユーザーが各自で作成した 1 つ以上の変数を使用できます。たとえば、「ME」という変数を作成した場合、それを次のように方程式に含めることができます：$Y^2 = G^2 + ME$。

他のアプリケーションで定義された関数も、Solve（解く）アプリケーションで参照できます。たとえば、Function（関数）アプリケーションで F1(X)を $X^2 + 10$ と定義した場合、Solve（解く）アプリケーションで「F1(X) = 50」と入力すると、方程式 $X^2 + 10 = 50$ を解くことができます。

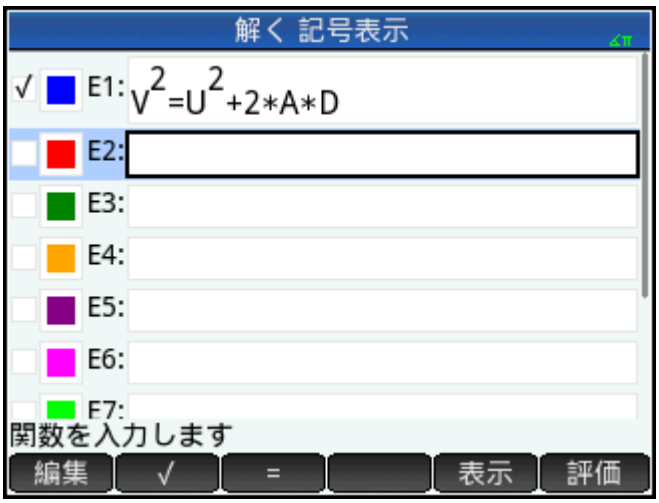
アプリケーションのクリアおよび方程式の定義

1. すでに定義されている方程式または式がすべて不要な場合は、 Shift Esc Clear を押します。

   OK をタップして、アプリケーションをクリアする操作を確定します。

2. 方程式を定義します。

   ALPHA alpha V x^2 Shift $\frac{\bullet}{=}$ ALPHA alpha U x^2 + Ans ; 2 ALPHA alpha A ALPHA alpha D Enter ≈



既知の変数の入力

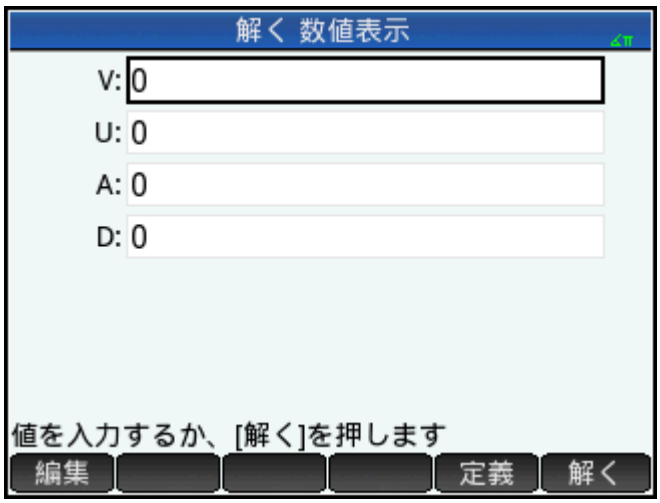
1. [数値表示]を表示します。

   Num ↳Setup

   ここで、既知の変数の値を指定し、解を求める変数を選択してから、 解く をタップします。

2. 既知の変数の値を入力します。

   27 [•] 78 Enter 16 [•] 67 Enter ▼ 100 Enter

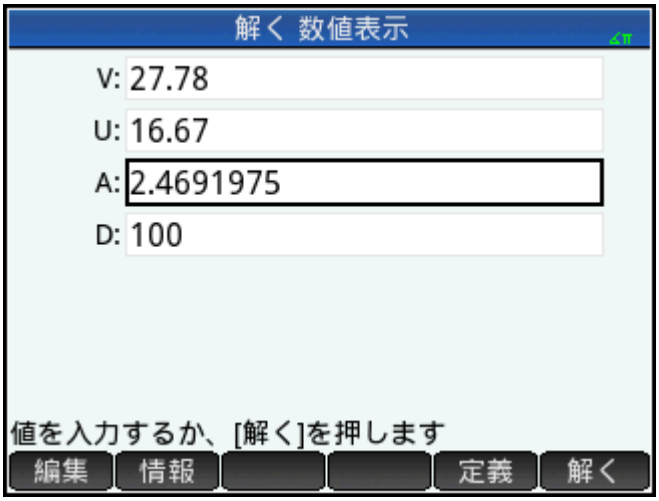


注記：[数値表示]を表示したときに、変数にこれらとは異なる値がすでに入力されている場合があります。これは、他の場所で変数に値が割り当てられている場合に発生します。たとえば、ホーム表示で「10」と入力し、 Sto ▸ をタップしてから「U」と入力すると、変数 U に 10 を割り当てたとします。その後、[数値表示]を開いて変数として U を含む方程式を解こうとすると、10 が U の初期値になります。これは、（アプリケーションまたはプログラムで）以前に行った何らかの計算で変数に値が指定された場合にも発生します。

事前に入力されているすべての変数をゼロにリセットするには、 Shift Esc Clear を押します。

未知の変数を求める

▲ 未知の変数 A を求めるには、カーソルを**[A]**ボックスに移動して 解く をタップします。



したがって、自動車の速度を 100 m の距離で 16.67 m/s（60 kph）から 27.78 m/s（100 kph）まで上げるために必要な加速度は、約 2.4692 m/s^2 になります。

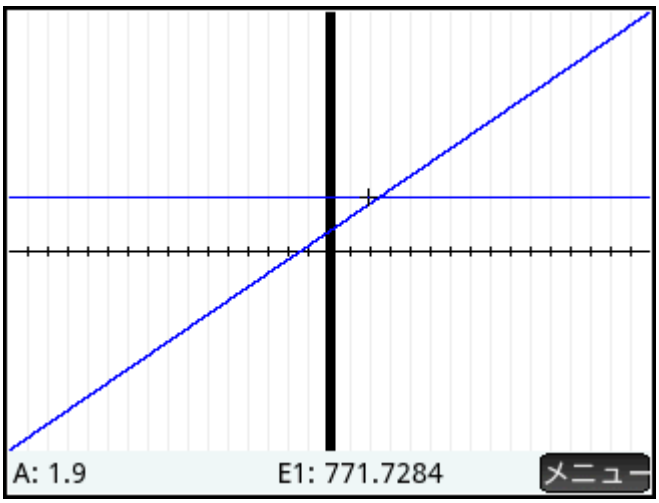
この方程式は、変数 A に関して一次方程式です。そのため、A の解はこれ以上存在しないと判断できます。このことは、方程式をプロットしてもわかります。

方程式のプロット

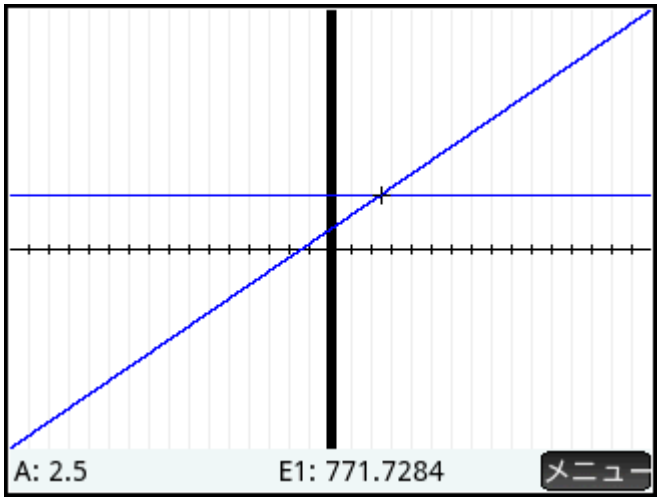
[プロット表示]には、解いた方程式の各辺について 1 つのグラフが表示されます。変数を[数値表示]で選択することによって、どの変数も独立変数として選択できます。そのため、この例では、A が強調表示されていることを確認してください。

現在の方程式は $V^2 = U^2 + 2AD$ です。[プロット表示]には、方程式の各辺について 1 つ、合計で 2 つの方程式がプロットされます。その 1 つが $Y = V^2$ で、V = 27.78 なので Y = 771.7284 になります。このグラフは水平線です。もう 1 つのグラフは $Y = U^2 + 2AD$ で、U = 16.67 および D = 100 なので Y = 200A + 277.8889 になります。このグラフも直線です。目的の解は、これらの 2 つの直線が交差する位置での A の値です。

1. 変数 A の方程式をプロットするには、 View Copy を押します。
2. **[自動スケール]**を選択します。
3. **[En の両辺]**を選択します（ここで、n は選択された方程式の番号です）。



4. 初期設定では、トレーサーが有効になっています。カーソル キーを使用して、交点に近付くまでどちらかのグラフに沿ってトレース カーソルを移動します。画面の左下隅付近に表示される A の値が、計算した A の値にほぼ一致することに注目してください。



[プロット表示]は、いくつかの解が考えられる場合に解の近似値を求めるときに便利です。目的の解（交点）の付近にトレース カーソルを移動してから、[数値表示]を開きます。[数値表示]で得られる解は、トレース カーソルに最も近い解です。

注記：設定した x 範囲および y 範囲から外れている部分のプロットをすばやく表示するには、画面を指で水平方向または垂直方向にドラッグします。

複数の方程式

[記号表示]では最大 10 個の方程式および式を定義し、系として一緒に解く方程式および式を選択することができます。たとえば、以下で構成される連立方程式を解くとします。

- $X^2 + Y^2 = 16$
- $X - Y = -1$

Solve（解く）アプリケーションを開く

1. Apps Info を押して、**[解く]**を選択します。
2. すでに定義されている方程式または式がすべて不要な場合は、Shift Esc Clear を押します。

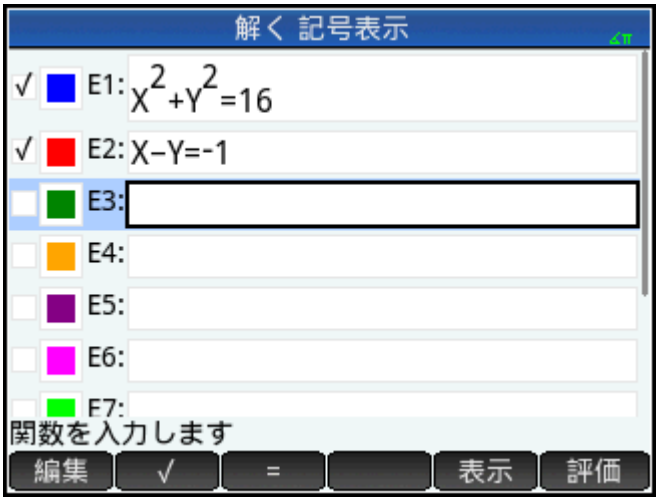
   OK をタップして、アプリケーションをクリアする操作を確定します。

方程式の定義

▲ 方程式を定義します。

ALPHA alpha X x^2 + Ans ALPHA alpha Y x^2 Shift = 16 Enter

ALPHA alpha X − Base ALPHA alpha Y Shift = +/− 1 Enter

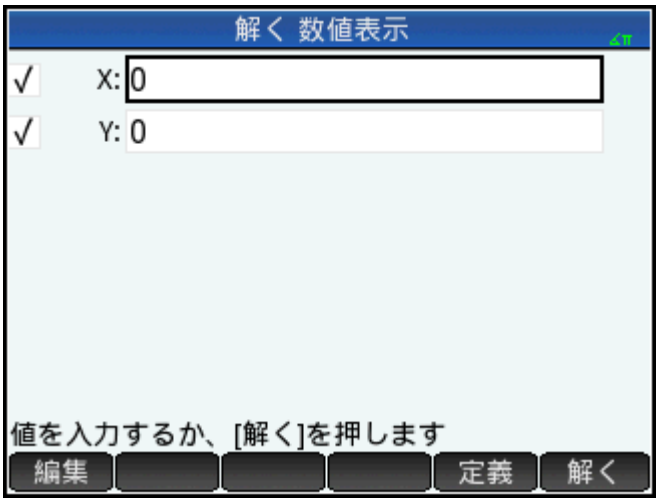


両方の方程式を満たす X および Y の値を求めているため、両方の方程式が選択されていることを確認します。

シード値の入力

1. [数値表示]を表示します。

   1 つの方程式の例とは異なり、この例ではどの変数にも値は指定されません。どれかの変数についてシード値を入力するか、電卓で自動的に解を求めることができます（通常、シード値とは、可能な場合に、他の値ではなくそのシード値に最も近い解を求めるよう電卓に指示するための値です）。この例では、X = 2 の近傍での解を探します。

2. [X]フィールドにシード値を入力します。

   たとえば、「2」と入力してから OK をタップします。

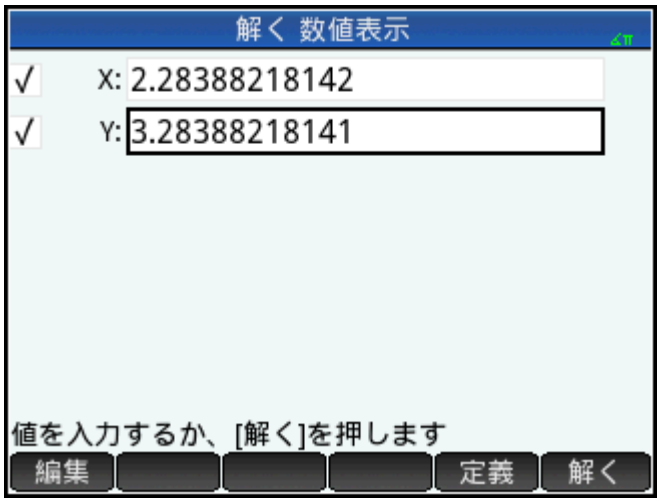
   （解が 1 つ存在する場合は）その解が表示されますが、複数の解が存在する場合はそのことが通知されません。その他に考えられる解を求めるには、シード値を変更します。

3. 解を求める変数を選択します。この例では、X と Y の両方の値を求めるので、両方の変数が選択されていることを確認します。

注記：3 つ以上の変数がある場合は、複数の変数についてシード値を入力できます。

未知の変数を求める

▲ 解く をタップして、選択された各方程式を満たす X = 2 付近の解を求めます。



解が見つかった場合は、選択された各変数の横に解が表示されます。

制限事項

[記号表示]で複数の方程式が選択されている場合、方程式をプロットすることはできません。

HP Prime 電卓は、複数の解が存在することを通知しません。特定の値付近に別の解が存在すると思われる場合は、その値をシードとして使用して操作を繰り返します（上で説明した例では、X のシード値として – 4 を入力すると別の解が見つかります）。

場合によっては、Solve（解く）アプリケーションは、解を探すときに乱数のシードを使用します。つまり、複数の解が存在する場合、どのシードによってどの解が導かれるかは必ずしも予測できません。

解の情報

単一の方程式を解いている場合、 解く をタップした後のメニューに 情報 ボタンが表示されます。 情報 をタップすると、見つかった解（存在する場合）に関する情報を含むメッセージが表示されます。このメッセージをクリアするには、 OK をタップします。

メッセージ	意味
Zero（ゼロ）	電卓が備える 12 桁の精度の範囲内で、方程式の両辺が等しくなるポイントか、式がゼロになるポイント（根）が見つかりました
Sign Reversal（符号反転）	方程式の両辺の符号が逆になるポイントが 2 つ見つかりましたが、値がゼロになるポイントがその 2 つの間に見つかりませんでした。同様に、式の場合は、式の値の符号が異なるものの、正確にゼロにはなりませんでした。2 つの値が近傍値である（12 桁目で 1 だけ異なる）か、方程式が 2 つのポイント間で実数解を持ちません。Solve（解く）では、値または差がよりゼロに近いポイントが返されます。方程式または式が常に実

メッセージ	意味
	数解を持つ場合、このポイントは Solve（解く）が求めた実際の解の最良近似です
Extremum（極値）	式の値が極小値（正の値の場合）または極大値（負の値の場合）の近似値となるポイントが見つかりました。このポイントは解である可能性も、解はない可能性もあります または 電卓で表示できる最大値である 9.99999999999E499 で、解を探す処理を停止しました **注記：[極値]**メッセージは、解がない可能性が高いことを示しています。これを検証するには[数値表示]を使用してください（また、表示されるすべての値が誤っている可能性があることに注意してください）
Cannot find solution（解が見つかりません）	選択された方程式または式を満たす値がありません
Bad Guess(es)（推測が間違っています）	推測の初期値が方程式の領域外にあります。そのため、解が実数ではなかったか、解が原因でエラーが発生しました
Constant?（定数？）	方程式の値は、標本を抽出したすべてのポイントで同一です

15 Linear Solver（線形ソルバー）アプリケーション

Linear Solver（線形ソルバー）アプリケーションでは、一次方程式のセットを解くことができます。このセットには、2 つまたは 3 つの一次方程式を含めることができます。

2 つの方程式のセットでは、各方程式を ax + by = k の形式にする必要があります。3 つの方程式のセットでは、各方程式を ax + by + cz = k の形式にする必要があります。

各方程式について a、b、および k（さらに 3 つの方程式のセットでは c）の値を指定すると、アプリケーションは x および y（さらに 3 つの方程式のセットでは z）の解を求めようとします。

HP Prime 電卓は、解が見つからない場合または解が無限に存在する場合には、そのことを通知します。

Linear Solver（線形ソルバー）アプリケーションでの作業の開始

以下の例では、次の方程式のセットを定義してから、未知の変数の解を求めています。

- 6x + 9y + 6z = 5
- 7x + 10y + 8z = 10
- 6x + 4y = 6

Linear Solver（線形ソルバー）アプリケーションを開く

▲ Apps Info を押した後、**[線形ソルバー]**を選択します。

[数値表示]でアプリケーションが開きます。

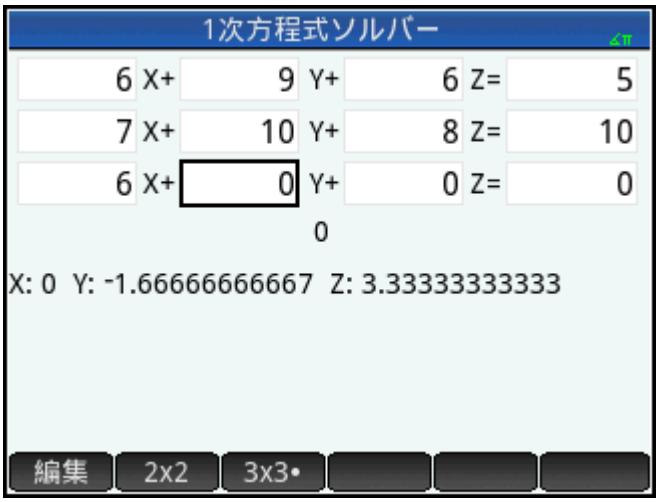
注記：最後に Linear Solver（線形ソルバー）アプリケーションを使用したときに 2 つの方程式の解を求めた場合は、2 つの方程式用の入力フォームが表示されます。3 つの方程式のセットを解くには、3x3 をタップします。これで、入力フォームに 3 つの方程式が表示されるようになります。

方程式を定義して解く

1. 解を求める方程式を定義するには、各方程式での各変数の係数、および定数項を入力します。カーソルが最初の方程式の x のすぐ左に配置され、x の係数（6）を入力できる状態になります。

   係数を入力し、OK をタップするか Enter を押します。

2. カーソルが次の係数に移動します。その係数を入力し、OK をタップするか Enter を押します。すべての方程式を定義するまで、同様の操作を繰り返します。



ソルバーで解の生成に必要な値がすべて入力されると、それらの解が画面の下部付近に表示されます。この例では、ソルバーは、最後の方程式の最初の係数が入力されるとすぐに、x、y、および z の解を求めることができました。

残りの既知の値を入力するごとに、解が変化します。次の図は、すべての係数および定数を入力した後の最終的な解を示しています。

2x2 系を解く

3 つの方程式用の入力フォームが表示されているときに、2 つの方程式のセットを解く場合は、以下の操作を行います。

▲ 2x2 をタップします。

注記：変数など、数値的な結果に解決されるどのような式も入力できます。変数の名前を入力してください。

メニュー項目

メニュー項目は次のとおりです。

メニュー項目	説明
編集	値を追加または変更できる入力行にカーソルを移動します。フィールドを選択し、値を入力して、Enter ≈ を押すこともできます。カーソルは自動的に次のフィールドに移動するため、そのフィールドで次の値を入力して Enter ≈ を押すことができます
2x2	2 つの変数と 2 つの式の連立一次方程式を解くページを表示します。有効になると、2x2• に変わります
3x3	3 つの変数と 3 つの式の連立一次方程式を解くページを表示します。有効になると、3x3• に変わります

16 Parametric（パラメトリック）アプリケーション

Parametric（パラメトリック）アプリケーションでは、パラメトリック方程式を操作できます。これらは、x と y の両方が t の関数として定義されている方程式です。x = f(t)および y = g(t)の形式を取ります。

Parametric（パラメトリック）アプリケーションでの作業の開始

Parametric（パラメトリック）アプリケーションでは、主要なアプリケーション表示をすべて使用できます。[記号表示]、[プロット表示]、[数値表示]の 3 つです。

このアプリケーションでは、各表示のメニュー ボタンもすべて使用できます。

この章では、パラメトリック方程式 x(T) = 8sin(T)および y(T) = 8cos(T)を例に説明します。これらの方程式によって、円が生成されます。

Parametric（パラメトリック）アプリケーションを開く

▲ Apps Info を押して、**[パラメトリック]**を選択します。

[記号表示]でパラメトリック アプリケーションが開きます。これは定義用の表示です。ここで、操作するパラメトリック方程式を記号で定義（指定）します。

[プロット表示]のグラフおよび[数値表示]の数値データは、ここで定義する記号関数から計算されます。

関数の定義

関数を定義するためのフィールドは 20 個あります。これらのフィールドには、X1(T)から X9(T)および X0(T)まで、および Y1(T)から Y9(T)および Y0(T)までのラベルが付いています。X の関数はそれぞれ、Y の関数と対になっています。

1. どれかのペアをタップするかそこまでスクロールで移動して、使用する関数のペアを選択します。新しい関数を入力する場合は、その関数を入力します。既存の関数を編集する場合は、

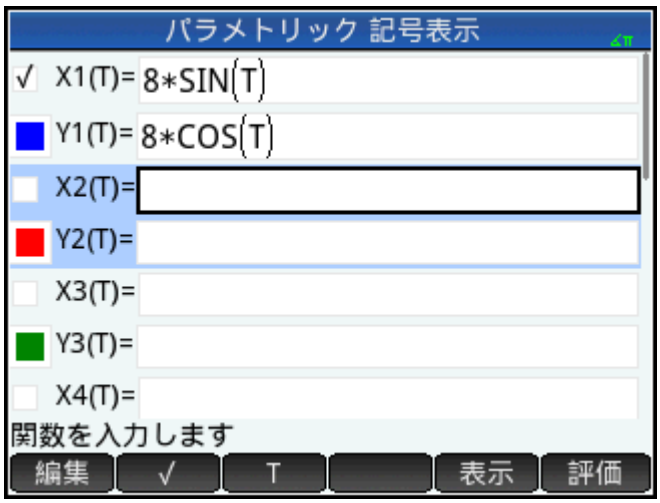
   編集 をタップして変更を加えます。関数の定義または変更が完了したら、 Enter ≈ を押します。

2. 2 つの式を定義します。

   8 SIN ASIN G $xt\theta n$ Define D Enter ≈

   8 COS ACOS H $xt\theta n$ Define D Enter ≈

   $xt\theta n$ Define D キーを押すと、現在のアプリケーションに応じた変数が入力されます。このアプリケーションでは、T が入力されます。



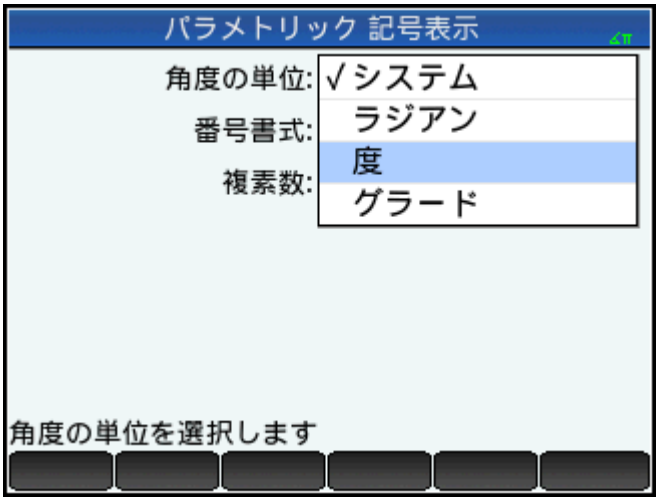
3. 以下のいずれかの操作を行います。
   - 関数のプロット時の色を変更する
   - 従属関数を評価する
   - 操作対象にしない定義をクリアする
   - 変数、数学コマンド、および CAS コマンドを定義に組み込む

この例ではこれらの操作は省略しますが、これらは便利な操作です。詳しくは、[記号表示]の共通操作の説明を参照してください。

角度の単位の設定

角度の単位を度に設定するには、以下の操作を行います。

1. Shift Symb Setup を押します。
2. **[角度の単位]**を選択してから**[度]**を選択します。



角度の単位は、**[ホーム設定]**画面で設定することもできます。ただし、ホーム設定はシステム全体に適用されます。ホーム表示ではなくアプリケーション内で角度の単位を設定すると、そのアプリケーションのみに設定が制限されます。

プロットのセットアップ

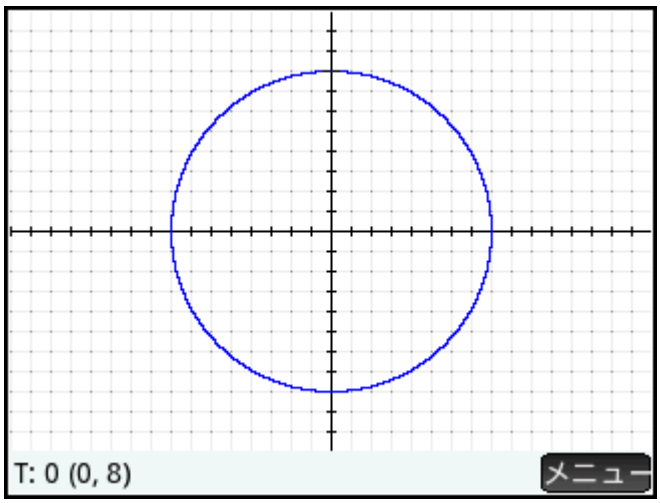
1. [プロットのセットアップ]表示を開くには、 Shift Plot Setup を押します。
2. プロットをセットアップするには、適切なグラフ作成オプションを指定します。この例では、T が 0°から 360°まで 5°ずつ変化するように**[T 範囲]**フィールドおよび**[T ステップ]**フィールドを設定します。

   2 つ目の**[T 範囲]**フィールドを選択し、次のように入力します。

   360 OK 5 OK

関数のプロット

▲ Plot Setup を押します。



グラフの操作

メニュー ボタンから、プロットを操作するための以下の共通ツールを利用できます。

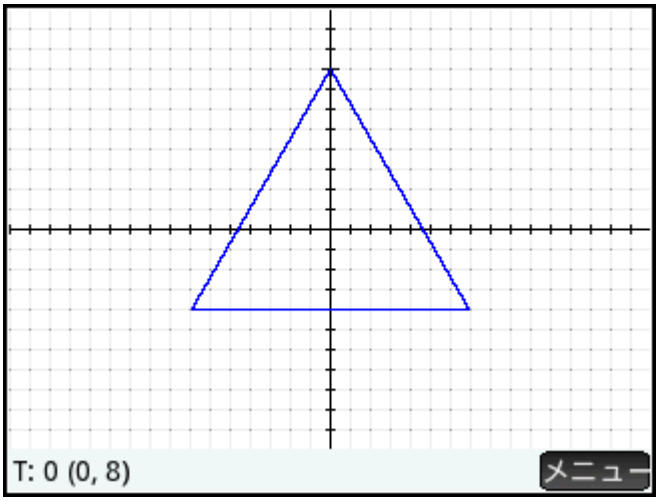
- ズーム：一連のズーム オプションを表示します（ + Ans ; キーおよび − Base ; キーを使用して拡大および縮小することもできます）。
- トレース：有効な場合、トレース カーソルをプロットに沿って移動できるようになります（このとき、カーソルの座標が画面の下部に表示されます）。
- 移動：T の値を指定すると、対応する x 座標および y 座標にカーソルが移動します。
- 定義：プロットの元になっている関数を表示します。

これらのツールは、[プロット表示]の共通操作です。

通常、プロットを変更するには、[記号表示]でプロットの定義を変更します。ただし、[プロットのセットアップ]のパラメーターを変更することで、一部のプロットを変更できます。たとえば、プロットのセットアップ パラメーターを 2 つ変更するだけで、円ではなく三角形をプロットできます。[記号表示]の定義は変更されません。これを行うには、以下の操作を行います。

1. Shift Plot Setup を押します。
2. **[T ステップ]**を**[120]**に設定します。
3. ページ 1/3 をタップします。
4. **[方式]**メニューから**[Fixed-Step セグメント]**を選択します。

5. Plot Setup を押します。



円ではなく三角形が表示されます。これは、**[T ステップ]**の新しい値によって、ほぼ連続する 5°間隔ではなく 120°間隔で点がプロットされるようになるためです。また、**[Fixed-Step セグメント]**を選択することで、120°間隔の点が線分で接続されます。

[数値表示]の表示

1. Num Setup を押します。

2. **[T]**列にカーソルを置いて新しい値を入力し、 OK をタップします。入力した値まで表がスクロールします。

パラメトリック 数値表示

T	X1	Y1
0	0	8
0.1	1.3962626927E-2	7.9999878153
0.2	2.7925211322E-2	7.99995126126
0.3	4.1887710651E-2	7.99989033798
0.4	5.5850082384E-2	7.99980504564
0.5	0.069812283987	7.99969538451
0.6	8.3774272930E-2	7.99956135493
0.7	9.7736006682E-2	7.99940295728

0

ズーム | その他 | 移動 | | 定義 |

独立変数を拡大または縮小することもできます（これによって、連続する値の増分が変更されます）。これらは[数値表示]の共通操作です。

[プロット表示]と[数値表示]を組み合わせ、この 2 つの表示を横に並べて表示することができます。

17 Polar（極座標）アプリケーション

Polar（極座標）アプリケーションでは、極方程式を操作できます。極方程式とは、ある点から原点への線分と極軸で形成される角度 q に関して、原点（0,0）からその点への距離 r を定義する方程式です。このような方程式は、r = f(θ)の形式を取ります。

Polar（極座標）アプリケーションでの作業の開始

Polar（極座標）アプリケーションでは、6 種類の標準的なアプリケーション表示を使用できます。この章では、Polar（極座標）アプリケーションで使用されるメニュー ボタンについても説明します。

この章では、式 $5\pi\cos(\theta/2)\cos(\theta)^2$ を例に説明します。

Polar（極座標）アプリケーションを開く

▲ Apps Info を押して、**[極座標]**を選択します。

[記号表示]でアプリケーションが開きます。

[プロット表示]のグラフおよび[数値表示]の数値データは、ここで定義する記号関数から計算されます。

関数の定義

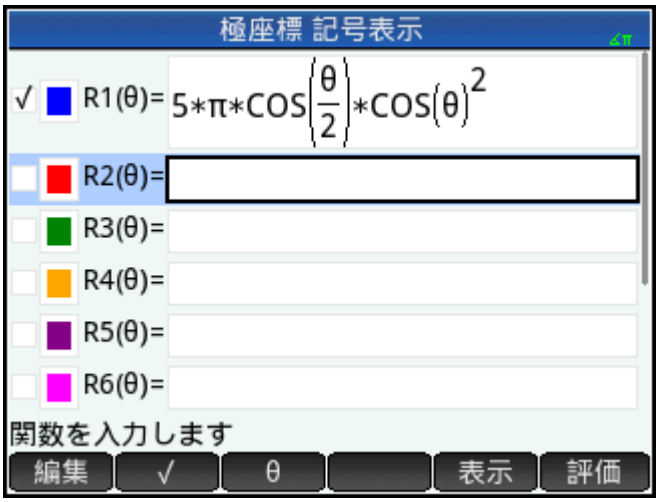
極関数を定義するためのフィールドは 10 個あります。これらのフィールドには、R1(θ) ~ R9(θ)および R0(θ)のラベルが付いています。

1. 使用するフィールドをタップするかそこまでスクロールして選択します。新しい関数を入力する場合は、その関数を入力します。既存の関数を編集する場合は、 編集 をタップして変更を加えます。関数の定義または変更が完了したら、 Enter ≈ を押します。

2. 式 $5\pi\cos(\theta/2)\cos(\theta)^2$ を定義します。

5 Shift 3 COS xtθn ÷ 2 ▶ ▶ COS xtθn ▶ x^2 Enter

xtθn キーを押すと、現在のアプリケーションに応じた変数が入力されます。このアプリケーションでは、θ が入力されます。



3. 必要に応じて、初期設定値以外のプロットの色を選択します。これを行うには、関数セットの左側にあるカラー ボックスを選択し、 選択 をタップして、カラー ピッカーから色を選択します。

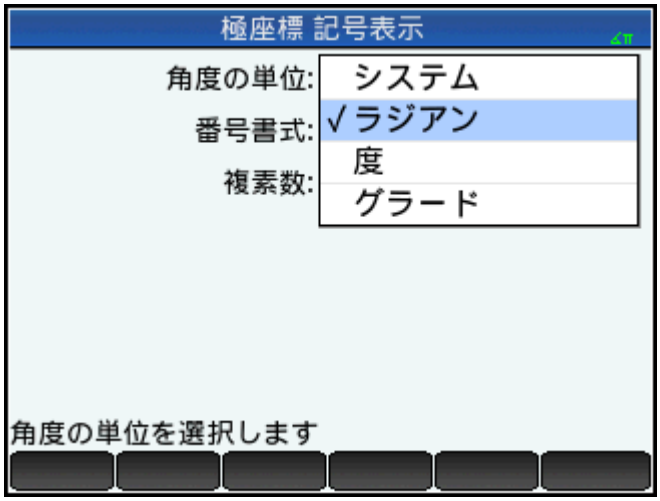
定義の追加、定義の変更、および従属定義の評価は、[記号表示]の共通操作です。

角度の単位の設定

角度の単位をラジアンに設定するには、以下の操作を行います。

1. Shift Symb Setup を押します。

2. **[角度の単位]**を選択してから**[ラジアン]**を選択します。



これらは、[記号のセットアップ]表示の共通操作です。

プロットのセットアップ

1. [プロットのセットアップ]表示を開くには、Shift Plot Setup を押します。

2. プロットをセットアップするには、適切なグラフ作成オプションを指定します。この例では、独立変数の範囲の上限を次のように**[4π]**に設定します。

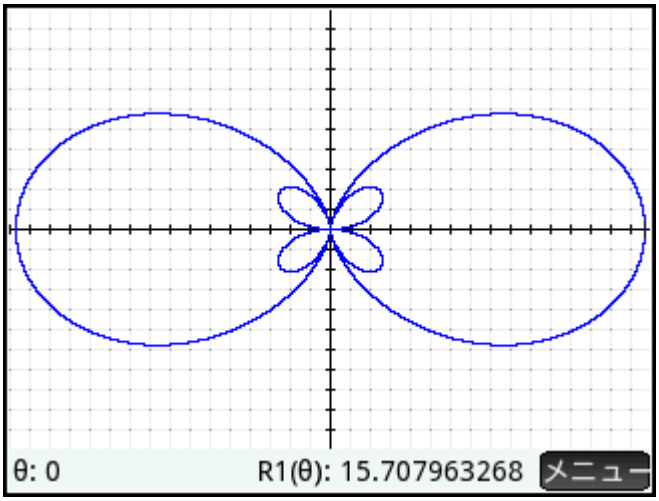
   2 つ目の**[T 範囲]**フィールドを選択し、次のように入力します。

   2 つ目の**[θ 範囲]**フィールドを選択して、「4 Shift π 3 OK 」と入力します。

[プロット表示]の外観は、[プロット表示]の共通操作を使用してさまざまな方法で構成できます。

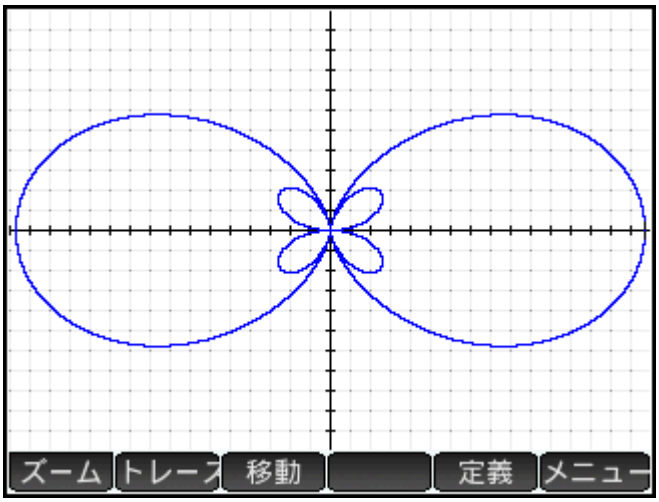
式のプロット

▲ Plot Setup を押します。



グラフの操作

▲ [プロット表示]メニューを表示するには、 メニュー を押します。



ズームやトレースなど、グラフを操作できるオプションがいくつか表示されます。特定の q 値を入力することで、その値に直接ジャンプすることもできます。[移動]画面が表示され、入力行に入力した数字が表示されます。 OK をタップして操作を確定します（ 移動 をタップして、ジャンプ先の値を指定することもできます）。

プロットされている極方程式が 1 つだけの場合は、 定義 をタップすることで、プロットの生成元の方程式を表示できます。プロットされている方程式が複数ある場合は、 ▲ または ▼ を押して目的のプロットにトレースカーソルを移動してから、 定義 をタップします。

プロットの操作は、[プロット表示]の共通操作です。

[数値表示]の表示

1. Num Setup を押します。

   [数値表示]に、θ および R1 の値の表が表示されます。[記号表示]で複数の極関数を指定および選択している場合は、関数ごとに評価の列（R2、R3、R4 など）が表示されます。

2. **[θ]**列にカーソルを置いて新しい値を入力し、 OK をタップします。入力した値まで表がスクロールします。

独立変数を拡大または縮小することもできます（これによって、連続する値の増分が変更されます）。このオプションやその他のオプションは[数値表示]の共通操作です。

[プロット表示]と[数値表示]を組み合わせ、この 2 つの表示を横に並べて表示することができます。

18 数列アプリケーション

HP Prime の[数列]アプリでは、明示的または再帰的に数列を定義できます。再帰的定義では、U(N－1)のみ、または U(N－1)と U(N－2)の両方を用いて U(N)を定義できます。同様に、U(N)のみを用いて U(N＋1)を定義することも、U(N)と U(N＋1)の両方を用いて U(N＋2)を定義することもできます。最後に、N は 1（初期設定値）、0、または任意の正の整数から開始できます。

[記号表示]では、必要に応じて、最初の 2 つのボックスに数列の最初の 2 項の数値を入力します。明示的に定義された数列の場合は、値を空にすることができます。再帰的に定義された数列の場合は、定義の特性に応じて、少なくとも 1 つの値を入力する必要があります。

注記：値のラベルは、**[オプション]**ボックスで選択されている N の初期値に応じて変わります。

3 番目のボックスには、記号定義を入力します。

[オプション]ボックスでは、記号定義の条件を選択します。初期設定では、U(N)が選択されています。つまり、N、U(N-1)、U(N-1)と U(N-2)の両方、またはこれら 3 つのオプションの組み合わせを用いた記号定義で U(N)が定義されます。その他に、U(N+k)を選択できます。これは、U(N)を用いた U(N+1)の記号定義、または U(N+1)と U(N)を用いた U(N+2)の記号定義です。

[オプション]ボックスの横にある別のボックスを使用すると、N の初期値を入力できます。この初期値は 0 または任意の正の整数にできます。

以下の例では、フィボナッチ数列を U1(1) = 1、U1(2) = 1、および U1(N) = U1(N－1) + U1(N－2)として定義しています。[オプション]の初期設定は U(N)、N の初期値は 1 です。この例は305 ページの数列アプリケーションでの作業の開始で使用されます。

以下の例では、フィボナッチ数列を U1(1) = 1、U1(2) = 1、および U1(N＋2) = U1(N) + U1(N＋1)として定義しています。[オプション]には U(N+k)を選択しており、N の初期値は 1 です。

数列アプリケーションでの作業の開始

以下の例では、よく知られたフィボナッチ数列を使用して説明します。この数列では、3 番目の項以降の各項が先行する 2 項の和になります。この例では、3 つの数列フィールド（初項、2 番目の項、および以降のすべての項を生成するための規則）を指定します。

数列アプリケーションを開く

▲ Apps Info を押して、**[数列]**を選択します。

[記号表示]でアプリケーションが開きます。

式の定義

次のフィボナッチ数列を定義するには、以下の操作を行います。

$U_1 = 1$、$U_2 = 1$、$U_n = U_{n-1} + U_{n-2}$（$n > 2$ の場合）

1. **[U1(1)]**フィールドで、数列の 1 番目の項および N の開始値を指定します。

   1 Enter

2. **[U1(2)]**フィールドで、数列の 2 番目の項を指定します。

   1 Enter

3. **[U1(N)]**フィールドで、先行する 2 項から数列の n 番目の項を求めるための式を指定します（画面の下部にあるボタンを使用すると、簡単に入力できます）。

   U1 (N−1) + U1 (N−2) Enter

4. オプションで、グラフの色を選択します。

プロットのセットアップ

1. [プロットのセットアップ]表示を開くには、Shift Plot Setup を押します。

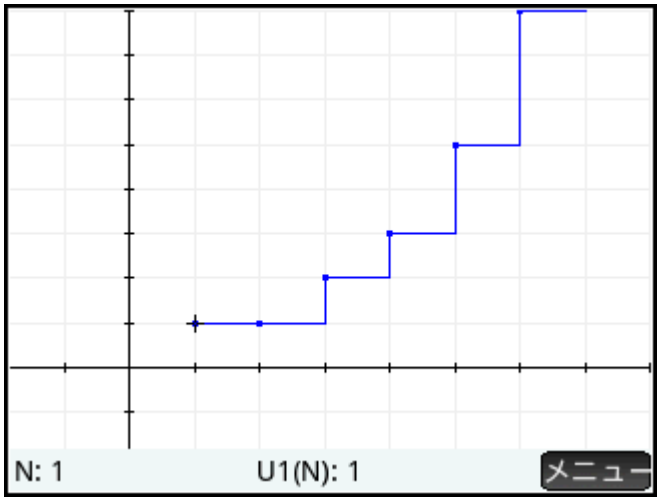
2. すべての設定を初期設定値にリセットするには、Shift Esc Clear を押します。

3. **[]**シーケンスプロット]メニューから**[階段状]**を選択します。

4. **[X 範囲]**の最大値および**[Y 範囲]**の最大値を、それぞれ**[8]**に設定します（次の図を参照してください）。

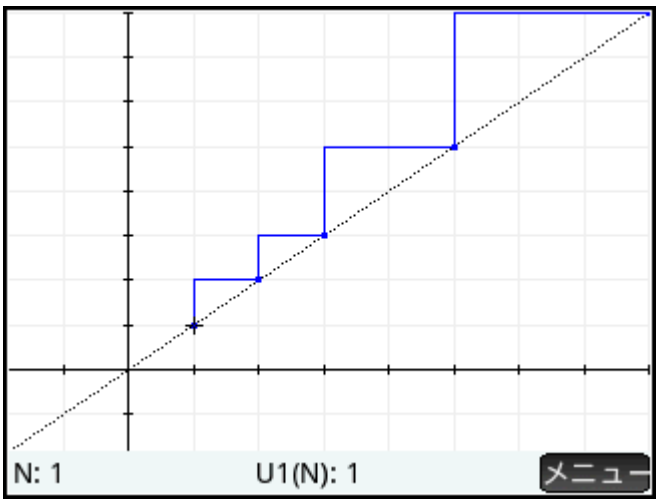
数列のプロット

1. Plot Setup を押します。



2. クモの巣状のオプションを使用して数列をプロットするには、[プロットのセットアップ]表示（Shift Plot Setup）に戻り、**[シーケンスプロット]**メニューから**[クモの巣]**を選択します。

3. Plot Setup を押します。



グラフの操作

メニュー ボタンから、プロットを操作するための以下のような共通ツールを利用できます。

- ズーム：プロットを拡大または縮小します
- トレース：グラフに沿ってトレースします
- 移動：指定された n 値に移動します
- 定義：数列の定義を表示します

これらのツールは、[プロット表示]の共通操作です。

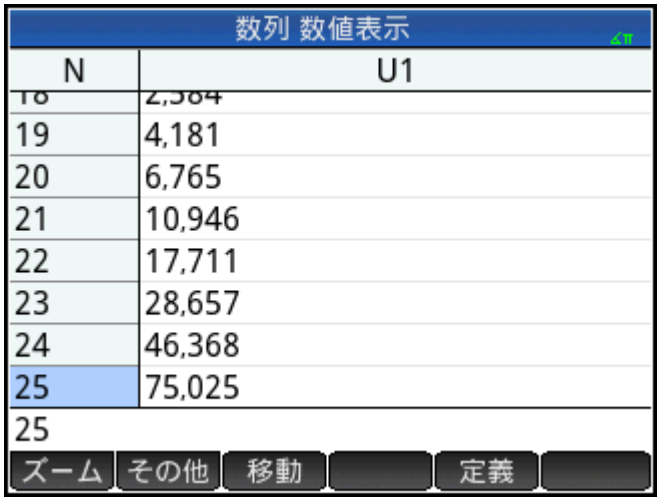
View Copy を押すと、画面分割および自動スケールのオプションも使用できます。

[数値表示]の表示

1. [数値表示]を表示します。

   Num Setup

2. **[N]**列の任意の場所にカーソルを置いて新しい値を入力し、 OK をタップします。



入力した値まで値の表がスクロールします。その後、数列の対応する値を確認できます。上の図では、フィボナッチ数列の 25 番目の値が 75,025 であることが示されています。

値の表の操作

[数値表示]では、表を操作するための以下のような共通ツールを利用できます。

- ズーム：連続する値の増分を変更します
- 定義：数列の定義を表示します
- 列：表示する数列の数を選択します

これらのツールは、[数値表示]の共通操作です。

View Copy を押すと、画面分割および自動スケールのオプションも使用できます。

値の表のセットアップ

[数値のセットアップ]表示にはほとんどのグラフ作成アプリケーションに共通のオプションが用意されています（ただし、数列の領域が一連の自然数であるため、ズーム倍率はありません）。これらは、[数値のセットアップ]表示の共通操作です。

別の例：明示的に定義された数列

以下の例では、数列の n 番目の項を n 自体によって定義します。この場合、最初の 2 項のどちらも、数値で入力する必要はありません。

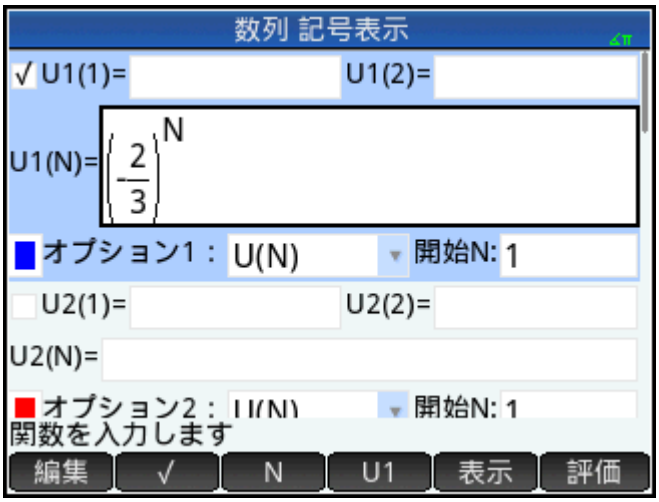
式の定義

▲ U1(N) = (– 2/3)^N を定義します。

次のように U1N を選択します。

「( ) +/– Units」と入力し、□/□ を選択します。

「2 ▼ 3 ▶ ▶ x^y N Enter ≈」と入力します。

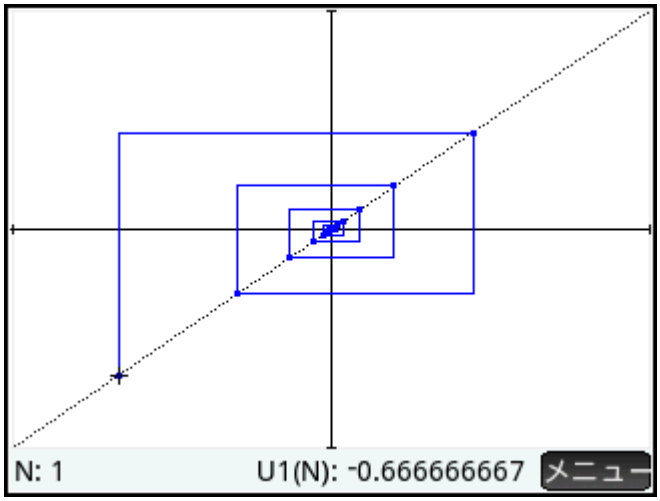


プロットのセットアップ

1. [プロットのセットアップ]表示を開くには、 Shift Plot Setup を押します。
2. すべての設定を初期設定値にリセットするには、 Shift Esc Clear を押します。
3. **[シーケンスプロット]**をタップして、**[クモの巣]**を選択します。
4. 次の図のように、**[X 範囲]**と**[Y 範囲]**の両方を**[－1, 1]**に設定します。

数列のプロット

▲ Plot Setup を押します。



Enter ≈ を押すと、上の図の点線が表示されます。ボタンをもう一度押すと、点線が非表示になります。

値の表の操作

1. Num Setup を押します。
2. 列 をタップし、**[1]**を選択して、数列の値を表示します。

19 Finance（金融）アプリケーション

Finance（金融）アプリケーションでは、貨幣の時間価値（TVM）および償却の問題を解くことができます。このアプリケーションを使用して、複利計算を実行したり、償却表を作成したりできます。

複利とは、累積的な利息（すでに得ている利息に対する利息）のことです。指定した複利計算期間で、与えられた元本に対して得られた利息が元本に加えられた後、総計額に対して一定の割合で利息が得られます。複利に関係する財務計算には、普通預金口座、住宅ローン、年金基金、リース、年金などがあります。

Finance（金融）アプリケーションでの作業の開始

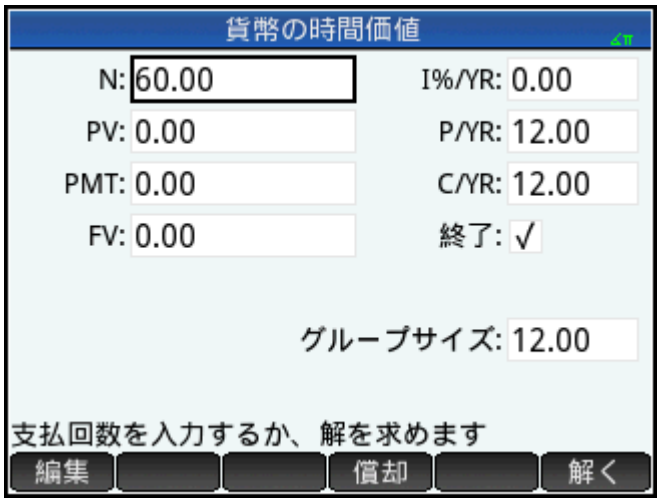
車を購入するため、毎月複利計算される年率 5.5%で 5 年ローンを組むとします。車の購入価格は 19,500 ドルで、頭金は 3,000 ドルです。まず、月々いくら支払う必要があるのかを計算します。次に、月々の支払いの最高額が 300 ドルである場合に返済可能なローンの最高額を計算します。支払いは、第 1 期の期末から開始するとします。

1. Finance（金融）アプリケーションを開くには、 Apps Info を押して、**[金融]**を選択します。

   [数値表示]でアプリケーションが開きます。

2. **[N]**フィールドに「5 × 12」と入力して、 Enter ≈ を押します。

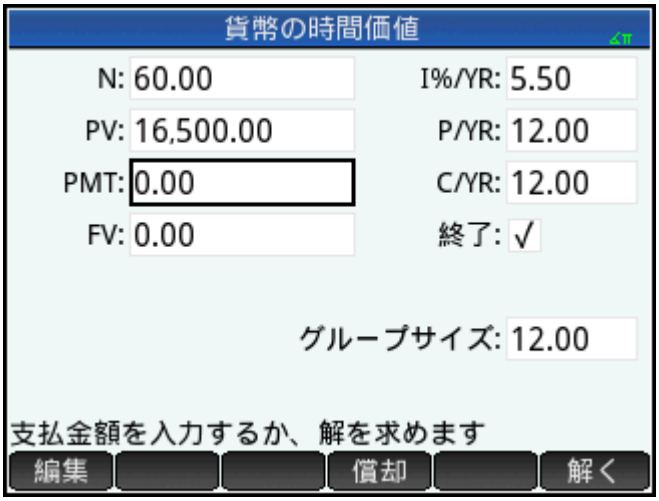
   フィールドに計算結果（60）が表示されます。これは、5 年間の月数です。



3. **[I%/YR]**フィールドに 5.5（金利）を入力して、 Enter ≈ を押します。

4. **[PV]**フィールドに「19500 − 3000」と入力して、 Enter ≈ を押します。これは、購入価格から頭金を除いたローンの現在価値です。

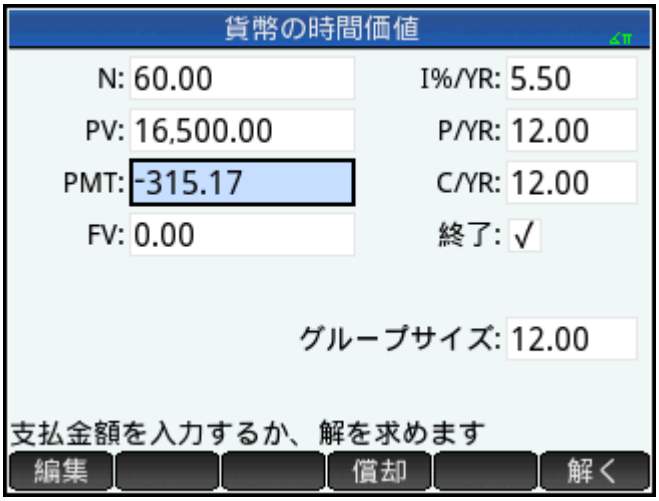
5. **[P/YR]**と**[C/YR]**の両方を 12（初期値）のままにします。支払いオプションとして**[期末]**が選択されたままにします。また、将来価値である**[FV]**も**[0]**のままにします（ローンの将来価値が最終的に 0 になることを目標とするため）。



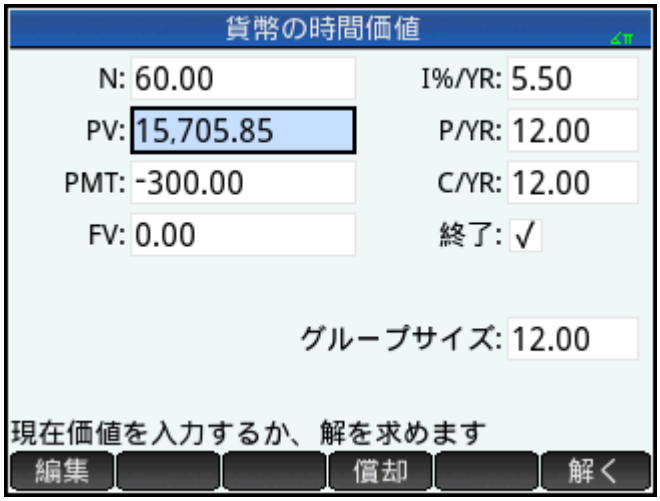
6. カーソルを**[PMT]**フィールドに移動し、 解く をタップします。PMT の値の計算結果は –315.17 です。つまり、月々の支払額は 315.17 ドルになります。

   支払い義務のある金額であることを示すため、PMT の値は負になっています。

   PMT の値が 300 を超えており、毎月支払うことが可能な金額を超えていることに注意してください。そのため、計算を再実行する必要があります。今度は PMT の値を – 300 に設定し、新しい PV の値を計算します。



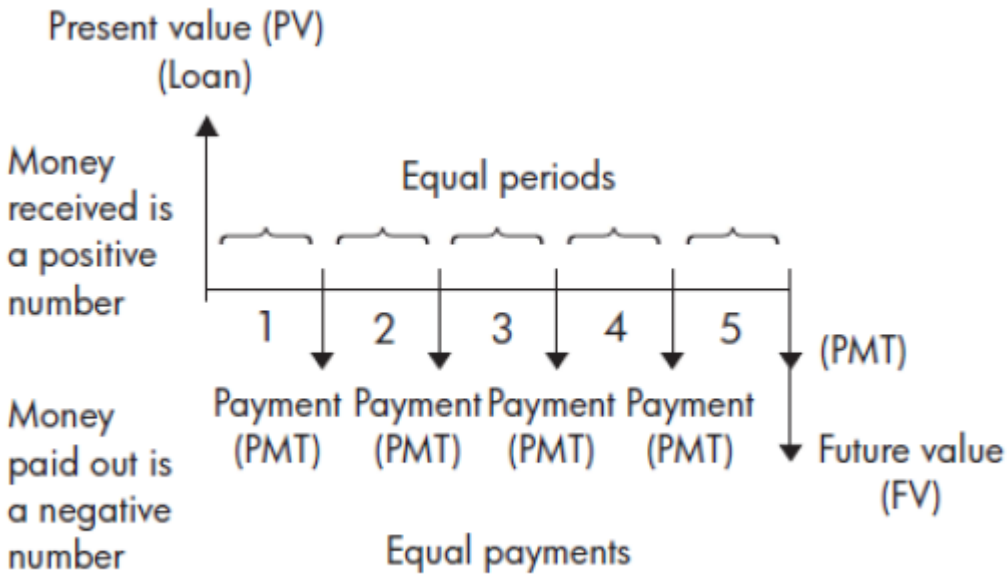
7. [PMT]フィールドに「 +/− 300」と入力し、カーソルを**[PV]**フィールドに置いて、 解く をタップします。



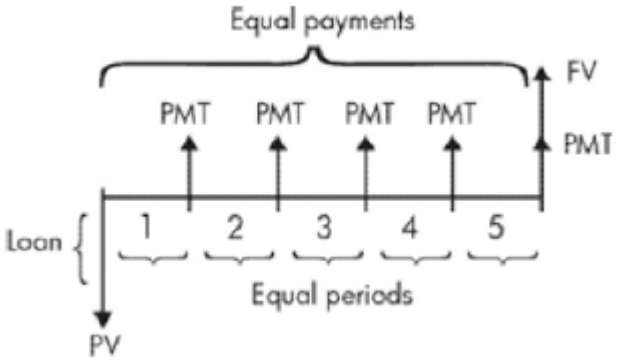
PV の値の計算結果は 15,705.85 です。これが、借りることができる最高額になります。そのため、3,000 ドルの頭金を含めると、購入する余裕があるのは価格が 18,705.85 ドルまでの車です。

キャッシュ フロー ダイアグラム

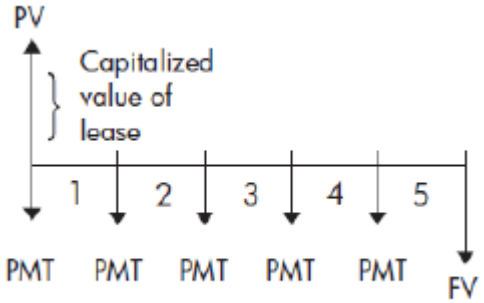
取り引きによる TVM の変動は、キャッシュ フロー ダイアグラムで表すことができます。キャッシュ フロー ダイアグラムは、複利計算期間を表す均等な線分に分割された時系列の図です。矢印がキャッシュ フローを表します。これらのフローは、貸し手と借り手のどちらの視点に立つかによって、正（上向きの矢印）になる場合も、負（下向きの矢印）になる場合もあります。次のキャッシュ フロー ダイアグラムは、借り手の視点から見たローンを示しています。



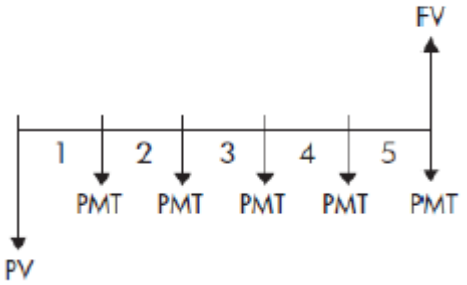
次のキャッシュ フロー ダイアグラムは、貸し手の視点から見たローンを示しています。



キャッシュ フロー ダイアグラムでは、複利計算期間に関連していつ支払いが行われるかも示されます。次の図は、各期の期首に発生する支払いリース料を示しています。



次の図は、各期の期末に口座に振り込まれる入金額（PMT）を示しています。



貨幣の時間価値（TVM）

貨幣の時間価値（TVM）の計算では、今日得られる 1 ドルは将来のある時点で得られる 1 ドルより価値があるという考え方を活用します。今日の 1 ドルを一定の金利で投資すれば、将来の 1 ドルでは生み出せない利益を生み出すことができます。TVM の原則は、金利、複利、および利益率という考え方の基礎となっています。

TVM 変数としては、次の 7 つがあります。

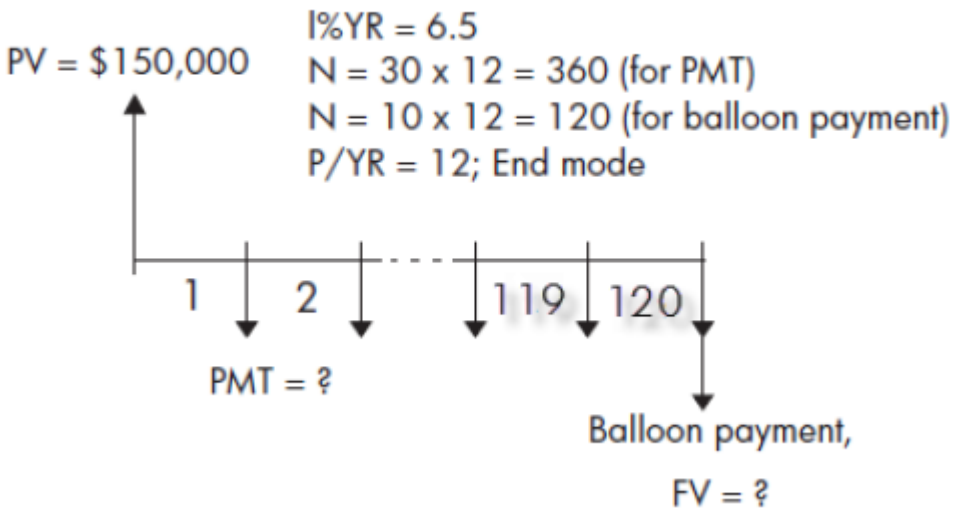
変数	説明
N	複利計算期間または支払いの合計数

変数	説明
1%/YR	名目年金利（または投資率）。この金利が 1 年あたりの支払数（P/YR）によって除算され、複利計算期間あたりの名目金利が算出されます。これが、TVM 計算で実際に使用される金利です
PV	最初のキャッシュ フローの現在価値。貸し手または借り手にとっては、PV はローンの金額です。投資家にとっては、PV は初期投資額です。PV は常に最初の期間の始めに発生します
P/YR	年間の支払い回数
PMT	定期的な支払額。支払額は各期で同じであり、TVM 計算では毎回支払いが行われることを前提とします。支払い時期は、各複利計算期間の期首または期末にすることができます。どちらにするかは、**[期末]**オプションを選択またはクリアすることで決定します
C/YR	年間の複利計算期間の数
FV	取り引きの将来価値。つまり、最終キャッシュ フローの金額、またはそれ以前の一連のキャッシュ フローの複合価値です。ローンの場合、これは（定期的な支払い以外の）最終一括返済の金額です。投資の場合、これは投資期間終了時の投資の価値です

別の例：TVM 計算

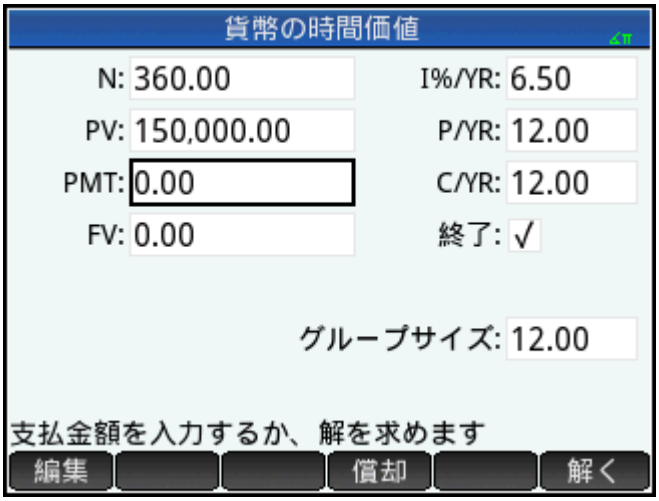
30 年の 150,000 ドルの住宅ローンを 6.5%の年利率で組んだとします。10 年後にその家を売り、最終一括返済でローンを返済しようと考えています。最終一括返済の金額、つまり、10 年間支払った後の住宅ローンの価値を求めます。

次のキャッシュ フロー ダイアグラムは、最終一括返済の住宅ローンの例です。



1. Finance（金融）アプリケーションを開くには、 Apps Info を押して、**[金融]**を選択します。
2. すべてのフィールドを初期設定値に戻すには、 Shift Esc Clear を押します。

3. 下の図のように、既知の TVM 変数を入力します。



4. **[PMT]**を選択し、 解く をタップします。[PMT]フィールドに – 984.10 と表示されます。つまり、月々の支払額は 948.10 ドルです。

5. 10 年後の住宅ローンの最終一括返済額または将来価値（FV）を求めるには、**[N]**に「120」と入力し、**[FV]**を選択して、 解く をタップします。

[FV]フィールドに – 127,164.19 と表示されます。これは、ローンの将来価値（まだ支払い義務のある金額）が 127,164.19 ドルであることを示しています。

償却

償却の計算では、1 回の支払いまたは一連の支払いで元本および利息に適用される金額を求めます。また、TVM 変数も使用します。

償却の計算

1. Finance（金融）アプリケーションを開くには、 Apps Info を押して、**[金融]**を選択します。

2. 年間の支払回数（**[P/YR]**）を指定します。

3. 支払いを各期の期首に行うか、期末に行うかを指定します。

4. **[I%/YR]**、**[PV]**、**[PMT]**、および**[FV]**の値を入力します。

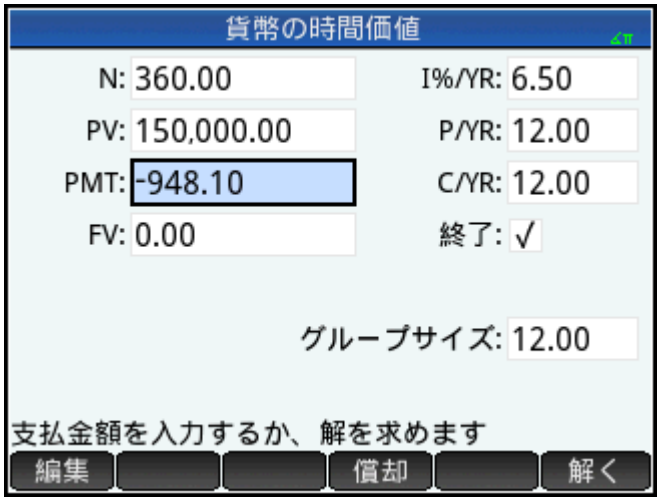
5. **[グループサイズ]**ボックスに、償却期間あたりの支払回数を入力します。初期設定で、グループサイズは年間償却を表す**[12]**です。

6. 償却 をタップします。計算機に償却表が表示されます。この表には、償却期間ごとに、利息および元本に適用される金額、およびローンの残金が表示されます。

住宅ローンの例での償却

前述の最終一括返済を行う住宅ローンの例のデータ（317 ページの別の例：TVM 計算を参照してください）を使用して、元本に適用された金額、利息として支払われた金額、および最初の 10 年が経過した後（つまり、12 × 10 = 120 回の支払いの後）の残金を計算します。

1. 次の図に示されている内容とデータが一致することを確認します。

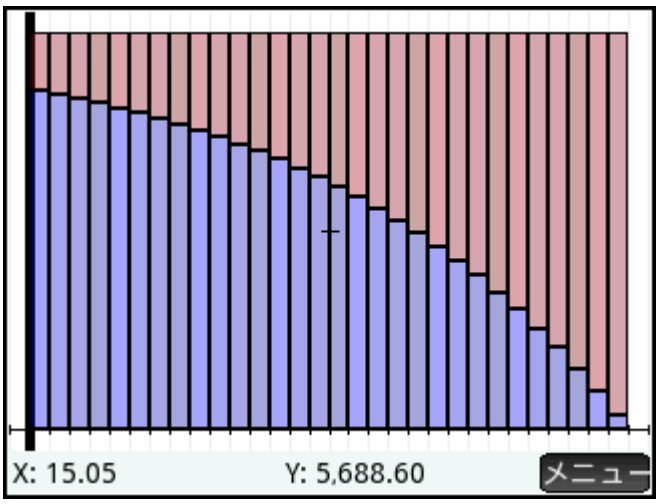


2. 償却 をタップします。

3. 支払いグループ 10 まで表を下にスクロールします。10 年後には、元本から 22,835.53 ドル、利息として 90,936.47 ドルが支払われており、最終一括返済額は 127,164.47 ドルになっています。

償却グラフ

▲ Plot Setup を押して、償却スケジュールをグラフで表示します。



各支払いグループの終了時の残金が棒の高さで示されています。支払いグループの期間中に減少した元本の金額および支払われた利息が、画面の下部に表示されます。上の例では、最初の支払いグループが選択されています。これは、12 回の支払いグループの最初のグループ（または最初の年が終わった時点でのローンの状態）を表します。その年の終わりまでに、元本は 1,676.57 ドル減少し、利息として 9,700.63 ドルが支払われています。

その他の支払いグループの期間中に減少した元本の金額および支払われた利息を表示するには、◀ または ▶ をタップします。

20 Triangle Solver（三角ソルバー）アプリケーション

Triangle Solver（三角ソルバー）アプリケーションでは、三角形のある辺の長さまたはある角の角度を、その他の辺の長さ、角の角度、またはその両方について指定した情報から計算できます。

指定可能な 6 つの値（3 辺の長さおよび 3 つの角の角度）のうち少なくとも 3 つを指定しないと、このアプリケーションでその他の値を計算することはできません。さらに、指定する値のうち少なくとも 1 つを長さにする必要があります。たとえば、2 辺の長さおよび 1 つの角度を指定したり、2 つの角度および 1 つの長さを指定したり、3 つの長さをすべて指定したりできます。どの場合も、アプリケーションによって残りの値が計算されます。

HP Prime 電卓は、解が見つからない場合または指定するデータが不足している場合には、そのことを通知します。

直角三角形の長さおよび角度を求める場合は、 ⊿ をタップすると、より単純な入力フォームを使用できます。

Triangle Solver（三角ソルバー）アプリケーションでの作業の開始

以下の例では、既知の 2 辺（長さが 4 および 6）が 30 度の角度で交わる三角形について、未知の辺の長さを計算します。

Triangle Solver（三角ソルバー）アプリケーションを開く

1. Apps Info を押した後、**[三角ソルバー]**を選択します。

   [数値表示]でアプリケーションが開きます。

   ラジアン

2. 以前の計算による不要なデータが存在している場合、 Shift Esc Clear を押せばすべてクリアできます。

角度の単位の設定

角度単位モードが適切であることを確認します。初期設定では、アプリケーションは角度モードで起動します。ユーザーが持っている角度情報がラジアンで、現在の角度単位モードが角度の場合は、モードを角度に変更してからソルバーを実行します。必要なモードに応じて 度 または ラジアン をタップします（このボタンはトグル ボタンです）。

注記：辺の長さのラベルは**[a]**、**[b]**、および**[c]**、角度のラベルは**[A]**、**[B]**、および**[C]**になります。既知の値を適切なフィールドに入力することが重要です。ここでの例では、2 つの辺の長さおよびそれらの辺が交わる角度がわかっています。したがって、辺**[a]**および**[b]**の長さを指定した場合は、角度を**[C]**として入力する必要があります（a と b が交わる角度は C であるため）。そうではなく、長さを**[b]**および**[c]**として入力した場合は、角度を**[A]**として指定する必要があります。既知の値をどこに入力すればよいのかを判断するには、電卓の表示を確認してください。

既知の値の指定

▲ 値を知っているフィールドに移動し、その値を入力して、OK をタップするか、Enter ≈ を押します。この作業をすべての既知の値について繰り返します。

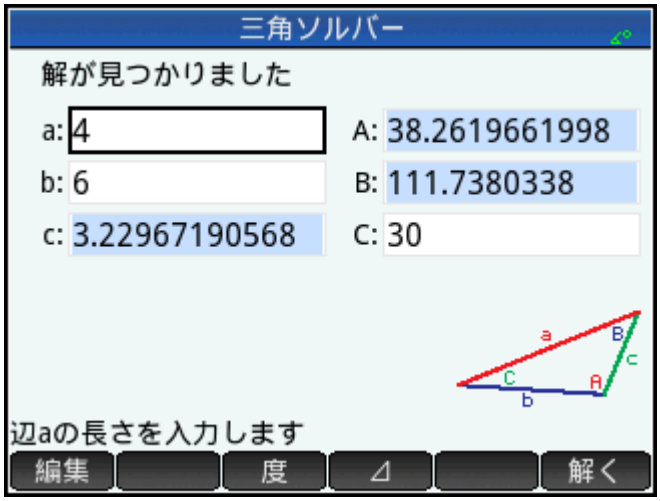
a. **[a]**ボックスに 4 を入力して、Enter ≈ を押します。

b. **[b]**ボックスに 6 を入力して、Enter ≈ を押します。

c. **[C]**ボックスに 30 を入力して、Enter ≈ を押します。

未知の値を求める

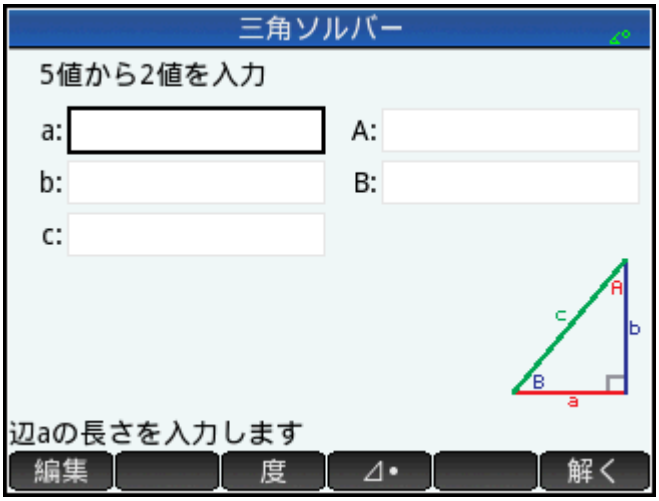
▲ 解く をタップします。



このアプリケーションは未知の変数の値を表示します。上の図に示されているように、この例では、未知の辺の長さは 3.22967…になっています。また、他の 2 つの角度も計算されています。

三角形のタイプの選択

Triangle Solver（三角ソルバー）アプリケーションの入力フォームは 2 つあります。一般的な入力フォーム、および直角三角形に特化したより単純なフォームです。一般的な入力フォームが表示されていて、調査対象が直角三角形である場合は、 ⊿ をタップしてより単純な入力フォームを表示します。一般的な入力フォームに戻るには、 ⊿• をタップします。調査対象の三角形が直角三角形でないか、三角形のタイプが不明な場合は、一般的な入力フォームを使用する必要があります。

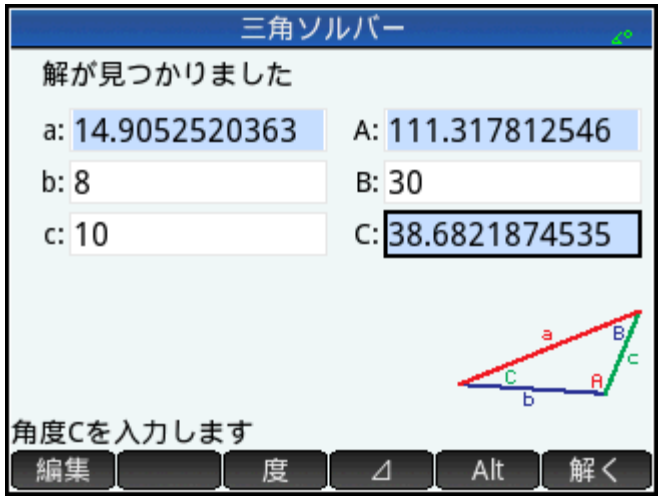


特殊な場合

解が 1 つに決まらない場合

2 つの辺と 1 つの隣接する鋭角を入力し、2 つの解が存在している場合、最初に表示されるのは 1 つだけです。

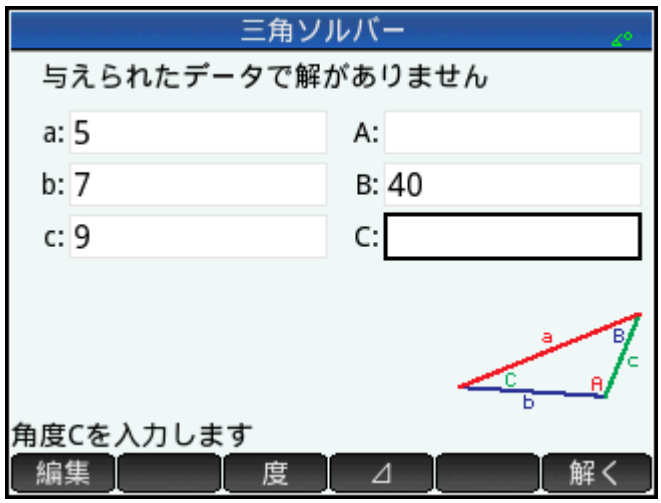
この場合、 Alt ボタンが表示されます（次の図を参照してください）。 Alt をタップすると 2 つ目の解が表示され、 Alt をもう一度タップすると最初の解に戻ります。



与えられたデータで解がない

一般的な入力フォームを使用して 4 つ以上の値を入力した場合、それらの値が互いに矛盾しているため、指定されたすべての値を持つ三角形が存在しない可能性があります。このような場合は、画面上に**[与えられたデータで解がありません]**と表示されます。

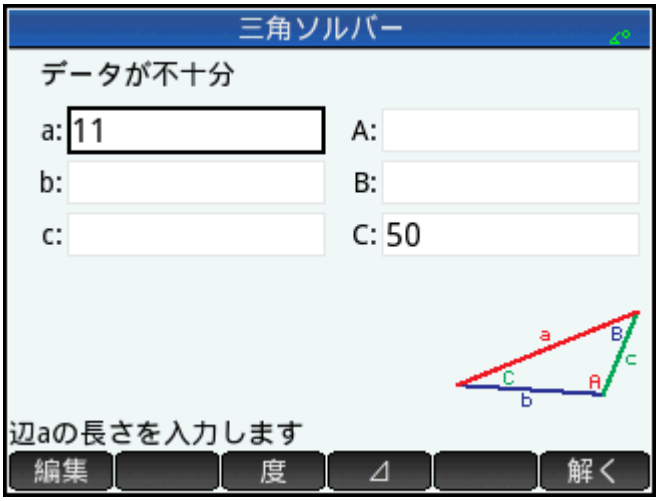
（直角三角形用の）より単純な入力フォームを使用して 3 つ以上の値を入力した場合も、同じような状況になります。



データが不十分

一般的な入力フォームを使用している場合、少なくとも 3 つの値を指定しないと、Triangle Solver（三角ソルバー）が三角形の残りの属性を計算できません。指定した値が 2 つ未満の場合は、画面上に**[データが不十分]**と表示されます。

（直角三角形用の）より単純な入力フォームを使用している場合は、少なくとも 2 つの値を指定する必要があります。


三角ソルバー
データが不十分
a: 11
A:
b:
B:
c:
C: 50
a
B
c
C
A
b
辺aの長さを入力します
編集
度
解く

21 エクスプローラー アプリケーション

エクスプローラー アプリケーションは 3 つあります。これらのアプリケーションは、関数のパラメーターとその関数のグラフの形状の関係を参照できるように設計されています。エクスプローラー アプリケーションは、次のとおりです。

- 1 次エクスプローラー

  1 次関数の操作用です

- 2 次エクスプローラー

  2 次関数の操作用です

- 三角法エクスプローラー

  正弦関数の操作用です。

操作モードは 2 つあります。グラフ モードと方程式モードです。グラフ モードでは、グラフを操作し、操作に応じて方程式がどのように変化するかを確認できます。方程式モードでは、方程式を操作し、操作に応じてグラフ表示がどのように変化するかを確認できます。各エクスプローラー アプリケーションには、操作対象としていくつかの方程式やグラフが用意されているほか、テスト モードも用意されています。テスト モードでは、方程式をグラフに一致させられるかをテストします。

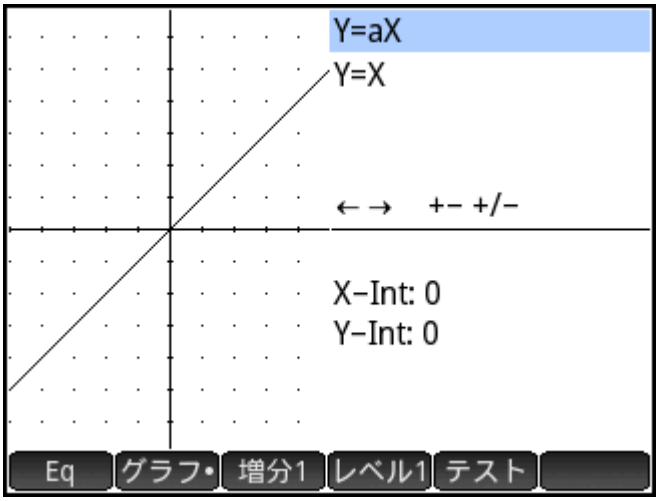
Linear Explorer（線形エクスプローラー）アプリケーション

The Linear Explorer1 次エクスプローラーアプリケーションでは、a および b の値の変化に応じた y=ax および y = ax + b のグラフの反応を調べることができます。

アプリケーションを開く

Apps Info を押し、**[1 次エクスプローラー]**を選択します。

画面の左半分には、1 次関数のグラフが表示されます。右半分には、操作する方程式の一般的な形式が上部に表示され、その形式の現在の方程式がその下に表示されます。グラフまたは方程式を操作するために使用できるキーが、その方程式の下に表示されます。下部には x 切片および y 切片が表示されます。



操作可能な 1 次方程式のタイプ（レベル）は 2 つあります。y = ax および y = ax + b です。どちらかを選択するには、 レベル1 または レベル2 をタップします。

グラフまたは方程式の操作するために使用できるキーは、選択したレベルによって異なります。たとえば、レベル 1 の方程式の画面には次が表示されます。

←→ + − +/ −

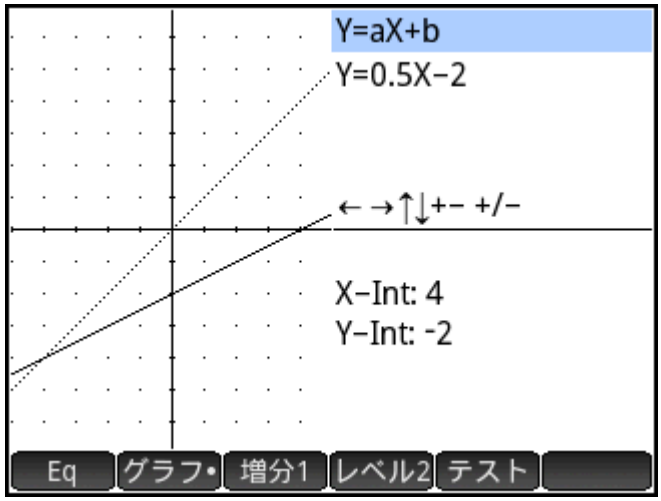
したがって、 ◀ 、 ▶ 、 + Ans ; 、 − Base ; 、および +/− |x| M を押すことができます。レベル 2 の方程式を選択した場合、画面には次が表示されます。

←→ ↑ ↓ + − +/ −

したがって、 ◀ 、 ▶ 、 ▲ 、 ▼ 、 + Ans ; 、 − Base ; 、および +/− |x| M を押すことができます。

グラフ モード

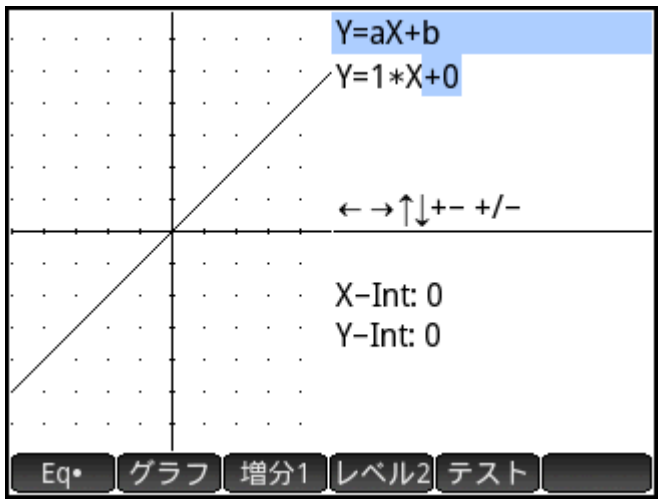
アプリケーションを開いたときはグラフ モードになっています（画面下部の**[グラフ]**ボタンにドットが表示されます）。グラフ モードでは、 ▲ キーおよび ▼ キーを押すと、グラフが垂直方向に平行移動し、その結果直線の y 切片が変化します。をタップすると、垂直方向の平行移動の増分間隔が変化します。 ◀ キーおよび ▶ キー（または − Base ; および + Ans ; ）を押すと、傾きが減少および増加します。 +/− |x| M を押すと、傾きの符号が変更されます。



1 次関数の形式が画面の右上に表示され、グラフに一致する現在の方程式がそのすぐ下に表示されます。グラフを操作すると、方程式が更新されてその変更内容が反映されます。

方程式モード

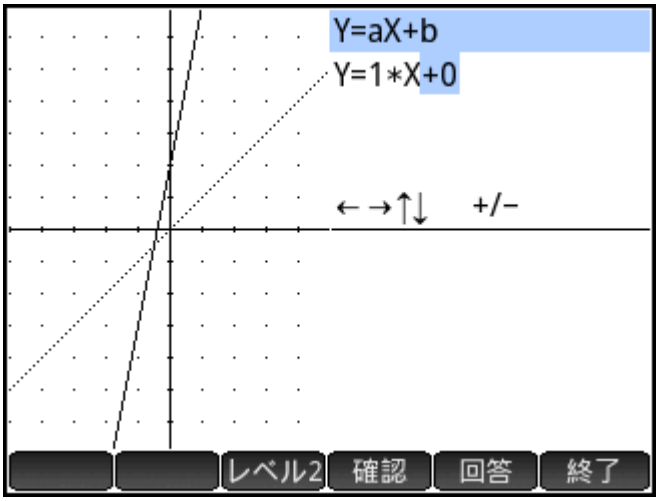
Eq をタップすると、方程式モードに入ります。画面下部の[方程式]ボタンにドットが表示されます。



方程式モードでは、カーソル キーを使用して、方程式内のパラメーター間を移動したり、それらの値を変更したりして、表示されているグラフへの影響を確認します。 または を押すと、選択したパラメーターの値が増減します。 または を押すと、別のパラメーターが選択されます。 +/− を押すと、*a* の符号が変更されます。

テスト モード

テスト をタップすると、テスト モードに入ります。テスト モードでは、表示されたグラフと方程式を一致させられるかをテストします。テスト モードは、カーソル キーを使用して方程式の各パラメーター値を選択および変更するという点で、方程式モードと似ています。その目標は、表示されたグラフに一致させることです。



このアプリケーションでは、ユーザーが選択したレベルで決まる形式からランダムに選択された 1 次関数のグラフが表示されます（レベルを変更するには レベル1 または レベル2 をタップします）。ここで、カーソル キーを押し、パラメーターを選択して値を設定します。準備が整ったら 確認 をタップし、表示されたグラフと方程式を正しく一致されられたかを確認します。

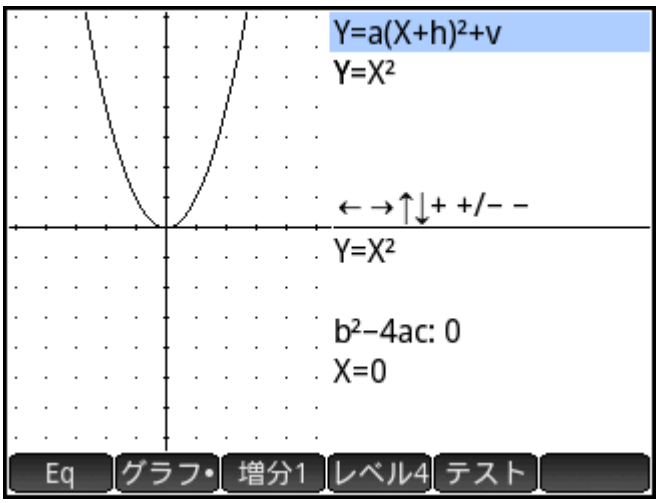
正解を表示するには 回答 をタップし、テスト モードを終了するには 終了 をタップします。

2 次エクスプローラーアプリケーション

2 次エクスプローラーアプリケーションでは、a、h、および v の値の変化に応じた $y = a(x+h)^2 + v$ の反応を調べることができます。

アプリケーションを開く

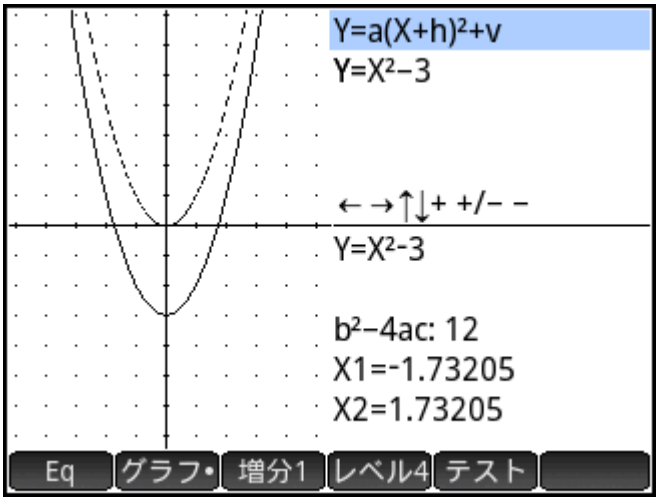
Apps Info を押し、**Quadratic Explorer**（2 次エクスプローラー）を選択します。



画面の左半分には、2 次関数のグラフが表示されます。右半分には、操作する方程式の一般的な形式が上部に表示され、その形式の現在の方程式がその下に表示されます。グラフまたは方程式を操作するために使用できるキーが、その方程式の下に表示されます（これらは、ユーザーが選択した方程式のレベルに応じて変わります）。これらのキーの下に表示されるのは、方程式、判別式（つまり $b^2 - 4ac$）、および 2 次方程式の根です。

グラフ モード

このアプリケーションはグラフ モードで開かれます。グラフ モードでは、利用可能なキーを使ってグラフのコピーを操作します。元のグラフはそのまま残されるため（ただし点線に変換されます）、操作の結果を簡単に確認できます。



操作可能な 2 次方程式の一般的な形式は 4 つあります。

y = *ax*²（レベル 1）

y = (*x*+*h*)²（レベル 2）

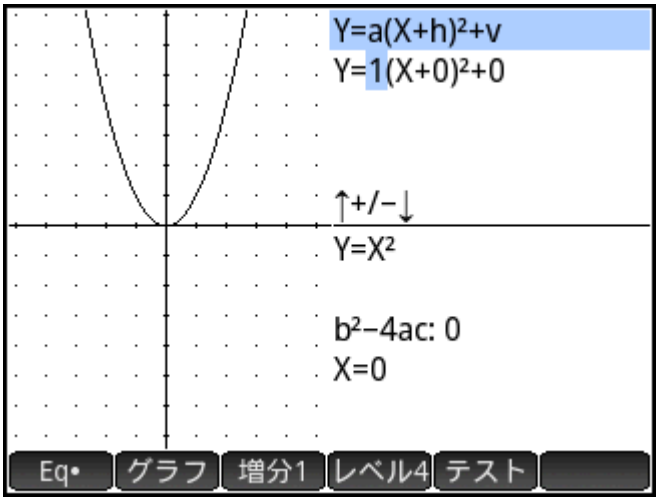
y = *x*² + *v*

y = *a*(*x*+*h*)² + *x*（レベル 4）

レベル1、レベル2などのレベル ボタンをタップして一般的な形式を選択し、必要な形式を表示させます。グラフを操作するために使用できるキーは、レベルごとに異なります。

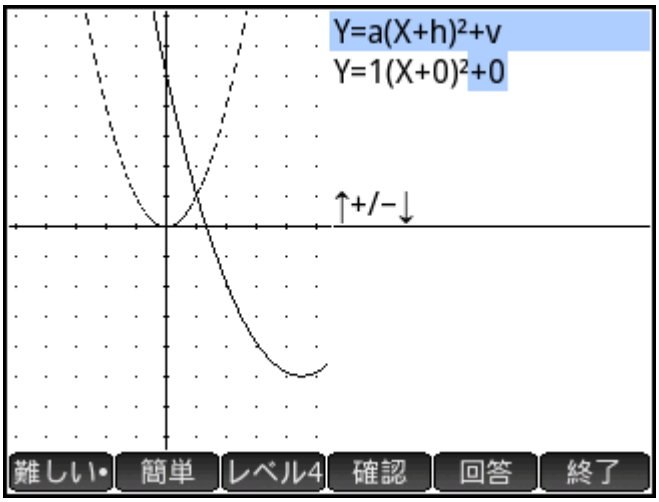
方程式モード

Eq をタップすると、方程式モードに移ります。方程式モードでは、カーソル キーを使用して、方程式内のパラメーター間を移動したり、それらの値を変更したりして、表示されているグラフへの影響を確認します。 ▼ または ▲ を押すと、選択したパラメーターの値が増減します。 ▶ または ◀ を押すと、別のパラメーターが選択されます。 +/− を押すと、符号が変更されます。グラフの形式（またはレベル）は 4 つありますが、方程式の操作に使用できるキーは、選択するレベルによって異なります。



テスト モード

テスト をタップすると、テスト モードに入ります。テスト モードでは、表示されたグラフと方程式を一致させられるかをテストします。テスト モードは、カーソル キーを使用して方程式の各パラメーター値を選択および変更するという点で、方程式モードと似ています。その目標は、表示されたグラフに一致させることです。



このアプリケーションでは、ランダムに選択された 2 次関数のグラフが表示されます。2 次方程式の 4 つの形式から 1 つを選択するには、レベル ボタンをタップします。また、一致させることが比較的簡単なグラフや、比較的難しいグラフを選択することもできます (それぞれ 簡単 または 難しい をタップします)。

ここで、カーソル キーを押し、パラメーターを選択して値を設定します。準備が整ったら 確認 をタップし、表示されたグラフと方程式を正しく一致されられたかを確認します。

正解を表示するには 回答 をタップし、テスト モードを終了するには 終了 をタップします。

三角法エクスプローラーアプリケーション

三角法エクスプローラーアプリケーションでは、*a*、*b*、*c*、および *d* の値の変化に応じたグラフ $y = a \cdot \sin(bx + c) + d$ および $y = a \cdot \cos(bx + c) + d$ の反応を調べることができます。

このアプリケーションで使用できるメニュー項目は次のとおりです。

- Eq または グラフ ：グラフ モードと方程式モードを切り替えます
- SIN または COS ：正弦グラフと余弦グラフを切り替えます
- ラド または DEG ：x の角度の単位としてラジアンと度を切り替えます
- ORIG または EXTR ：グラフの平行移動（ ORIG ）と、その周波数または振幅の変更（ EXTR ）を切り替えます。これらはカーソル キーを使用して変更します
- テスト ：テスト モードに入ります
- π/9 または 20° ：パラメーター値が変化する増分を、π/9、π/6、π/4、または 20°、30°、45°（角度の単位の設定による）の順に切り替えます

アプリケーションを開く

Apps Info を押し、`Trig Explorer`（三角法エクスプローラー）を選択します。

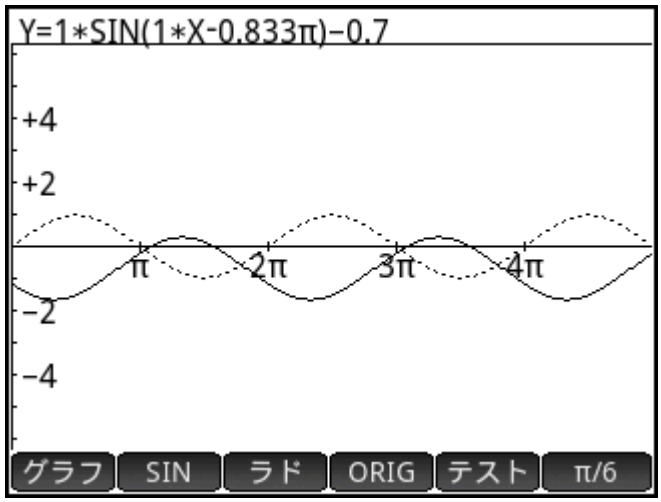
画面上部に方程式が表示され、その下にグラフが表示されます。

COS

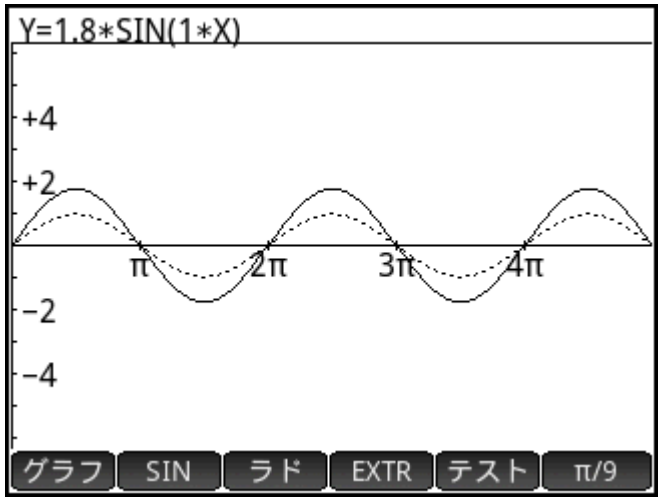
調査する関数のタイプを選択するは、 SIN または COS をタップします。

グラフ モード

このアプリケーションはグラフ モードで開かれます。グラフ モードでは、カーソル キーを押してグラフのコピーを操作します。4 つのキーがすべて使用できます。元のグラフはそのまま残されるため（ただし点線に変換されます）、操作の結果を簡単に確認できます。



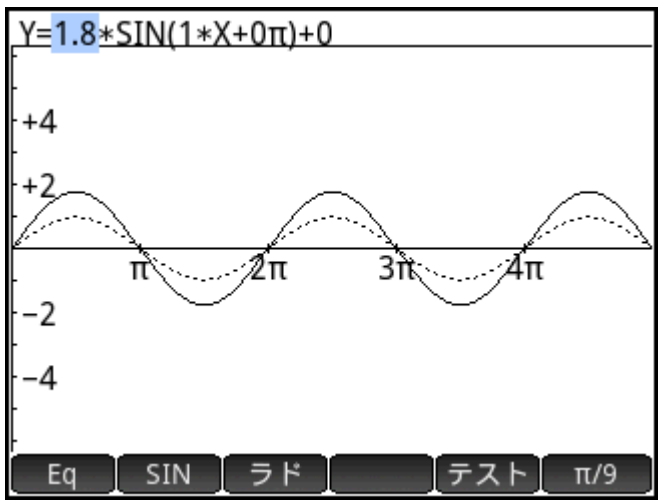
ORIG が選択されているときにカーソル キーを押すと、グラフは単純に水平および垂直方向に平行移動します。 EXTR が選択されているときに ▲ または ▼ を押すと、グラフの振幅が変化します（つまり、グラフが垂直方向に伸長または収縮されます）。また、 ◀ または ▶ を押すと、グラフの周波数が変化します（つまり、グラフが水平方向に伸長または収縮されます）。



メニューの右端にある π/9 ボタンまたは 20° ボタンは、カーソル キーを 1 回押したときにグラフが移動する増分を決定します。初期設定では、増分は π/9 または 20°に設定されます。

方程式モード

グラフ をタップすると、方程式モードに切り替わります。方程式モードでは、カーソル キーを使用して、方程式のパラメーター間を移動したり、それらの値を変更したりします。その後、表示されるグラフへの影響を確認できます。 ▼ または ▲ を押すと、選択したパラメーターの値が増減します。 ▶ または ◀ を押すと、別のパラメーターが選択されます。

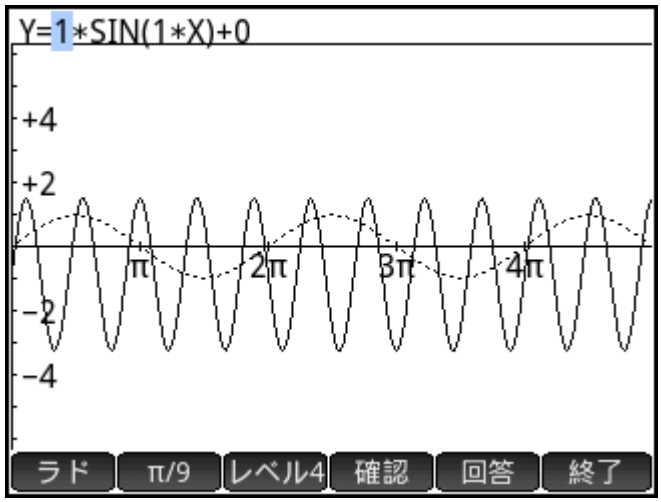


元のグラフ モードに切り替えるには、 Eq をタップします。

テスト モード

テスト をタップすると、テスト モードに入ります。テスト モードでは、表示されたグラフと方程式を一致させられるかをテストします。テスト モードは、カーソル キーを使用して方程式の 1 つ以上のパラメーター値を選択および変更するという点で、方程式モードと似ています。その目標は、表示されたグラフに一致させることです。

このアプリケーションでは、ランダムに選択された正弦関数のグラフが表示されます。5 種類の正弦方程式の中から 1 つを選択するには、 レベル1 、 レベル2 などのレベル ボタンをタップします。



ここで、カーソル キーを押し、各パラメーターを選択して値を設定します。準備が整ったら 確認 をタップし、表示されたグラフと方程式を正しく一致されられたかを確認します。

正解を表示するには 回答 をタップし、テスト モードを終了するには テスト をタップします。

22 関数およびコマンド

多くの数学関数を電卓のキーボードから利用できます。これらの関数については、101 ページの「キーボード関数」を参照してください。その他の関数やコマンドはツールボックス メニュー（ ）に用意されています。ツールボックス メニューには次の 5 つがあります。

[数学]

記号を使用しない数学関数の集合（342 ページの[数学]メニューを参照してください）

[CAS]

記号を使用する数学関数の集合（354 ページの[CAS]メニューを参照してください）

[アプリ]

アプリケーション関数の集合で、ホーム表示、CAS 表示、スプレッドシート アプリケーション、プログラム内など、電卓内の他の場所から呼び出すことができます（376 ページの[アプリ]メニューを参照してください）

ただし、幾何アプリケーション関数は電卓の他の場所から呼び出せるものの、幾何アプリケーション内で使用されるように設計されていることに注意してください。そのため、この章では幾何アプリケーション関数について説明しません。それらについては、「幾何」の章を参照してください。

[ユーザー]

ユーザーが作成した関数（457 ページのユーザー独自の関数の作成を参照）、およびユーザーが作成したプログラムのうち、エクスポートされた関数を含むもの。

[カタログ]

次のすべての関数およびコマンドです。

- **[数学]**メニュー上にあるもの
- **[CAS]**メニュー上にあるもの
- 幾何アプリケーションで使用されるもの
- プログラミングで使用されるもの
- Matrix（行列）エディターで使用されるもの
- List（リスト）エディターで使用されるもの
- およびいくつかの追加の関数およびコマンド

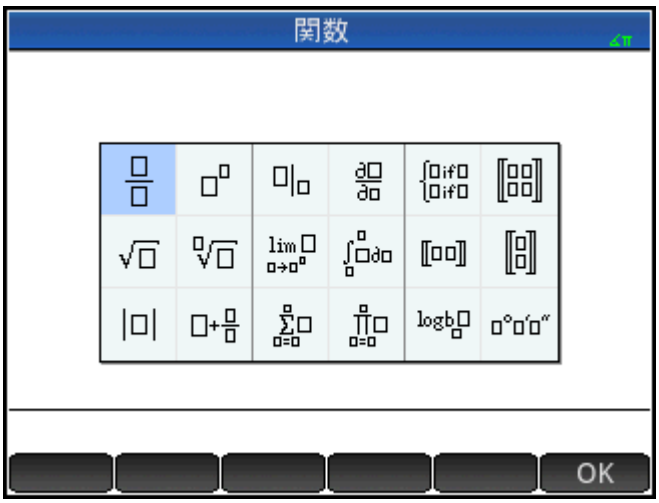
405 ページの[カタログ]メニューを参照してください。

[カタログ]メニューにはすべてのプログラミング コマンドが含まれていますが、[プログラム]エディターのコマンド メニュー（ コマンド ）にも、すべてのプログラミング コマンドがカテゴリ別に含ま

れています。また、一般的なプログラミング構造を含むテンプレート メニュー（ 雛形 ）も含まれています。

注記：一部の関数は数学テンプレート（ を押して表示）から選択できます。

また、ユーザー独自の関数を作成することもできます。457 ページのユーザー独自の関数の作成を参照してください。



メニュー項目の形式の設定

[数学]メニューおよび[CAS]メニューの項目を説明的な名前とコマンド名のどちらで表示させるかを選択できます（[カタログ]メニューの項目は常にコマンド名で表示されます）。

説明的な名前	コマンド名
Factor List（因数リスト）	ifactors
Complex Zeros（複素零点）	cZeros
Groebner Basis（グレブナー基底）	gbasis
Factor by Degree（次数による因数分解）	factor_xn
Find Roots（根を求める）	proot

初期設定のメニュー表示モードでは、数学関数および CAS 関数に対して説明的な名前が表示されます。関数をコマンド名で表示するには、[ホーム設定]画面の 2 ページ目で**[メニュー表示]**オプションの選択を解除します。

この章で使用する略語

関数およびコマンドの構文を説明する際に、次の略語および表記が使用されます。

`Eqn`：方程式

`Expr`：数式

`Fnc`：関数

`Frac`：分数

`Intgr`：整数

`Obj`：ここでは複数の種類のオブジェクトが許可されることを示します

`Poly`：多項式

`RatFrac`：有理分数

`Val`：実数

`Var`：変数

オプションのパラメーターは、`NORMAL_ICDF([μ,σ,]p)`のように角かっこで囲まれます。

読みやすくなるように、コンマを使用してパラメーターを区切っていますが、コンマが必要になるのは複数のパラメーターを区切る場合だけです。したがって、単一パラメーターのコマンドでは、以降に示す構文でそのパラメーターとオプションのパラメーターの間にコンマがあったとしても、パラメーターの後にコンマを挿入する必要はありません。構文 `zeros(Expr,[Var])`はその一例です。コンマが必要になるのは、オプションのパラメーター `Var` を指定する場合だけです。

キーボード関数

最もよく使用される関数は、キーボードから直接利用できます。多くのキーボード関数は、引数として複素数も受け入れます。以降で示すようにキーおよび入力情報を入力して Enter を押すと、式が評価されます。

注記：以降の例では、[Shift]キーを押してアクセスできる関数は、実際に押すキーおよびかっこ付きの関数名で示されます。たとえば、 Shift SIN (ASIN) は、逆正弦計算（ASIN）を行う場合は Shift SIN を押すことを意味しています。

以降の例では、ホーム表示で得られる結果を示しています。CAS 表示の場合は、結果が簡略化された記号形式で表示されます。例：

Shift x^2 320 は、ホーム表示では 17.88854382 を返しますが、CAS 表示では 8*√5 を返します。

+ − × ÷

加算、減算、乗算、除算。複素数、リスト、および行列も受け入れます。

value1 + value2 など

LN

自然対数。複素数も受け入れます。

LN(value)

例：

`LN(1)` は 0 を返します

Shift LN (e^x)

自然指数関数。複素数も受け入れます。

e^{value}

例：

e^5 は 148.413159103 を返します

LOG

常用対数。複素数も受け入れます。

LOG(*value*)

例：

LOG(100) は 2 を返します

Shift LOG (10^x)

10 を底とする指数（真数）。複素数も受け入れます。

ALOG(*value*)

例：

ALOG(3) は 1000 を返します

SIN COS TAN

基本三角関数である正弦、余弦、および正接。

SIN(*value*)

COS(*value*)

TAN(*value*)

例：

TAN(45) は 1 を返します（角度モード）

Shift SIN (ASIN)

逆正弦：$\sin^{-1}x$。出力範囲は – 90° ~ 90°または – π/2 ~ π/2 です。入力および出力は現在の角度形式に依存します。複素数も受け入れます。

ASIN(*value*)

例：

ASIN(1) は 90 を返します（角度モード）

Shift COS (ACOS) （ACOS）

逆余弦：$\cos^{-1}x$。出力範囲は 0° ~ 180°または 0 ~ π です。入力および出力は現在の角度形式に依存します。複素数も受け入れます。余弦の通常の値域-1 ≤ x ≤ 1 の外側にある値の場合、出力は複素数になります。

ACOS(*value*)

例：

ACOS(1) は 0 を返します（角度モード）

Shift TAN (ATAN) （ATAN）

逆正接：$\tan^{-1}(x)$。出力範囲は – 90° ~ 90°または – π/2 ~ π/2 です。入力および出力は現在の角度形式に依存します。複素数も受け入れます。

ATAN(*value*)

例：

ATAN(1) は 45 を返します（角度モード）

x^2

二乗。複素数も受け入れます。

value^2

例：

18^2 は 324 を返します

Shift x^2

平方根。複素数も受け入れます。

√value

例：

√320 は 17.88854382 を返します

x^y

x の y 乗。複素数も受け入れます。

value^{power}

例：

2^8 は 256 を返します

x の n 乗根。

`root√value`

例：

`3√8` は 2 を返します

逆数。

value^{-1}

例：

3^{-1} は.333333333333 を返します

否定。複素数も受け入れます。

-value

例：

`-(1+2*i)` は-1-2*i を返します

絶対値。

`|value|`

`|x+y*i|`

`|matrix|`

複素数の場合、|x+y*i|は $\sqrt{x^2+y^2}$ を返します。行列の場合、|matrix|は行列のフロベニウス ノルムを返します。

例：

`|-1|` は 1 を返します

`|(1,2)|` は 2.2360679775 を返します

代替の構文形式として ABS()および abs()も使用できますが、これらは一部の入力では結果が若干異なります。たとえば、abs(matix)は行列の 12norm を返します。

小数から分数への変換。ホーム表示の場合に、ホーム表示で最後に入力された値の形式を、小数、分数、帯分数の間で切り替えます。履歴内の結果が選択された場合は、選択結果をこれらの形式に順番

に切り替えます。リストおよび行列でも動作します。CAS 表示では、小数表示と分数表示の間でのみ切り替えを行い、新しいエントリとして履歴に追加します。

例：

ホーム表示で、2.4 が履歴の最後のエントリになっているか履歴内で選択されている場合、 a b/c を押すと 12/5 と表示されます。もう一度 a b/c を押すと 2+2/5 と表示され、さらにもう一度 a b/c を押すと 2.4 に戻ります。

Shift a b/c

10 進法から 60 進法への変換。ホーム表示の場合に、ホーム ビューで最後に入力された値の形式を、10 進数と 60 進数の間で切り替えます。履歴内の結果が選択された場合は、選択結果をこれらの形式に順番に切り替えます。リストおよび行列でも動作します。CAS 表示では、新しいエントリとして履歴に追加します。

例：

ホーム表示で、2.4 が履歴の最後のエントリになっているか履歴内で選択されている場合、 Shift a b/c を押すと 2°24′0″ と表示されます。もう一度 Shift a b/c を押すと、2.4 に戻ります。

EEX

EEX キーは指数表記の数を入力するために使用します。

HP Prime 電卓では、指数表記の数は E 文字（ EEX キーに対応）で区切られた 2 つの部分で表されます。1 つ目の部分は、一般に仮数部と呼ばれる実数です。2 つ目の部分は、一般に指数と呼ばれる整数です。この表記で表される数は、仮数部*10^指数です。

たとえば、 3 EEX 5 を押すとコマンド ラインに 3E5 と表示されます。これは、300,000 という数を返します。

例：

3E2 は 300 を返します。

Shift 9

よく使用される数学記号およびギリシャ文字のメニューを開くには、 Shift 9 または Shift Vars のどちらかを押します。

Shift 6

一般的なブール演算子のメニューを開くには、Shift 6を押します。これらの演算子は、カタログから探すこともできます。

Shift 2

虚数単位 i。

虚数単位 i を挿入します。

Shift 3

定数 π。

超越定数 π を挿入します。

[数学]メニュー

Mem を押すと、ツールボックスのメニューが開きます（その 1 つが[数学]メニュー）。[数学]メニューで利用できる関数およびコマンドは、メニュー上の分類に従って一覧表示されています。

[数]

[天井]

value 以上の最小の整数。

```
CEILING(value)
```

例：

`CEILING(3.2)` は 4 を返します

`CEILING(-3.2)` は-3 を返します

[床]

value 以下の最大の整数。

```
FLOOR(value)
```

例：

`FLOOR(3.2)` は 3 を返します

`FLOOR(-3.2)` は-4 を返します

[IP]

整数部。

```
IP(value)
```

例：

IP(23.2)は 23 を返します

[FP]

小数部。

FP(value)

例：

FP (23.2)は.2 を返します

[丸め]

places に指定した小数位までに value を丸めます。複素数も受け入れます。

ROUND(value,places)

places に負の整数を指定することで、 ROUND で有効桁数に丸めることもできます（以下の 2 番目の例を参照してください）。

例：

ROUND(7.8676,2)は 7.87 を返します

ROUND(0.0036757-3)は 0.00368 を返します

[切り捨て]

小数点以下 places 桁に value を切り捨てます。複素数も受け入れます。

TRUNCATE(value,places)

例：

TRUNCATE(2.3678,2)は 2.36 を返します

TRUNCATE(0.0036757,-3)は 0.00367 を返します

[仮数部]

浮動小数点数 value の仮数部（つまり有効桁数）。

MANT(value)

例：

MANT(21.2E34)は 2.12 を返します

[指数]

value の指数。つまり、value を生成する 10 のべき乗の整数部分です。

XPON(value)

例：

XPON(123456)は 5 を返します（105.0915...が 123456 に等しくなるため）

[算術計算]

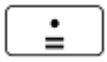
[最大]

最大値。2 つの値のうち大きいもの。

```
MAX(value1,value2)
```

例：

```
MAX(8/3,11/4)
```
は 2.75 を返します

ホーム表示では、整数以外の結果は小数として与えられることに注意してください。結果を分数で表示するには、  を押します。このキーを押すと、小数、分数、帯分数の各表示に順番に切り替わります。または、 を押すこともできます。この場合、数式処理システムが開きます。ホーム表示に戻って計算を続けるには、 を押します。

[最小]

最小値。与えられた値のうち最小のもの、またはリストの最小値を返します。

```
MIN(value1,value2)
```

例：

```
MIN(210,25)
```
は 25 を返します

[係数]

法。value1/value2 の余り。

```
value1 MOD value2
```

例：

```
74 MOD 5
```
は 4 を返します

[根を求める]

関数 root-finder（解くアプリケーションに類似）。指定の変数で式が最も 0 に近く評価される値を求めます。guess は初期予測値として使用します。

```
FNROOT(expression,variable,guess)
```

例：

```
FNROOT((A*9.8/600)-1,A,1)
```
は 61.2244897959 を返します

[パーセント]

y の x パーセント。つまり、x/100*y。

```
%(x,y)
```

例：

```
%(20,50)
```
は 10 を返します

[算術計算] : [複素数]

[偏角]

偏角。複素数によって定義される角度を求めます。入力および出力では、ホーム表示で設定された現在の角度形式が使用されます。

`ARG(x+y*i)`

例：

`ARG(3+3*i)` は 45（角度モード）を返します

[共役]

共役複素数。共役とは複素数の虚部を否定（符号反転）したものです。

`CONJ(x+y*i)`

例：

`CONJ(3+4*i)` は `(3-4*i)` を返します

[実部]

複素数(x+y*i)の実部 x。

`RE(x+y*i)`

例：

`RE(3+4*i)` は 3 を返します

[虚部]

複素数(x+y*i)の虚部 y。

`IM(x+y*i)`

例：

`IM(3+4*i)` は 4 を返します

[単位ベクトル]

value の符号。正の場合、結果は 1 になり、負の場合、 – 1 になります。ゼロの場合、結果はゼロです。複素数の場合、これは数の方向の単位ベクトルになります。

`SIGN(value)`

`SIGN((x,y))`

例：

`SIGN(POLYEVAL([1,2,-25,-26,2],-2))` は – 1 を返します

`SIGN((3,4))` は(.6+.8i)を返します

[算術計算]：[指数]

[ALOG]

真数（指数）。

```
ALOG(value)
```

[EXPM1]

指数マイナス 1：e^x-1。

```
EXPM1(value)
```

[LNP1]

自然対数プラス 1：ln(x+1)。

```
LNP1(value)
```

[三角関数]

三角関数は複素数も引数として指定できます。SIN、COS、TAN、ASIN、ACOS、および ATAN については、337 ページのキーボード関数を参照してください。

[CSC]

余割：1/sin(x)。

```
CSC(value)
```

[ACSC]

逆余割：csc^{-1}(x)。

```
ACSC(value)
```

[SEC]

正割：1/cos(x)。

```
SEC(value)
```

[ASEC]

逆正割：sec^{-1}(x)。

```
ASEC(value)
```

[COT]

余接：cos(x)/sin(x)

```
COT(value)
```

[ACOT]

逆余接：cot^{-1}(x)。

```
ACOT(value)
```

[双曲線]

双曲線三角関数は複素数も引数として指定できます。

[SINH]

双曲線正弦。

```
SINH(value)
```

[ASINH]

双曲線正弦の逆関数：$\sinh^{-1}x$。

```
ASINH(value)
```

[COSH]

双曲線余弦

```
COSH(value)
```

[ACOSH]

双曲線余弦の逆関数：$\cosh^{-1}x$。

```
ACOSH(value)
```

[TANH]

双曲線正接。

```
TANH(value)
```

[ATANH]

双曲線正接の逆関数：$\tanh^{-1}x$。

```
ATANH(value)
```

[確率]

[階乗]

正の整数の階乗。整数以外の場合、$x! = \Gamma(x + 1)$となります。これはガンマ関数を計算します。

```
value!
```

例：

```
5!
```
は 120 を返します

[組み合わせ]

n 個の中から同時に r 個を取った場合の組み合わせの数（順番は関係なし）。

```
COMB(n,r)
```

例：5 個から同時に 2 個を取るときの組み合わせの数を求めるとします。

`COMB(5,2)` は 10 を返します

[順列]

n 個の中から同時に r 個を取った場合の順列の数（順番は関係あり）：n!/(n – r)!.

`PERM (n,r)`

例：5 個から同時に 2 個を取るときの順列の数を求めるとします。

`PERM(5,2)` は 20 を返します

[確率]：[乱数]

[番号]

乱数。引数なしの場合、この関数は 0 ~ 1 の間の乱数を返します。引数が 1 つ（a）の場合、0 ~ a の間の乱数を 1 つ返します。引数が 2 つ（a および b）の場合、a ~ b の間の乱数を 1 つ返します。引数が 3 つ（n、a、および b）の場合、a ~ b の間の乱数を n 個返します。

`RANDOM`

`RANDOM(a)`

`RANDOM(a,b)`

`RANDOM(n,a,b)`

[整数]

乱数（整数）。引数がない場合、この関数は 0 または 1 をランダムに返します。引数が 1 つ（a）の場合、0 ~ a の間の整数乱数を 1 つ返します。引数が 2 つ（a および b）の場合、a ~ b の間の整数乱数を 1 つ返します。引数が 3 つ（n、a、および b）の場合、a ~ b の間の整数乱数を n 個返します。

`RANDINT`

`RANDINT(a)`

`RANDINT(a,b)`

`RANDINT(n,a,b)`

[正規]

正規乱数。正規分布から乱数を生成します。

`RANDNORM(μ,σ)`

例：

`RANDNORM(0,1)` は、標準正規分布から乱数を返します。

[シード]

乱数関数が処理するシード値を設定します。2 つ以上の電卓で同じシード値を指定すれば、乱数関数の実行時にそれぞれの電卓で同じ乱数が生成されるようになります。

RANDSEED(value)

[確率]：[密度]

[正規]

正規確率密度関数。正規分布の平均値 μ および標準偏差 σ を指定し、値 x の確率密度を計算します。指定された引数が 1 つだけの場合、その引数は x と見なされ、μ=0 および σ=1 と仮定されます。

NORMALD([μ,σ,]x)

例：

NORMALD(0.5) と NORMALD(0,1,0.5) はどちらも 0.352065326764 を返します。

[T]

スチューデントの t 確率密度関数。自由度が n の場合の、スチューデントの t 分布の x での確率密度を計算します。

STUDENT(n,x)

例：

STUDENT(3,5.2) は 0.00366574413491 を返します。

[χ^2]

χ^2 確率密度関数。自由度が n の場合の、x^2 分布の x での確率密度を計算します。

CHISQUARE(n,x)

例：

CHISQUARE(2,3.2) は 0.100948258997 を返します。

[F]

フィッシャー（またはフィッシャー-スネデカー）確率密度関数。分子および分母の自由度がそれぞれ n および d の場合の、値 x での確率密度を計算します。

FISHER(n,d,x)

例：

FISHER(5,5,2) は 0.158080231095 を返します。

[二項]

二項確率密度関数。成功確率が p のとき、n 回の試行のうち成功が k 回の確率を計算します。3 番目の引数がない場合は、Comb(n,k)を返します。n および k は整数で、k≤n であることに注意してください。

BINOMIAL(n,k,p)

例：公正な硬貨を 20 回投げて表が出る回数が 6 回のみになる確率を求めるとします。

BINOMIAL(20,0.5,6) は 0.0369644165039 を返します。

[幾何]

幾何確率密度関数。確率 p での x における幾何分布の確率密度を計算します。

```
GEOMETRIC(p, x)
```

例：

```
GEOMETRIC(0.3, 4)
```
は 0.1029 を返します。

[ポワソン]

ポワソン確率質量関数。過去のある時間間隔での事象の平均発生頻度 μ が与えられた場合に、将来同じ時間間隔でその事象が発生する回数が k になる確率を計算します。この関数で、k は負ではない整数、μ は実数です。

```
POISSON(μ,k)
```

例:1 日あたり平均 20 通の電子メールを受け取っているとします。明日 1 日で 15 通受け取る確率は、次のようになります。

```
POISSON(20,15)
```
は 0.0516488535318 を返します。

[確率] : [累積]

[正規]

累積正規分布関数。正規分布の平均 μ および標準偏差 σ を指定し、値 x の正規確率密度関数の下側裾確率を返します。指定された引数が 1 つだけの場合、その引数は x と見なされ、μ=0 および σ=1 と仮定されます。

```
NORMALD_CDF([μ,σ,]x)
```

例：

```
NORMALD_CDF(0,1,2)
```
は 0.977249868052 を返します。

[T]

累積スチューデントの t 分布関数。自由度が n の場合の、x でのスチューデントの t 確率密度関数の下側裾確率を返します。

```
STUDENT_CDF(n,x)
```

例：

```
STUDENT_CDF(3,-3.2)
```
は 0.0246659214814 を返します。

[X^2]

累積 X^2 分布関数。自由度が n の場合の、値 X での X^2 確率密度関数の下側裾確率を返します。

```
CHISQUARE_CDF(n,k)
```

例：

```
CHISQUARE_CDF(2, 6.3)
```
は 0.957147873133 を返します。

[F]

フィッシャーの累積分布関数。分子および分母の自由度がそれぞれ n および d の場合の、値 x でのフィッシャー確率密度関数の下側裾確率を返します。

```
FISHER_CDF(n,d,x)
```

例：

`FISHER_CDF(5,5,2)` は 0.76748868087 を返します。

[二項]

累積二項分布関数。各試行の成功確率が p のとき、n 回の試行のうち成功が k 回以下の確率を返します。n および k は整数で、k≤n であることに注意してください。

```
BINOMIAL_CDF(n,p,k)
```

例：公正な硬貨を 20 回投げて表が出る回数が 0、1、2、3、4、5、または 6 回になる確率を求めるとします。

`BINOMIAL_CDF(20,0.5,6)` は 0.05765914917 を返します。

[幾何]

累積幾何分布関数。引数 2 つ（p および x）の場合、確率 p での、値 x についての幾何確率密度関数の下側裾確率を返します。引数 3 つ（p、x_1、および x_2）の場合、確率 p での、幾何確率密度関数の下側の x_1 と x_2 の間の面積を返します。

```
GEOMETRIC_CDF(p, x)
```

```
GEOMETRIC_CDF(p, x1, x2)
```

例：

`GEOMETRIC_CDF(0.3, 4)` は 0.7599 を返します。

`GEOMETRIC_CDF(0.5, 1, 3)` は 0.375 を返します。

[ポワソン]

累積ポワソン分布関数。予想される発生頻度が与えられた場合に、与えられた時間間隔での事象の発生頻度が x 以下になる確率を返します。

```
POISSON_CDF( ,x)
```

例：

`POISSON_CDF(4,2)` は 0.238103305554 を返します。

[確率]：[逆]

[正規]

累積正規分布関数の逆関数。正規分布の平均値 μ および標準偏差 σ を指定し、下側裾確率 p に関連付けられた累積正規分布値を返します。指定された引数が 1 つだけの場合、その引数は p と見なされ、μ=0 および σ=1 と仮定されます。

```
NORMALD_ICDF([μ,σ,]p)
```

例：

NORMALD_ICDF(0,1,0.841344746069) は 1 を返します。

[T]

累積スチューデントの t 分布関数の逆関数。自由度が n の場合の、x でのスチューデントの t 下側裾確率が p になるような値 x を返します。

STUDENT_ICDF(n,p)

例：

STUDENT_ICDF(3,0.0246659214814) は –3.2 を返します。

[χ^2]

累積 χ^2 分布関数の逆関数。自由度が n の場合の、χ での χ^2 の下側裾確率が p になるような値 χ を返します。

CHISQUARE_ICDF(n,p)

例：

CHISQUARE_ICDF(2, 0.957147873133) は 6.3 を返します。

[F]

フィッシャーの累積分布関数の逆関数。分子および分母の自由度がそれぞれ n および d の場合の、x でのフィッシャー下側裾確率が p になるような値 x を返します。

FISHER_ICDF(n,d,p)

例：

FISHER_ICDF(5,5,0.76748868087) は 2 を返します。

[二項]

累積二項分布関数の逆関数。確率が p の試行を n 回行い、k 回以下成功する確率が q となるような成功回数 k を返します。

BINOMIAL_ICDF(n,p,q)

例：

BINOMIAL_ICDF(20,0.5,0.6) は 11 を返します。

[幾何]

累積幾何分布関数の逆関数。確率が p のとき、下側裾確率の値が k になるような x 値を返します。

GEOMETRIC_ICDF(p, k)

例：

GEOMETRIC_ICDF(0.3, 0.95) は 9 を返します。

[ポワソン]

累積ポワソン分布関数の逆関数。ある時間間隔での事象の予想（または平均）発生頻度 μ が与えられた場合に、同じ時間間隔でその事象が発生する回数が x 以下になる確率が p になるような値 x を返します。

```
POISSON_ICDF( ,p)
```

例：

```
POISSON_ICDF(4,0.238103305554)
```
は 3 を返します。

[リスト]

これらの関数の処理対象は、リスト内のデータになります。詳しくは、『HP Prime グラフ電卓ユーザー ガイド』の「リスト」の章を参照してください。

[行列]

これらの関数の処理対象は、行列変数に格納された行列データになります。詳しくは、『HP Prime グラフ電卓ユーザー ガイド』の「行列」の章を参照してください。

[特殊]

[ベータ]

2 つの数 a および b についてのベータ関数（B）の値を返します。

```
Beta(a,b)
```

[ガンマ]

数 a についてのガンマ関数（Γ）の値を返します。

```
Gamma(a)
```

[Psi]

x=a でのディガンマ関数の n 次導関数の値を返します。ディガンマ関数は、ln(Γ(x))の 1 次導関数です。

```
Psi(a,n)
```

[Zeta]

実数 x についてのゼータ関数(Z)の値を返します。

```
Zeta(x)
```

[erf]

誤差関数の x=a での浮動小数点値を返します。

```
erf(a)
```

[erfc]

相補誤差関数の x=a での値を返します。

```
erfc(a)
```

[Ei]

式の指数積分を返します。

```
Ei(Expr)
```

[Si]

式の正弦積分を返します。

```
Si(Expr)
```

[Ci]

式の余弦積分を返します。

```
Ci(Expr)
```

[CAS]メニュー

を押すと、ツールボックスのメニューが開きます（その 1 つが[CAS]メニュー）。[CAS]メニューの関数は、最も一般的に使用される関数です。使用可能な関数は他にもたくさんあります。405 ページの[カタログ]メニューを参照してください。Geometry（幾何）関数は[アプリ]メニューに表示されることに注意してください。

CAS コマンドの結果は CAS 設定によって異なる場合があります。この章の例では、他に記載がない限り、CAS の初期設定を使用します。

[代数]

[簡略化]

簡略化された式を返します。

```
simplify(Expr)
```

例：

simplify(4*atan(1/5)-atan(1/239))は(1/4)*pi を返します

[整理]

多項式（または多項式のリスト）の同類項を整理します。CAS 設定によっては、結果を因数分解します。

collect(Poly)または collect({Poly1, Poly2,..., Polyn})

例：

collect(x+2*x+1-4)は 3*x-3 を返します

collect(x^2-9*x+5*x+3+1)は(x-2)^2 を返します

[展開]

式を展開して返します。

expand(Expr)

例：

expand((x+y)*(z+1))は y*z+x*z+y+x を返します

[因数]

因数分解された多項式を返します。

factor(Poly)

例：

factor(x^4-1)は(x-1)*(x+1)*(x^2+1)を返します

[置換]

値を使用して式の変数を置換します。

構文：subst(Expr,Var=value)

例：

subst(x/(4-x^2),x=3)は-3/5 を返します

[部分分数]

分数について部分分数分解を実行します。

partfrac(RatFrac または Opt)

例：

partfrac(x/(4-x^2))は(-1/2)/(x-2)-(1/2)/((x+2)を返します

[代数]：[抽出]

[分子]

簡略化された分子。整数 a および b について、分数 a/b を簡略化した後の分子を返します。

numer(a/b)

例：

numer(10/12) は 5 を返します

[分母]

簡略化された分母。整数 a および b について、分数 a/b を簡略化した後の分母を返します。

denom(a/b)

例：

denom(10/12) は 6 を返します

[左辺]

方程式の左辺か、区間の左端を返します。

left(Expr1=Expr2) または left(Real1..Real2)

例：

left(x^2-1=2*x+3) は x^2-1 を返します

[右辺]

方程式の右辺か、区間の右端を返します。

right(Expr1=Expr2) または right(Real1..Real2)

例：

right(x^2-1=2*x+3) は 2*x+3 を返します

[微積分]

[微分]

1 つの式を引数として指定した場合は、x についての式の導関数を返します。1 つの式および 1 つの変数を引数として指定した場合は、その変数についての式の導関数または偏導関数を返します。1 つの式および複数の変数を引数として指定した場合は、2 番目の引数の変数についての式の導関数を返します。これらの引数の後に$k（k は整数）を付け、その変数について式を導き出す必要がある回数を示すことができます。たとえば、diff(exp(x*y),x$3,y$2,z)は diff(exp(x*y),x,x,x,y,y,z)と同じです。

diff(Expr,[var])

または

diff(Expr,var1$k1,var2$k2,...)

例：

diff(x^3-x) は 3*x^2-1 を返します

[積分]

式の積分を返します。1 つの式を引数として指定して、x についての不定積分を返します。オプションの 2 番目、3 番目、および 4 番目の引数を指定すると、積分の変数および定積分の境界を指定できます。

```
int(Expr,[Var(x)],[Real(a)],[Real(b)])
```

例：

`int(1/x)` は `ln(abs(x))` を返します

[極限]

変数が極限点または+/ – 無限大に近づくとき、式の極限を返します。オプションの 4 番目の引数を指定すると、下方向または上方向からの制限であるか、両方向の制限であるかを指定できます（下方向からの極限の場合は – 1、上方向からの極限の場合は+1、両方向の極限の場合は 0）。4 番目の引数を指定しない場合、返される極限は両方向です。limit 関数から±∞が返されることがありますが、これは複素無限大（引数が不明な複素平面内の無限数）を示します。極限の文脈では、複素無限大は通常、極限が未定義であることを意味するものと解釈されます。

```
limit(Expr,Var,Val,[Dir(1, 0, -1)])
```

例：

`limit((n*tan(x)-tan(n*x))/(sin(n*x)-n*sin(x)),x,0)` は 2 を返します

たとえば、lim(1/x, x, 0)は±∞を返します。これは数学的に正しく、この場合には極限が未定義であることを示しています。

[数列]

与えられた変数の値の近傍での式の級数展開を返します。オプションの 3 番目および 4 番目の引数を指定すると、級数展開の次数および方向を指定できます。次数が指定されない場合、返される級数は 5 次です。方向が指定されない場合、級数は両側です。

```
series(Expr,Equal(var=limit_point),[Order],[Dir(1,0,-1)])
```

例：

`series((x^4+x+2)/(x^2+1),x=0,5)` は 2+x-2x^2-x^3+3x^4+x^5+x^6*order_size(x)を返します。

[総和]

変数 Var について、Real1 から Real2 までの Expr の離散的な和を返します。テンプレート メニューの総和テンプレートも使用できます。最初の 2 つの引数のみの場合、変数に関する式の離散的な不定積分を返します。

```
sum(Expr,Var,Real1, Real2,[Step])
```

例：

`sum(n^2,n,1,5)` は 55 を返します

[微積分] : [微分]

[回転]

ベクトル場の回転を返します。Curl([A B C], [x y z])は[dC/dy-dB/dz dA/dz-dC/dx dB/dx-dA/dy]と定義されます。

```
curl([Expr1, Expr2, …, ExprN], [Var1, Var2, …, VarN])
```

例：

```
curl([2*x*y,x*z,y*z],[x,y,z])
```
は[z-x,0,z- 2*x]を返します

[発散]

以下で定義されるベクトル場の発散を返します。

divergence([A,B,C],[x,y,z])=dA/dx+dB/dy+dC/dz

```
divergence([Expr1, Expr2, …, ExprN], [Var1, Var2, …, VarN])
```

例：

```
divergence([x^2+y,x+z+y,z^3+x^2],[x,y,z])
```
は 2*x+3*z^2+1 を返します

[勾配]

式の勾配を返します。2 番目の引数として変数のリストが指定された場合、偏導関数のベクトルを返します。

```
grad(Expr,LstVar)
```

例：

```
grad(2*x^2*y-x*z^3,[x,y,z])
```
は[2*2*x*y-z^3,2*x^2,-x*3*z^2]を返します

[ヘッセ]

式のヘッセ行列を返します。

```
hessian(Expr,LstVar)
```

例：

```
hessian(2*x^2*y-x*z,[x,y,z])
```
は[[4*y,4*x,-1],[2*2*x,0,0],[-1,0,0]]を返します

[微積分] : [積分]

[部分積分 u]

式 f(x)=u(x)*v'(x)の部分積分を実行します。f(x)を最初の引数、u(x)（または 0）を 2 番目の引数とします。特に、最初の要素が u(x)*v(x)、2 番目の要素が v(x)*u'(x)のベクトルを返します。オプションの 3 番目、4 番目、および 5 番目の引数については、積分の変数および積分の区域を指定できます。積分の変数が指定されない場合、変数は x と見なされます。

```
ibpu(f(Var), u(Var), [Var], [Real1], [Real2])
```

例：

```
ibpu(x*ln(x), x)
```
は[x*(x*ln(x) – x*ln(x)+x]を返します

[部分積分 v]

式 f(x)=u(x)*v'(x)の部分積分を実行します。f(x)を最初の引数、v(x)（または 0）を 2 番目の引数とします。特に、最初の要素が u(x)*v(x)、2 番目の要素が v(x)*u'(x)のベクトルを返します。オプションの 3 番目、4 番目、および 5 番目の引数については、積分の変数および積分の区域を指定できます。積分の変数が指定されない場合、変数は x と見なされます。

`ibpdv(f(Var), v(Var), [Var], [Real1], [Real2])`

例：

`ibpdv(ln(x),x)` は x*ln(x)-x を返します

[F(b) – F(a)]

F(b) – F(a)を返します。

`preval(Expr(F(var)),Real(a),Real(b),[Var])`

例：

`preval(x^2-2,2,3)` は 5 を返します

[微積分] : [極限]

[リーマン和]

合計を、[0,1]で定義される連続関数に関連付けられたリーマン和と見なす場合、（n=+∞の近傍で）var2 について var2=1 から var2=var1 までの Expr の合計の相当量を返します。

`sum_riemann(Expr, [Var1 Var2])`

例：

`sum_riemann(1/(n+k),[n,k])` は ln(2)を返します

[テイラー]

ある点または無限での式のテイラー級数展開を返します（初期設定では、x=0 の周りで相対次数=5）。

`taylor(Expr,[Var=Value],[Order])`

例：

`taylor(sin(x)/x,x=0)` は 1-(1/6)*x^2+(1/120)*x^4+x^6*order_size(x)を返します

[商のテイラー]

2 つの多項式の商についての n 次のテイラー多項式を返します。

`divpc(Poly1,Poly2,Integer)`

例：

`divpc(x^4+x+2,x^2+1,5)` は、5 次多項式 x^5+3*x^4-x^3-2*x^2+x+2 を返します

[微積分] : [変換]

[ラプラス]

式のラプラス変換を返します。

`laplace(Expr,[Var],[LapVar])`

例：

`laplace(exp(x)*sin(x))` は 1/(x^2-2*x+2)を返します

[ラプラス逆]

Returns the inverse Laplace transform of an expression.

`ilaplace(Expr,[Var],[IlapVar])`

例：

`ilaplace(1/(x^2+1)^2)` は((-x)*cos(x))/2+sin(x)/2 を返します

[FFT]

1 つの引数（ベクトル）を指定すると、R の離散フーリエ変換を返します。

`fft(Vect)`

2 つの整数引数 a および p を追加で指定すると、1 の原始 n 乗根（n=size(vector)）を使用して、体 Z/pZ で離散フーリエ変換を返します。

`fft((Vector, a, p)`

例：

`fft([1,2,3,4,0,0,0,0])` は[10.0,-0.414213562373-7.24264068712*(i),-2.0+2.0*i, 2.41421356237-1.24264068712*i,-2.0,2.41421356237+1.24264068712*i,-2.0-2.0*i]を返します

[逆 FFT]

離散フーリエ変換の逆変換を返します。

`ifft(Vector)`

例：

`ifft([100.0,-52.2842712475+6*i,-8.0*i,4.28427124746-6*i, 4.0,4.28427124746+6*i,8*i,-52.2842712475-6*i])` は [0.99999999999,3.99999999999,10.0,20.0,25.0,24.0,16.0,-6.39843733552e-12]を返します

[解く]

[解く]

多項方程式または多項方程式のセットに対する解のリスト（実数および複素数）を返します。

`solve(Eq,[Var])` または `solve({Eq1, Eq2,…}, [Var])` または `solve(Eq, Var=Guess)` または `solve(Eq, Var=Val1…Val2)`

解の近似値がわかっている場合に最適な結果を得るには、推定値を入力するか、解を求める区間を定義します。

ソルバーに初期値を入力するには、構文 `Var=Guess` を使用します。

閉区間[Val1, Val2]を定義するには、構文 `Var=Val1...Val2` を使用します。

例：

`solve(x^2-3=1)` は{-2,2}を返します

`solve({x^2-3=1, x+2=0},x)` は{-2}を返します

`solve(x^2-(LN(x)+5)=0, x=2)` および `solve(x^2-(LN(x)+5)=0, x=2...3)` は、どちらも 2.42617293082 を返します

[ゼロ]

1 つの式が引数として指定された場合、式の実零点を返します（つまり、式をゼロに等しいと置いた場合の解）。

式のリストが引数として指定された場合、各式をゼロに等しいと置くことによって形成される系の実数解が各行に格納された行列を返します。

`zeros(Expr,[Var])` または `zeros({Expr1, Expr2,…},[{Var1, Var2,…}])`

例：

`zeros(x^2-4)` は[-2 2]を返します

[複素数を解く]

多項方程式または多項方程式のセットに対する複素数解のリストを返します。

`cSolve(Eq,[Var])` または `cSolve({Eq1, Eq2,…}, [Var])`

例：

`cSolve(x^4-1=0, x)` は{1 -1 -i i}を返します

[複素零点]

1 つの式が引数として指定された場合、式の複素零点を含むベクトルを返します（つまり、式をゼロに等しいと置いた場合の解）。

式のリストが引数として指定された場合、各式をゼロに等しいと置くことによって形成される系の複素数解が各行に格納された行列を返します。

`cZeros(Expr,[Var]` または `cZeros({Expr1, Expr2,…},[{Var1, Var2,…}])`

例：

`cZeros(x^4-1)` は[1 -1 -i i]を返します

[数値的に解く]

方程式または方程式系の数値解を返します。

オプションで、3 番目の引数を使用して、解の推測を指定したり、解が存在すると予想される区間を指定したりできます。

オプションで、4 番目の引数を使用して、ソルバーによって使用される反復アルゴリズムの名前を指定することができます。

1 つの変数について解く場合、反復アルゴリズムのオプションは bisection_solver、newton_solver、および newtonj_solver です。2 つの変数について解く場合のオプションは newton_solver だけです。

`fSolve(Eq,Var)` または `fSolve(Expr, Var=Guess)`

例：

`fSolve(cos(x)=x,x,-1..1)` は[0.739085133215]を返します

`fSolve([x²+y-2,x+y²-2],[x,y],[0,0])` は[1.,1.]を返します

[微分方程式]

微分方程式の解を返します。

`deSolve(Eq,[TimeVar],Var)`

例：

`desolve(y''+y=0,y)` は G_0*cos(x)+G_1*sin(x)を返します

[常微分方程式を解く]

常微分方程式ソルバー。VectrVar で宣言されている変数、および VectrInit で宣言されているこれらの変数の初期条件を使用して、Expr で指定される常微分方程式を解きます。たとえば、odesolve(f(t,y),[t,y],[t0,y0],t1)は、変数 t および変数 y について、初期条件 t=t0 および y=y0 での y'=f(t,y)の近似解を返します。

`odesolve(Expr,VectVar,VectInitCond,FinalVal,[tstep=Val,curve])`

例：

`odesolve(sin(t*y),[t,y],[0,1],2)` は[1.82241255674]を返します

[線形系]

1 次方程式のベクトルおよび対応する変数のベクトルを与えると、1 次方程式系の解を返します。

`linsolve([LinEq1, LinEq2,…], [Var1, Var2,…])`

例：

`linsolve([x+y+z=1,x-y=2,2*x-z=3],[x,y,z])` は[3/2,-1/2,0]を返します

[書き換え]

[lncollect]

対数を 1 つにまとめることで式を書き換えます。ln(a)+n*ln(b) = ln(a*b^n)を適用します（n は整数）。

`lncollect(Expr)`

例：

`lncollect(ln(x)+2*ln(y))` は ln(x*y^2)を返します

[powexpand]

式に含まれる和または積の形式のべき乗を、べき乗の積に書き換えます。a^(b+c)=(a^b)*(a^c)を適用します。

`powexpand(Expr)`

例：

`powexpand(2^(x+y))` は(2^x)*(2^y)を返します

[テイラー展開]

超越式を展開します。

`texpand(Expr)`

例：

`texpand(sin(2*x)+exp(x+y))` は exp(x)*exp(y)+ 2*cos(x)*sin(x))を返します

[書き換え]：[指数関数と自然対数]

[$e^{y*\ln x} \rightarrow x^y$]

$e^{n*\ln(x)}$形式の式を x のべき乗に書き換えたものを返します。$e^{n*\ln(x)}=x^n$ を適用します。

`exp2pow(Expr)`

例：

`exp2pow(exp(3*ln(x)))` は x^3 を返します

[$x^y \rightarrow e^{y*\ln x}$]

べき乗を指数で書き換えた式を返します。基本的に、`exp2pow` の逆変換です。

`pow2exp(Expr)`

例：

`pow2exp(a^b)` は exp(b*ln(a))を返します

[exp2trig]

式の複素数指数を正弦および余弦を使って書き換えたものを返します。

`exp2trig(Expr)`

例：

`exp2trig(exp(i*x))` は cos(x)+(i)*sin(x)を返します

[expexpand]

式の指数を展開形式にして返します。

`expexpand(Expr)`

例：

`expexpand(exp(3*x))` は exp(x)^3 を返します

[書き換え] : [正弦]

[asinx→acosx]

asin(x)を π/2 – acos(x)で書き換えた式を返します。

`asin2acos(Expr)`

例 :

`asin2acos(acos(x)+asin(x))` は π/2 を返します

[asinx→atanx]

asin(x)を $\operatorname{atan}\left(\frac{x}{\sqrt{1-x^2}}\right)$ で書き換えた式を返します。

`asin2atan(Expr)`

例 :

`asin2atan(2*asin(x))` は $2\cdot\operatorname{atan}\left(\frac{x}{\sqrt{1-x^2}}\right)$ を返します

[sinx→cosx*tanx]

sin(x)で cos(x)*tan(x)に書き換えた式を返します。

`sin2costan(Expr)`

例 :

`sin2costan(sin(x))` は tan(x)*cos(x)を返します

[書き換え] : [余弦]

[acosx→asinx]

acos(x)を π/2 – asin(x)で書き換えた式を返します。

`acos2asin(Expr)`

例 :

`acos2asin(acos(x)+asin(x))` は π/2 を返します

[acosx→atanx]

acos(x)を $\frac{\pi}{2}-\operatorname{atan}\left(\frac{x}{\sqrt{1-x^2}}\right)$ で書き換えた式を返します。

`cos2atan(Expr)`

例 :

`acos2atan(2*acos(x))` は $2\cdot\left(\frac{\pi}{2}-\operatorname{atan}\left(\frac{x}{\sqrt{1-x^2}}\right)\right)$ を返します

[cosx → sinx/tanx]

cos(x)を sin(x)/tan(x)で書き換えた式を返します。

`cos2atan(Expr)`

例：

`cos2sintan(cos(x))` は sin(x)/tan(x)を返します

[書き換え]：[正接]

[atanx→asinx]

atan(x)を $\text{asin}\left(\frac{x}{\sqrt{1-x^2}}\right)$ で書き換えた式を返します。

`atan2asin(Expr)`

例：

`atan2asin(atan(2*x))` は $\text{asin}\left(\frac{2\cdot x}{\sqrt{1-(2\cdot x)^2}}\right)$ を返します

[atanx→acosx]

atan(x)を $\frac{\pi}{2}-\text{acos}\left(\frac{x}{\sqrt{1+x^2}}\right)$ で書き換えた式を返します。

`atan2acos(Expr)`

tanx→sinx/cosx

tan(x)を sin(x)/cos(x)で書き換えた式を返します。

`tan2sincos(Expr)`

例：

`tan2sincos(tan(x))` は sin(x)/cos(x)を返します

[halftan]

sin(x)、cos(x)、または tan(x)を tan(x/2)で書き換えた式を返します。

`halftan(Expr)`

例：

`halftan(sin(x))` は 2*tan(1/2*x)/(tan(1/2*x)²+1)を返します

[書き換え]：[三角法]

[trigx→sinx]

sin(x)^2+cos(x)^2=1 および tan(x)=sin(x)/cos(x)を使用して簡略化された式を返します。結果では sin(x)が cos(x)および tan(x)よりも優先されます。

`trigsin(Expr)`

例：

`trigsin(cos(x)^4+sin(x)^2)` は sin(x)^4-sin(x)^2+1 を返します

[trigx→cosx]

sin(x)^2+cos(x)^2=1 および tan(x)=sin(x)/cos(x)を使用して簡略化された式を返します。結果では cos(x)が sin(x)および tan(x)よりも優先されます。

`trigcos(Expr)`

例：

`trigcos(sin(x)^4+sin(x)^2)` は cos(x)^4-3*cos(x)^2+2 を返します

[trigx→tanx]

sin(x)^2+cos(x)^2=1 および tan(x)=sin(x)/cos(x)を使用して簡略化された式を返します。結果では tan(x)が sin(x)および cos(x)よりも優先されます。

`trigtan(Expr)`

例：

`trigtan(cos(x)^4+sin(x)^2)` は(tan(x)^4+tan(x)^2+1)/(tan(x)^4+2*tan(x)^2+1)を返します

[atrig2ln]

逆三角関数を使用した式を、自然対数関数を使用して書き直して返します。

`trig2ln(Expr)`

例：

`atrig2ln(atan(x))` は $\frac{i}{2}\cdot\ln\frac{(i+x)}{(i-x)}$ を返します

[tlin]

三角関数の式に含まれる積および整数乗を線形化して返します。

`tlin(ExprTrig)`

例：

`tlin(sin(x)^3)` は $\frac{3}{4}\cdot\sin(x)-\frac{1}{4}\cdot\sin(3\cdot x)$ を返します

[tcollect]

線形化され、同じ角度の正弦項および余弦項をまとめた三角関数の式を返します。

`tcollect(Expr)`

例：

`tcollect(sin(x)+cos(x))` は次を返します

$\sqrt{2}\cdot\cos\left(x-\frac{1}{4}\cdot\pi\right)$

[trigexpand]

三角関数の式を展開形式で返します。

`trigexpand(Expr)`

例：

`trigexpand(sin(3*x))` は(4*cos(x)^2- 1)*sin(x)を返します

[trig2exp]

三角関数を使用した式を複素指数関数として書き換えて返します（線形化なし）。

`trig2exp(Expr)`

例：

`trig2exp(sin(x))` は次を返します

$$\frac{-i}{2}\cdot\left(\exp(i\cdot x)-\frac{1}{\exp(i\cdot x)}\right)$$

[整数]

[除数]

整数または整数リストの約数を含むリストを返します。

`idivis(Integer)` または `idivis({Intgr1, Intgr2,…})`

例：

`idivis(12)` は[1, 2, 3, 4, 6, 12]を返します

[因数]

整数を素因数分解したものを返します。

注記： 場合によっては、`ifactor` が失敗することがあります。この場合、-1 と入力の反数の積が返されます。-1 は、因数分解に失敗したことを示します。

`ifactor(Integer)`

例：

`ifactor(150)` は 2*3*5^2 を返します

[因数リスト]

整数または整数リストの素因数を含むベクトルを返します。各因数の後には重複度が続きます。

`ifactor(Integer)`

または

`ifactors({Intgr1, Intgr2,…})`

例：

`ifactors(150)` は[2, 1, 3, 1, 5, 2]を返します

[GCD]

2 つ以上の整数の最大公約数を返します。

`gcd(Intgr1, Intgr2,…)`

例：

`gcd(32,120,636)` は 4 を返します

[LCM]

2 つ以上の整数の最小公倍数を返します。

`lcm(Intgr1, Intgr2,…)`

例：

`lcm(6,4)` は 12 を返します

[整数] : [素数]

[素数判定]

指定された整数が素数かどうかを検査します。

`isPrime(Integer)`

例：

`isPrime(19999)` は false を返します

[N 番目の素数]

n 番目の素数を返します。

`ithprime(Intg(n))`（ここで、n は 1 ~ 200,000 の間です）

例：

`ithprime(5)` は 11 を返します

[次の素数]

ある整数の後の次の素数または疑似素数を返します。

`nextprime(Integer)`

例：

`nextprime(11)` は 13 を返します

[前の素数]

ある整数より小さい素数または疑似素数のうち、その整数に最も近いものを返します。

`prevprime(Integer)`

例：

`%(11)` は 7 を返します

[オイラー]

ある整数についてオイラーのトーシェントを計算します。

```
euler(Integer)
```

例：

```
euler(6)
```
は 2 を返します

[整数]：[除算]

[商]

2 つの整数に関するユークリッド除法の整数の商を返します。

```
iquo(Intgr1, Intgr2)
```

例：

```
iquo(63, 23)
```
は 2 を返します

[余り]

2 つの整数に関するユークリッド除法の剰余を返します。

```
irem(Intgr1, Intgr2)
```

例：

```
irem(63, 23)
```
は 17 を返します

[a^n MOD p]

3 つの整数 a、n、および p について、[0, p-1]で a の n 乗を p で割った余りを返します。

```
powmod(a, n, p,[Expr],[Var])
```

例：

```
powmod(5,2,13)
```
は 12 を返します

[中国式剰余]

2 つの等式に対する中国式整数剰余定理。[a p]および[b q]という 2 つの整数ベクトルを指定し、x ≡ r mod n となる 2 つの整数を含むベクトル[r n]を返します。この場合、x は、x ≡ a mod p および x ≡ b mod q を満たし、かつ n=p*q です。

```
ichinrem([a,p],[b,q])
```

例：

```
ichinrem([2, 7], [3, 5])
```
は[23, 35]を返します

[多項式]

[根を求める]

x の多項式（または多項式の係数を含むベクトル）を与えると、その根を含むベクトルを返します。

proot(Poly)または proot(Vector)

例：

proot([1,0,-2])は[-1.41421356237,1.41421356237]を返します

[係数]

x の多項式を与えると、その係数を含むベクトルを返します。多項式の変数が x 以外の場合は、その変数を 2 番目の引数として宣言します。オプションの 3 番目の引数として整数が指定された場合、多項式の係数のうち、その整数に一致する次数を持つ係数を返します。

coeff(Poly, [Var], [Integer])

例：

coeff(x^2-2)は[1 0 -2]を返します

coeff(y^2-2, y, 1)は 0 を返します

[除数]

多項式を与えると、その約数を含むベクトルを返します。

divis(Poly)または divis({Poly1, Poly2,…})

例：

divis(x^2-1)は[1 -1+x 1+x (-1+x)*(1+x)]を返します

[因数リスト]

多項式または多項式リストの素因数を含むベクトルを返します。各因数の後には重複度が続きます。

factors(Poly)または factors({Poly1, Poly2,…})

例：

factors(x^4-1)は[x-1 1 x+1 1 x^2+1 1]を返します

[GCD]

2 つ以上の多項式の最大公約数を返します。

gcd(Poly1,Poly2...)

例：

gcd(x^4-1, x^2-1)は x^2-1 を返します

[LCM]

2 つ以上の多項式の最小公倍数を返します。

lcm(Poly1, Poly2,…)

例：

lcm(x^2-2*x+1,x^3-1)は(x-1)*(x^3-1)を返します

[多項式]：Create（作成）

[多項式から係数へ]

多項式を与えると、その係数を含むベクトルを返します。2 番目の引数として変数を指定すると、その変数について多項式の係数を返します。2 番目の引数として変数のリストを指定すると、多項式の内部形式を返します。

`symb2poly(Expr,[Var])` または `symb2poly(Expr, {Var1, Var2,…})`

例：

`symb2poly(x*3+2.1)` は[3 2.1]を返します

[係数から多項式へ]

1 つのベクトルを引数として指定し、引数のベクトルから取得した係数（降順）を使用して、x についての多項式を返します。2 番目の引数として変数が指定された場合は、その変数についての同様な多項式を返します。

`poly2symb(Vector, [Var]))`

例：

`poly2symb([1,2,3],x)` は(x+2)*x+3 を返します

[根から係数へ]

引数のベクトルに指定された根を持つ 1 変数多項式の係数（降順）を含むベクトルを返します。

`pcoef(List)`

例：

`pcoeff({1,0,0,0,1})` は[1 -2 1 0 0 0]を返します

[根から多項式へ]

ベクトルを引数として取ります。このベクトルには、有理関数の各根または極が含まれます。各根または極の後にはその次数が続きます（極の次数は負）。引数のベクトルで指定された根と極（およびそれらの次数）を持つ x の有理関数を返します。

`fcoeff(Vector)`（ここで、`Vector` の形式は `[Root1, Order1, Root2, Order2, …]`）

例：

`fcoeff([1,2,0,1,3,-1])` は(x-1)^2*x*(x-3)^-1 を返します

[乱数]

次数が `Integer` の多項式の係数を含むベクトルを返します。係数は、一様分布による-99 から 99 までの範囲か、`Interval` で指定される区間での乱数整数です。任意の変数でランダムな多項式を作成するには、poly2symbol とともに使用します。

`randpoly(Integer, Interval, [Dist])`（ここで、`Interval` の形式は `Real1..Real2.`）

例：

`randpoly(t, 8, -1..1)`は、すべてが－1 から 1 までの 9 つの整数乱数によるベクトルを返します。

[最小]

行列のみを引数として指定すると、行列の x についての最小多項式を係数のリストとして書き出して返します。行列および変数を引数として指定すると、その行列の最小多項式を、その変数について記号形式で書き出して返します。

`pmin(Mtrx,[Var])`

例：

`pmin([[1,0],[0,1]],x)`は x-1 を返します

[多項式] : [代数]

[商]

2 つの多項式のユークリッド商の係数を含むベクトルを返します。多項式は、係数のリストまたは記号形式で記述できます。

`quo(List1, List2, [Var])`

または

`quo(Poly1, Poly2, [Var])`

例：

`quo({1, 2, 3, 4}, {-1, 2})`は[-1 -4 -11]を返します

[余り]

2 つの多項式のユークリッド商の余りの係数を含むベクトルを返します。多項式は、係数のリストまたは記号形式で記述できます。

`rem(List1, List2, [Var])`

または

`rem(Poly1, Poly2, [Var])`

例：

`rem({1, 2, 3, 4}, {-1, 2})`は[26]を返します

[次数]

多項式の次数を返します。

`degree(Poly)`

例：

`degree(x^3+x)`は 3 を返します

[次数による因数分解]

x についての n 次の多項式から因数 xn を括り出し、結果の積を返します。

factor_xn(Poly)

例：

factor_xn(x^4-1)は x^4*(1-x^-4)を返します

[係数 GCD]

多項式の係数の最大公約数（GCD）を返します。

content(Poly,[Var])

例：

content(2*x^2+10*x+6)は 2 を返します

[ゼロカウント]

a および b が実数の場合、区間内[a,b]で指定された多項式の符号変化の数が返されます。a または b が実数でない場合、a および b で囲まれる長方形の内部での複素根の数が返されます。Var が省略された場合、変数は x と見なされます。

sturmab(Poly[,Var],a,b)

例：

sturmab(x^2*(x^3+2),-2,0)は 1 を返します

sturmab(n^3-1,n,-2-i,5+3i)は 3 を返します

[中国式剰余]

多項式の係数を 2 つの行にそれぞれ含む行列を 2 つ与えると、多項式の中国式剰余も行列で書き出して返します。

chinrem(Matrix1,Matrix2)

例：

chinrem $\left(\begin{bmatrix}1&2&0\\1&0&1\end{bmatrix},\begin{bmatrix}1&1&0\\1&1&1\end{bmatrix}\right)$ は次を返します。

[[2 2 1] [1 1 2 1 1]]

[多項式]：[特殊]

[円分]

整数の円分多項式の係数のリストを返します。

cyclotomic(Integer)

例：

cyclotomic(20)は[1 0 −1 0 1 0 −1 0 1]を返します

[グレブナー基底]

多項式のベクトルと、変数のベクトルを与えると、多項式のセットによるイデアルのグレブナー基底を返します。

```
gbasis([Poly1 Poly2…], [Var1 Var2…])
```

例：

`gbasis([x^2-y^3,x+y^2],[x,y])` は[y^4- y^3,x+y^2]を返します

[グレブナー余り]

多項式と、多項式のベクトルおよび変数のベクトルの両方を与えると、多項式のベクトルのグレブナー基底で多項式を割った余りを返します。

```
greduce(Poly1, [Poly2 Poly3 …], [Var1 Var2…])
```

例：

`greduce(x*y-1,[x^2-y^2,2*x*y-y^2,y^3],[x,y])` は 1/2*y^2-1 を返します

[エルミート]

次数 n のエルミート多項式を返します。n は 1556 より小さい整数です。

```
hermite(Integer)
```

例：

`hermite(3)` は 8*x^3-12*x を返します

[ラグランジュ]

横座標のベクトルおよび縦座標のベクトルを与えると、2 つのベクトルで指定される点のラグランジュ多項式を返します。この関数は、最初の行に横座標、2 番目の行に縦座標を含む行列を引数として指定することもできます。

```
lagrange([X1 X2…], [Y1 Y2…]))
```

または

$$\text{lagrange}\left(\begin{bmatrix} X1 & X2 & \ldots \\ Y1 & Y2 & \ldots \end{bmatrix}\right)$$

例：

`lagrange([1,3],[0,1])` は(x-1)/2 を返します

[ラゲール]

整数 n を与えると、次数 n のラゲール多項式を返します。

```
laguerre(Integer)
```

例：

`laguerre(4)` は 1/24*a^4+(-1/6)*a^3*x+5/ 12*a^3+1/4*a^2*x^2+(-3/2)*a^2*x+35/24*a^2+(-1/6)*a*x^3+7/4*a*x^2+(-13/3)*a*x+25/12*a+1/ 24*x^4+(-2/3)*x^3+3*x^2-4*x+1 を返します

[ルジャンドル]

整数 n を与えると、次数 n のルジャンドル多項式を返します。

```
legendre(Integer)
```

例：

`legendre(4)` は 35/8 · x^4 + 15/4 x^2 + 3/8 を返します

[チェビシェフの Tn]

整数 n を与えると、次数 n のチェビシェフ多項式（第 1 種）を返します。

```
tchebyshev1(Integer)
```

例：

`tchebyshev1(3)` は 4*x^3-3*x を返します

[チェビシェフの Un]

整数 n を与えると、次数 n のチェビシェフ多項式（第 2 種）を返します。

```
tchebyshev2(Integer)
```

例：

`tchebyshev2(3)` は 8*x^3-4*x を返します

[プロット]

[関数]

幾何アプリケーションの[記号表示]で関数グラフを定義するために使用されます。独立変数 x を使って記述された式のグラフをプロットします。変数は小文字にしてください。

```
plotfunc(Expr)
```

例：

`plotfunc(3*sin(x))` は y=3*sin(x)のグラフを描画します

[等高線]

幾何アプリケーションの[記号表示]で等高線グラフを定義するために使用されます。x と y の式および変数リストと値リストを与えると、面 z=f(x,y)の等高線グラフをプロットします。具体的には、値リストで定義される等高線（z1、z2 など）をプロットします。x および y の両方のステップ値を指定することもできます。

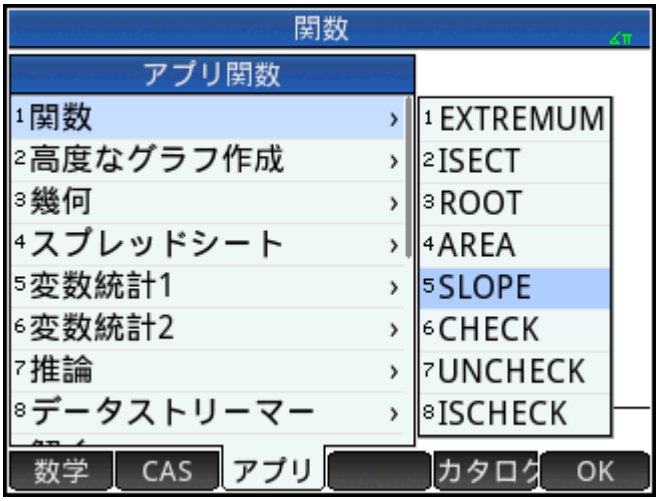
```
plotcontour(Expr, [ListVars], [ListVals], [xstep=val1], [ystep=val2])
```

例：

`plotcontour(x^2+2*y^2-2, {x, y}, {2, 4, 6})` は、z=x^2+2*y^2 – 2 の 3 つの等高線（z=2、z=4、z=6 に対応）を描画します。

[アプリ]メニュー

を押すと、ツールボックスのメニューが開きます（その 1 つが[アプリ]メニュー）。アプリケーション関数は、一般的な計算を実行するために HP アプリケーション内で使用される関数です。たとえば関数アプリケーションでは、[プロット表示]の Fcn（関数）メニューに、指定した関数の指定した点での傾きを計算する SLOPE（傾き）関数があります。SLOPE（傾き）関数は、ホーム表示やプログラムから使用しても同じ結果が得られます。このセクションで説明するアプリケーション関数は、アプリケーション別にグループ化されています。



関数アプリケーション関数

関数アプリケーション関数は、関数アプリケーションの[関数]メニューにある機能と同じ機能を提供します。これらすべての演算は、関数に対して機能します。関数は、X の式か、関数アプリケーション変数の名前 F0 ~ F9 として指定できます。

[面積]

曲線の下または曲線間の面積。関数の下または 2 つの関数の間の符号付き面積を求めます。X 値の下側から X 値の上側までの間で、関数 Fn の下か、関数 Fn の下で関数 Fm の上の部分の面積を求めます。

```
AREA(Fn,[Fm,]lower,upper)
```

例：

`AREA(-X,X²-2,-2,1)` は 4.5 を返します

[極値]

関数の極値。関数 Fn について、推測した X 値である guess に最も近い極値（存在する場合）を求めます。

```
EXTREMUM(Fn, guess)
```

例：

`EXTREMUM(X)²-X-2,0` は 0.5 を返します

[ISECT]

2 つの関数の交点。推測した X 値である guess に最も近い、2 つの関数 Fn と Fm の交点（存在する場合）を求めます。

`ISECT(Fn,Fm,guess)`

例：

`ISECT(X,3-X,2)` は 1.5 を返します

[根]

関数の根。推測した X 値である guess に最も近い関数 Fn の根（存在する場合）を求めます。

`ROOT(Fn,guess)`

例：

`ROOT(3-X², 2)` は 1.732… を返します

[傾き]

関数の傾き。関数 Fn の X 値での傾き（その値で関数の導関数が存在する場合）を返します。

`SLOPE(Fn,value)`

例：

`SLOPE(3-X²,2)` は-4 を返します

解くアプリケーション関数

解くアプリケーションでは、単一の関数を使用して、与えられた方程式または式をその変数のどれかについて解きます。En は、方程式または式にするか、Solve Symbolic（記号を解く）変数 E0 ~ E9 のどれかの名前にすることができます。

[解く]

変数のどれかについて方程式を解きます。変数 var の値の初期値として guess の値を使用し、変数 Var の方程式 En を解きます。En が式の場合、式をゼロに等しくする変数 var の値が返されます。

`SOLVE(En,var,guess)`

例：

`SOLVE(X²-X-2,X,3)` は 2 を返します

この関数は、見つかった解のタイプを示す次のような整数も返します。

`0`：正確な解が見つかりました

`1`：近似解が見つかりました

`2`：解にできるだけ近い極値が見つかりました

`3`：解、近似解、極値のどれも見つかりませんでした

スプレッドシート アプリケーション関数

スプレッドシート アプリケーション関数は、Toolbox（ツールボックス）の[アプリ]メニューから選択できます。 Mem を押し、 アプリ をタップして、**[スプレッドシート]**を選択します。スプレッドシート アプリケーションが開いているときに[表示]メニュー（ View Copy ）から選択することもできます。

ほとんどのスプレッドシート関数の構文は、次のパターンに従います。

`functionName(input,[optional parameters])`

`input` は関数の入力リストです。これは、セル範囲参照や単純なリストにしたり、値のリストになるような任意のオブジェクトにしたりすることができます。

便利なオプション パラメーターの 1 つが、`configuration` です。これは、出力する値を制御する文字列です。このパラメーターを指定しなかった場合は、初期設定の出力が生成されます。また、値の順序を、文字列内で表示する順序によって制御できます。

例：`=STAT1(A25:A37)` は、セル A25 ~ A37 の数値に基づいて、次のような初期設定の出力を生成します。

ただし、標本数および標準偏差のみを表示させたい場合は、`=STAT1(A25:A37,"h n σ")` と入力します。ここで構成文字列が示しているのは、行見出し（h）が必要で、標本数（n）および標準偏差（σ）のみが表示されるということです。

[和]

1 つの範囲の数値の和を計算します。

```
SUM([input])
```

たとえば、SUM(B7:B23) は、B7 から B23 の範囲の数の和を返します。SUM(B7:C23) のように、セルのブロックを指定することもできます。

指定された範囲のセルに数値ではないオブジェクトが含まれている場合、エラーが返されます。

[平均]

数の範囲の算術平均を計算します。

```
AVERAGE([input])
```

たとえば、AVERAGE(B7:B23) は、B7 から B23 までの範囲の数の算術平均を返します。AVERAG(B7:C23) のように、セルのブロックを指定することもできます。

指定された範囲のセルに数値ではないオブジェクトが含まれている場合、エラーが返されます。

[償却]

償却。指定された期間でのローンの元本、利息、および残金を計算します。金融アプリケーションで 償却 を押したときと同じです。

```
AMORT(Range, NbPmt, IPYR, PV, PMTV[, PPYR=12, CPYR=PPYR, GSize=PPYR,
BEG=0, fix=current], "configuration"])
```

Range：結果が配置されるセル範囲です。1 つのセルのみが指定されると、そのセルを起点として範囲が自動的に計算されます。

Configuration：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

h：行見出しを表示します

H：列見出しを表示します

S：期間の開始を表示します

E：期間の終了を表示します

P：この期間に支払われた元本を表示します

B：期末の残金を表示します

I：この期間に支払われた利息を表示します

`fix` 以外の他の入力パラメーターはすべて、金融アプリケーションの数値表示変数です。詳しくは、400 ページの金融アプリケーション関数を参照してください。必須のパラメーターは最初の 4 つのみです。`fix` は、表示される結果で使用される小数点以下の桁数です。

[STAT1]

STAT1 関数は、データのリストに基づいて一連の 1 変数統計を提供します。

```
STAT1(input_range, [mode], [configuration])
```

input_range はデータ ソースです（A1:D8 など）。

mode は、入力の処理方法を定義します。

有効な mode の値は以下のとおりです。

1 = 単一データ。各列が独立したデータセットとして処理されます。

2 = 頻度データ。列は組で使用され、2 列目は 1 列目の出現頻度と見なされます。

複数の列が指定される場合、それらの列は互いに異なる入力データ セットと見なされます。1 つの行のみが選択された場合、その行は 1 つのデータ セットと見なされます。2 つの列が選択された場合、モードの初期設定は頻度になります。

configuration は、どの行にどの値があるか、および行と列に見出しがあるかどうかを定義します。スプレッドシートの値は、それぞれの値の記号が入力された順序で表示されます。

有効な configuration の値は以下のとおりです。

- H（列見出しを挿入します）
- h（行見出しを挿入します）
- MeanX
- Σ
- Σ^2
- s
- s^2
- σ
- σ^2
- serr
- ss
- n
- min
- q1

- med
- q3
- 最大

たとえば、「h n σ」と指定すると、結果のスプレッドシートの 1 列目には行の見出しが含まれ、1 行目は入力データの項目数、2 行目は母集団の標準偏差になります。

例：

```
STAT1(A25:A37)

STAT1(A25:A37,"h n σ")
```

STAT2

STAT2 関数は、一連の 2 変数統計を提供します。

```
STAT2(input_range, [mode], [configuration])
```

input_range はデータ ソースです（A1:D8 など）。

mode は、入力の処理方法を定義します。

有効な mode の値は以下のとおりです。

1 = 単一データ。各列ペアが独立したデータセットとして処理されます。

2 = 頻度データ。列は 3 列のグループで使用され、3 列目が列ペアの外観の頻度として使用されます。

3 つ以上の列が指定される場合、各ペアは互いに異なる入力データ セットと見なされます。1 つの列ペアのみが選択された場合、その列ペアは 1 つのデータ セットと見なされます。3 つの列が選択された場合、モードの初期設定は頻度になります。

configuration は、どの行にどの値があるか、および行と列に見出しがあるかどうかを定義します。スプレッドシートの値は、それぞれの値の記号が入力された順序で表示されます。

有効な configuration の値は以下のとおりです。

- H（列見出しを挿入します）
- h（行見出しを挿入します）
- MeanX
- Σx
- Σx^2
- sx
- sx^2
- σx
- σx^2
- serrx
- ssx
- n
- $\bar{y}$

- Σy
- Σy²
- sy
- sy²
- σy
- σy²
- serry
- ssy
- Σxy

たとえば、「h n σy」と指定すると、結果のスプレッドシートの 1 列目には行の見出しが含まれ、1 行目は入力データの項目数、2 行目は y の標準偏差になります。

例：

```
STAT2(A25:B37)

STAT2(A25:B37,"h n σy")
```

[REGRS]

入力データを指定された関数にフィットしようとします（初期設定は線形）。

- [入力範囲]：たとえば、A1:D8 のようにデータ ソースを指定します。これには偶数個の列が含まれている必要があります。各組はそれぞれ異なるデータ ポイント セットと見なされます。
- [モデル]：回帰に使用されるモデルを指定します。

  1 y= sl*x+int

  2 y= sl*ln(x)+int

  3 y= int*exp(sl*x)

  4 y= int*x^sl

  5 y= int*sl^x

  6 y= sl/x+int

  7 y= L/(1 + a*exp(b*x))

  8 y= a*sin(b*x+c)+d

  9 y= cx^2+bx+a

  10 y= dx^3+cx^2+bx+a

  11 y= ex^4+dx^3+cx^2+bx+a
- [構成]：どの行にどの値を配置するか、および行と列の見出しが必要かどうかを示す文字列。スプレッドシート内に表示させる順序で、各パラメーターを配置します（構成文字列を指定しなかった場合は、初期設定の文字列が使用されます）。有効なパラメーターは次のとおりです。
  - H（列見出しを配置します）
  - h（行見出しを配置します）

- sl（傾き、モデル 1 から 6 の場合のみ有効です）
- int（切片、モデル 1 から 6 の場合のみ有効です）
- cor（相関、モデル 1 から 6 の場合のみ有効です）
- cd（決定係数、モデル 1 から 6、8 から 10 の場合のみ有効です）
- sCov（標本共分散、モデル 1 から 6 の場合のみ有効です）
- pCov（母共分散、モデル 1 から 6 の場合のみ有効です）
- L（L パラメーター、モデル 7 の場合）
- a（a パラメーター、モデル 7 から 11 の場合）
- b（b パラメーター、モデル 7 から 11 の場合）
- c（c パラメーター、モデル 8 から 11 の場合）
- d（d パラメーター、モデル 8、10 から 11 の場合）
- e（e パラメーター、モデル 11 の場合）
- py（2 つのセルを配置します。1 つはユーザー入力用、もう 1 つはその入力に対して予測される y の表示用です）
- px（2 つのセルを配置します。1 つはユーザー入力用、もう 1 つはその入力に対して予測される x の表示用です）

例：`REGRS(A25:B37,2)`

[predY]

指定された x に対して予測される Y を返します。

```
PredY(mode, x, parameters)
```

- `Mode` は、次で使用される回帰モデルを制御します。

  1 y= sl*x+int

  2 y= sl*ln(x)+int

  3 y= int*exp(sl*x)

  4 y= int*x^sl

  5 y= int*sl^x

  6 y= sl/x+int

  7 y= L/(1 + a*exp(b*x))

  8 y= a*sin(b*x+c)+d

  9 y= cx^2+bx+a

  10 y= dx^3+cx^2+bx+a

  11 y= ex^4+dx^3+cx^2+bx+a

- `parameters` は、1 つの引数（回帰直線の係数のリスト）、または連続する n 個の係数です。

[PredX]

指定された y に対して予測される x を返します。

```
PredX(mode, y, parameters)
```

- `Mode` は、次で使用される回帰モデルを制御します。

  1 y= sl*x+int

  2 y= sl*ln(x)+int

  3 y= int*exp(sl*x)

  4 y= int*x^sl

  5 y= int*sl^x

  6 y= sl/x+int

  7 y= L/(1 + a*exp(b*x))

  8 y= a*sin(b*x+c)+d

  9 y= cx^2+bx+a

  10 y= dx^3+cx^2+bx+a

  11 y= ex^4+dx^3+cx^2+bx+a

- `parameters` は、1 つの引数（回帰直線の係数のリスト）、または連続する n 個の係数です。

[HypZ1mean]

1 つの平均についての 1 標本 Z 検定。

```
HypZ1mean(x̄,n,μ0,σ,α,mode,[" configuration"])
```

入力パラメーターは、範囲参照、セル参照のリスト、または単純な値リストにすることができます。

[モード]：使用する対立仮説を指定します。

- 1：$\mu < \mu_0$
- 2：$\mu > \mu_0$
- 3：$\mu \neq \mu_0$

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- acc：検定結果。帰無仮説を棄却する場合は 0、棄却しない場合は 1
- tZ：Z 検定値
- tM：入力 $\bar{x}$ 値
- prob：下側裾確率
- cZ：入力 α 水準に関連付けられた Z 臨界値
- cx1：Z 臨界値に関連付けられた平均の下側臨界値

- cx2：Z 臨界値に関連付けられた平均の上側臨界値
- std：標準偏差

例：

`HypZ1mean(0.461368, 50, 0.5, 0.2887, 0.05, 1, "")`は、スプレッドシート アプリケーション内に 2 つの列を返します。最初の列には見出しが含まれ、2 番目の列には次のそれぞれの値が含まれます。Reject/Fail=1, Test Z = -0.94621, Test $\bar{x}$ = 0.461368, P= 0.172022, Critical Z= -1.64485, Critical $\bar{x}$ = 0.432843.

[HYPZ2mean]

2 つの平均の差に関する 2 標本 Z 検定。

`HypZ2mean(`$_{1,2}$`,n1,n2,σ1,σ2,α,mode,[" configuration"])`

[モード]：使用する対立仮説を指定します。

- 1：$\mu_1 < \mu_2$
- 2：$\mu_1 > \mu_2$
- 3：$\mu_1 \neq \mu_2$

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- acc：検定結果。帰無仮説を棄却する場合は 0、棄却しない場合は 1
- tZ：Z 検定値
- tM：入力 $\Delta\bar{x}$ 値
- prob：下側裾確率
- cZ：入力 α 水準に関連付けられた Z 臨界値
- cx1：Z 臨界値に関連付けられた $\Delta\bar{x}$ の下側臨界値
- cx2：Z 臨界値に関連付けられた $\Delta\bar{x}$ の上側臨界値
- std：標準偏差

例：

`HypZ2mean(0.461368, 0.522851, 50, 50, 0.2887, 0.2887, 0.05, 1, "")`

[HypZ1prop]

1 つの比率に関する 1 標本 Z 検定。

`HypZ1prop(x,n,π0,,α,mode,[" configuration"])`（x は標本の成功数）

[モード]：使用する対立仮説を指定します。

- 1：$\pi < \pi_0$
- 2：$\pi > \pi_0$
- 3：$\pi \neq \pi_0$

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- acc：検定結果。帰無仮説を棄却する場合は 0、棄却しない場合は 1
- tZ：Z 検定値
- tP：検定成功比率
- prob：下側裾確率
- cZ：入力 α 水準に関連付けられた Z 臨界値
- cp1：Z 臨界値に関連付けられた下側臨界成功比率
- cp2：Z 臨界値に関連付けられた上側臨界成功比率
- std：標準偏差

例：

```
HypZ1prop(21, 50, 0.5, 0.05,1, "")
```

[HypZ2prop]

2 つの比率を比較する 2 標本 Z 検定。

HypZ2prop(x_1,x_2,n_1,n_2,,α,mode,[" configuration"])（ここで、x_1 および x_2 は 2 つの標本の成功数）

- 1：$\pi_1 < \pi_2$
- 2：$\pi_1 > \pi_2$
- 3：$\pi_1 \neq \pi_2$

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- acc：検定結果。帰無仮説を棄却する場合は 0、棄却しない場合は 1
- tZ：Z 検定値
- tP：Δπ 検定値
- prob：下側裾確率
- cZ：入力 α 水準に関連付けられた Z 臨界値
- cp1：Z 臨界値に関連付けられた Δπ の下側臨界値
- cp2：Z 臨界値に関連付けられた Δπ の上側臨界値

例：

```
HypZ2prop(21, 26, 50, 50, 0.05, 1, "")
```

[HypT1mean]

平均についての 1 標本 t 検定。

HypT1mean($\bar{x}$,n,μ_0,α,mode,[" configuration"])

- 1：$\mu < \mu_0$
- 2：$\mu > \mu_0$
- 3：$\mu \neq \mu_0$

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- acc：検定結果。帰無仮説を棄却する場合は 0、棄却しない場合は 1
- tT：T 検定値
- tM：入力 $\bar{x}$ 値
- prob：下側裾確率
- df：自由度
- cT：入力 α 水準に関連付けられた T 臨界値
- cx1：T 臨界値に関連付けられた平均の下側臨界値
- cx2：T 臨界値に関連付けられた平均の上側臨界値

例：

```
HypT1mean(0.461368, 0.2776, 50, 0.5, 0.05, 1, "")
```

[HypT2mean]

2 つの平均の差についての 2 標本 T 検定。

HypT2mean ($\bar{x}_1$,$\bar{x}_2$,n_1,n_2,s_1,s_2,α,pooled,mode,[" configuration"])

[プール済み]：標本がプールされるかどうかを指定します

- 0：プールされない
- 1：プールされる
- 1：$\mu_1 < \mu_2$
- 2：$\mu_1 > \mu_2$
- 3：$\mu_1 \neq \mu_2$

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- acc：検定結果。帰無仮説を棄却する場合は 0、棄却しない場合は 1

- tT：T 検定値
- tM：入力 $\Delta\bar{x}$ 値
- prob：下側裾確率
- cT：入力 α 水準に関連付けられた T 臨界値
- cx1：T 臨界値に関連付けられた $\Delta\bar{x}$ の下側臨界値
- cx2：T 臨界値に関連付けられた $\Delta\bar{x}$ の上側臨界値

例：

```
HypT2mean(0.461368, 0.522851, 0.2776, 0.2943,50, 50, 0, 0.05, 1, "")
```

[ConfZ1mean]

平均についての 1 標本正規信頼区間。

```
ConfZ1mean(x̄,n,s,C,[" configuration"])
```

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- Z：Z 臨界値
- zXl：信頼区間の下側限界
- zXh：信頼区間の上側限界
- prob：下側裾確率
- std：標準偏差

例：

```
ConfZ1mean(0.461368, 50, 0.2887, 0.95, "")
```

[ConfZ2mean]

2 つの平均の差についての 2 標本正規信頼区間。

```
ConfZ2mean (x̄1,x̄2,n1,n2,s1,s2,C,[" configuration"]
```

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- Z：Z 臨界値
- zXl：信頼区間の下側限界
- zXh：信頼区間の上側限界
- prob：下側裾確率
- std：標準偏差

例：

```
ConfZ2mean(0.461368, 0.522851, 50, 50, 0.2887, 0.2887, 0.95, "")
```

[ConfZ1prop]

比率についての 1 標本正規信頼区間。

```
ConfZ1prop(x,n,C,[" configuration"])
```

[構成]:表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定(見出しを含むすべての結果)を表示します。構成文字列内のオプションはスペースで区切ります。

- h:見出しセルを作成します
- Z:Z 臨界値
- zXl:信頼区間の下側限界
- zXh:信頼区間の上側限界
- zXm:信頼区間の中点
- std:標準偏差

例:

```
ConfZ1prop(21, 50, 0.95, "")
```

[ConfZ2prop]

2 つの比率の差についての 2 標本正規信頼区間。

```
ConfZ2prop(x1,x2,n1,n2,C,[" configuration"])
```

[構成]:表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定(見出しを含むすべての結果)を表示します。構成文字列内のオプションはスペースで区切ります。

- h:見出しセルを作成します
- Z:Z 臨界値
- zXl:信頼区間の下側限界
- zXh:信頼区間の上側限界
- zXm:信頼区間の中点
- std:標準偏差

例:

```
ConfZ2prop(21, 26, 50, 50, 0.95, "")
```

[ConfT1mean]

平均についての 1 標本スチューデントの T 信頼区間。

```
ConfT1mean(x̄,s,n,C,[" configuration"])
```

[構成]:表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定(見出しを含むすべての結果)を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- DF：自由度
- T：T 臨界値
- tXl：信頼区間の下側限界
- tXh：信頼区間の上側限界
- std：標準偏差

例：

```
ConfT1mean(0.461368, 0.2776, 50, 0.95, "")
```

[ConfT2mean]

2 つの平均の差についての 2 標本スチューデント T 検定の信頼区間

ConfT2mean $(\bar{x}_1, \bar{x}_2, n_1, n_2, s_1, s_2, C,$ pooled, ["configuration"]

[構成]：表示される結果の内容および表示される順序を制御する文字列です。空の文字列""は初期設定（見出しを含むすべての結果）を表示します。構成文字列内のオプションはスペースで区切ります。

- h：見出しセルを作成します
- DF：自由度
- T：T 臨界値
- tXl：信頼区間の下側限界
- tXh：信頼区間の上側限界
- tXm：信頼区間の中点
- std：標準偏差

例：

```
ConfT2mean(0.461368, 0.522851, 0.2776, 0.2943, 50, 50, 0, 0.95, "")
```

1 変数統計アプリケーション関数

1 変数統計アプリケーションには、1 変数統計アプリケーションの[記号表示]で定義された統計分析（H1 から H5）のどれかに基づいて要約統計を計算するため、共に動作するように設計された 3 つの関数があります。

[Do1VStats]

1 変数統計を行います。1 変数統計アプリケーションの[数値表示]で 統計 をタップしたときと同じ計算を実行し、結果を対応する 1 変数統計アプリケーション変数に格納します。Hn は、1 変数統計アプリケーションの記号表示変数 H1 から H5 のどれかである必要があります。

```
Do1VStats(Hn)
```

例：

Do1VStats(H1) は、現在定義されている H1 分析の要約統計を実行します。

[SetFreq]

頻度を設定します。1 変数統計アプリケーションの[記号表示]で定義された統計分析のどれか（H1 から H5）に対する頻度を設定します。頻度は、列 D0 から D9 のどれか、または任意の正の整数とすることができます。Hn は、1 変数統計アプリケーションの記号表示変数 H1 から H5 のどれかである必要があります。Dn を使用する場合、Dn は列変数 D0 から D9 のどれかである必要があります。それ以外の場合、value は正の整数である必要があります。

`SetFreq(Hn,Dn)` または `SetFreq(Hn,value)`

例：

`SetFreq(H2,D3)` は、H2 分析の**[頻度]**フィールドでリスト D3 が使用されるように設定します。

[SetSample]

サンプル データを設定します。1 変数統計アプリケーションの[記号表示]で定義された統計分析のどれか（H1 から H5）に対するサンプル データを設定します。統計分析 H1 から H5 のどれかに対するデータ列を、列変数 D0 から D9 のどれかに設定します。

`SetSample(Hn,Dn)`

例：

`SetSample(H2,D2)` は、H2 分析の**[独立列]**フィールドでリスト D2 のデータが使用されるように設定します。

2 変数統計アプリケーション関数

2 変数統計アプリケーションには、たくさんの関数があります。一部は、2 変数統計アプリケーションの[記号表示]で定義された統計分析（S1 から S5）のどれかに基づいて要約統計を計算するように設計されています。そのほか、分析の一部で指定されるフィットに基づいて X 値および Y 値を予測するものもあります。

[PredX]

X を予測します。最初に見つかったアクティブな分析（S1 から S5）のフィットに基づいて、与えられた y 値に対する x 値を予測します。

`PredX(value)`

[PredY]

Y を予測します。最初に見つかったアクティブな分析（S1 から S5）のフィットに基づいて、与えられた x 値に対する y 値を予測します。

`PredY(value)`

[Resid]

残差。指定された分析（S1 から S5）に対する残差のリストを、[記号表示]でその分析に対して定義されたデータおよびフィットに基づいて返します。

`Resid(Sn)` または `Resid()`

Resid()は、[記号表示]で最初に定義された分析（S1 から S5）を検索します。

[Do2VStats]

2 変数統計を行います。2 変数統計アプリケーションの[数値表示]で 統計 をタップしたときと同じ計算を実行し、結果を対応する 2 変数統計アプリケーション変数に格納します。Sn は、2 変数統計アプリケーションの記号表示変数 S1 から S5 のどれかである必要があります。

```
Do2VStats(Sn)
```

例：

```
Do1VStats(S1)
```
は、現在定義されている S1 分析の要約統計を実行します。

[SetDepend]

従属列を設定します。統計分析 S1 から S5 のどれかの従属列を、列変数 C0 から C9 のどれかに設定します。

```
SetDepend(Sn,Cn)
```

例：

```
SetDepend(S1,C3)
```
は、S1 分析の[従属列]フィールドでリスト C3 のデータが使用されるように設定します。

[SetIndep]

独立列を設定します。統計分析 S1 から S5 のどれかの独立列を、列変数 C0 から C9 のどれかに設定します。

```
SetIndep(Sn,Cn)
```

例：

```
SetIndep(S1, C2)
```
は、S1 分析の**[独立列]**フィールドでリスト C2 のデータが使用されるように設定します。

推論アプリケーション関数

推論アプリケーションには、推論アプリケーションの[数値表示]で 計算 をタップしたときと同じ結果を返す単一の関数があります。結果は、推論アプリケーション変数 Method、Type、および AltHyp の内容に依存します。

[DoInference]

信頼区間を計算するか仮説を検証します。[記号表示]および[数値表示]の現在の設定を使用して、信頼区間を計算するか仮説を検定します。推論アプリケーションの[数値表示]で 計算 をタップしたときと同じ計算を実行し、結果を対応する推論アプリケーション変数に格納します。

```
DoInference()
```

[HypZ1mean]

1 つの平均についての 1 標本 Z 検定。以下を含むリストを返します（表示順）。

- 帰無仮説を棄却する場合は 0、棄却しない場合は 1
- Z 検定値

- 入力 $\bar{x}$ 値
- 上側裾確率
- 入力 α 水準に関連付けられた上側 Z 臨界値
- Z 臨界値に関連付けられた統計の臨界値

HypZ1mean($\bar{x}$,n,μ_0,σ,α,mode)

[モード]：使用する対立仮説を指定します。

- 1：$\mu < \mu_0$
- 2：$\mu > \mu_0$
- 3：$\mu \neq \mu_0$

例：

HypZ1mean(0.461368, 50, 0.5, 0.2887, 0.05, 1) returns {1, -.9462…, 0.4614, 0.8277…, 1.6448…, 0.5671…}

[HypZ2mean]

平均についての 2 標本 Z 検定。以下を含むリストを返します（表示順）。

- 帰無仮説を棄却する場合は 0、棄却しない場合は 1
- Z 検定値
- tZ：Z 検定値
- $\Delta\bar{x}$ 検定値
- 上側裾確率
- 入力 α 水準に関連付けられた上側 Z 臨界値
- Z 臨界値に関連付けられた $\Delta\bar{x}$ の臨界値

HypZ2mean($\bar{x}_1$ $\bar{x}_2$,n_1,n_2,σ_1,σ_2,α,mode)

[モード]：使用する対立仮説を指定します。

- 1：$\mu_1 < \mu_2$
- 2：$\mu_1 > \mu_2$
- 3：$\mu_1 \neq \mu_2$

例：

HypZ2mean(0.461368, 0.522851, 50, 50, 0.2887, 0.2887, 0.05, 1)は{1, -1.0648…, -0.0614…, 0.8565…, 1.6448…, 0.0334…}を返します。

[HypZ1prop]

1 比率の Z 検定。以下を含むリストを返します（表示順）。

- 帰無仮説を棄却する場合は 0、棄却しない場合は 1
- Z 検定値

- π 検定値
- 上側裾確率
- 入力 α 水準に関連付けられた上側 Z 臨界値
- Z 臨界値に関連付けられた π の臨界値

HypZ1mean(0.461368, 50, 0.5, 0.2887, 0.05, 1)HypZ1prop(x,n,π_0,α,mode)

[モード]：使用する対立仮説を指定します。

- 1：$\pi < \pi_0$
- 2：$\pi > \pi_0$
- 3：$\pi \neq \pi_0$

例：

HypZ1prop(21, 50, 0.5, 0.05,1) は{1, -1.1313…, 0.42, 0.8710…, 1.6448…, 0.6148…}を返します

[HypZ2prop]

比率についての 2 標本 Z 検定。以下を含むリストを返します（表示順）。

- 帰無仮説を棄却する場合は 0、棄却しない場合は 1
- Z 検定値
- Z 検定値
- Δπ 検定値
- 上側裾確率
- 入力 α 水準に関連付けられた上側 Z 臨界値
- Z 臨界値に関連付けられた Δπ の臨界値

HypZ2prop($\bar{x}_1$,$\bar{x}_2$,n_1,n_2,α,mode)

[モード]：使用する対立仮説を指定します。

- 1：$\pi_1 < \pi_2$
- 2：$\pi_1 > \pi_2$
- 3：$\pi_1 \neq \pi_2$

例：

HypZ2prop(21, 26, 50, 50, 0.05, 1) は{1, -1.0018…, -0.1, 0.8417…, 1.6448…, 0.0633…}を返します

[HypT1mean]

平均についての 1 標本 t 検定。以下を含むリストを返します（表示順）。

- 帰無仮説を棄却する場合は 0、棄却しない場合は 1
- T 検定値

- 入力 $\bar{x}$ 値
- 上側裾確率
- 自由度
- 入力 α 水準に関連付けられた上側 T 臨界値
- T 臨界値に関連付けられた統計の臨界値

HypT1mean($\bar{x}$,s,n,μ_0,α,mode)

[モード]：使用する対立仮説を指定します。

- 1：$\mu < \mu_0$
- 2：$\mu > \mu_0$
- 3：$\mu \neq \mu_0$

例：

HypT1mean(0.461368, 0.2776, 50, 0.5, 0.05, 1)は{1, -.9462…, 0.4614, 0.8277…, 1.6448…, 0.5671…}を返します

[HypT2mean]

平均についての 2 標本 T 検定。以下を含むリストを返します（表示順）。

- 帰無仮説を棄却する場合は 0、棄却しない場合は 1
- T 検定値
- $\Delta\bar{x}$ 検定値
- 上側裾確率
- 自由度
- 入力 α 水準に関連付けられた上側 T 臨界値
- T 臨界値に関連付けられた $\Delta\bar{x}$ の臨界値

HypT2mean(($\bar{x}_1$, $\bar{x}_2$,s_1,s_2,n_1,n_2,α,pooled,mode)

[プール済み]：標本がプールされるかどうかを指定します

- 0：プールされない
- 1：プールされる

[モード]：使用する対立仮説を指定します。

- 1：$\mu_1 < \mu_2$
- 2：$\mu_1 > \mu_2$
- 3：$\mu_1 \neq \mu_2$

例：

HypT2mean(0.461368, 0.522851, 0.2776, 0.2943,50, 50, 0.05, 0, 1)は{1, -1.0746…, -0.0614…, 0.8574…, 97.6674…, 1.6606…, 0.0335…}を返します

[ConfZ1mean]

平均についての 1 標本正規信頼区間。以下を含むリストを返します（表示順）。

- 下側 Z 臨界値
- 信頼区間の下側限界
- 信頼区間の上側限界

ConfZ1mean($\bar{x}$,n,σ,C)

例：

ConfZ1mean(0.461368, 50, 0.2887, 0.95)は{- 1.9599…, 0.3813…, 0.5413…}を返します

[ConfZ2mean]

2 つの平均の差についての 2 標本正規信頼区間。以下を含むリストを返します（表示順）。

- 下側 Z 臨界値
- 信頼区間の下側限界
- 信頼区間の上側限界

ConfZ2mean($\bar{x}_1$, $\bar{x}_2$,n_1,n_2,σ_1,σ_2,C)

例：

ConfZ2mean(0.461368, 0.522851, 50, 50, 0.2887, 0.2887, 0.95)は{-1.9599…, -0.1746…, 0.0516…)}を返します

[ConfZ1prop]

比率についての 1 標本正規信頼区間。以下を含むリストを返します（表示順）。

- 下側 Z 臨界値
- 信頼区間の下側限界
- 信頼区間の上側限界

ConfZ1prop(x,n,C)

例：

ConfZ1prop(21, 50, 0.95)は{-1.9599…, 0.2831…, 0.5568…}を返します

[ConfZ2prop]

2 つの比率の差についての 2 標本正規信頼区間。以下を含むリストを返します（表示順）。

- 下側 Z 臨界値
- 信頼区間の下側限界
- 信頼区間の上側限界

ConfZ2prop($\bar{x}_1$, $\bar{x}_2$,n_1,n_2,C)

例：

ConfZ2prop(21, 26, 50, 50, 0.95)は{-1.9599…, -0.2946…, 0.0946…)}を返します

[ConfT1mean]

平均についての 1 標本スチューデントの T 信頼区間。以下を含むリストを返します（表示順）。

- 自由度
- 信頼区間の下側限界
- 信頼区間の上側限界

ConfT1mean($\bar{x}$,s,n,C)

例：

ConfT1mean(0.461368, 0.2776, 50, 0.95)は{49, -.2009…, 0.5402…}を返します

[ConfT2mean]

2 つの平均の差についての 2 標本スチューデント T 検定の信頼区間 以下を含むリストを返します（表示順）。

- 自由度
- 信頼区間の下側限界
- 信頼区間の上側限界

ConfT2mean($\bar{x}_1$, $\bar{x}_2$,s_1,s_2,n_1,n_2,pooled,C)

例：

ConfT2mean(0.461368, 0.522851, 0.2887, 0.2887, 50, 50, 0.95,0)は{98.0000…, -1.9844, - 0.1760…, 0.0531…)}を返します

[Chi2GOF]

カイ 2 乗適合度検定。求められたカウント データのリスト、2 番目のリスト、および値 0 または 1 を引数として指定します。値が 0 の場合、2 番目のリストは期待される確率のリストと見なされます。値が 1 の場合、2 番目のリストは期待されるカウントのリストと見なされます。カイ 2 乗統計値、確率、および自由度を含むリストを返します。

Chi2GOF(List1, List2, Value)

例：

Chi2GOF({10,10,12,15,10,6},{.24,.2,.16,.14,.1 3,.13},0)は{10.1799..., 0.07029..., 5}を返します

[Chi2TwoWay]

カイ 2 乗 2 要因検定。カウント データの行列が与えられたときに、カイ 2 乗統計値、確率、および自由度を含むリストを返します。

Chi2TwoWay(Matrix)

例：

Chi2TwoWay([[30,35,30],[11,2,19],[43,35,35]])は{14.4302..., 0.0060..., 4}を返します

[LinRegrTConf- Slope]

傾きについての線形回帰信頼区間。説明変数データ（X）のリスト、応答変数データ（Y）のリスト、および信頼水準が与えられたときに、以下の値を以下の順で含むリストを返します。

- C：指定された信頼水準
- Critical T：指定された信頼水準に関連付けられた t の値
- DF：自由度
- β_1：線形回帰方程式の傾き
- serrSlope：傾きの標準誤差
- Lower：傾きについての信頼区間の下側限界
- Upper：傾きについての信頼区間の上側限界

```
LinRegrTConfSlope(List1, List2, C-value)
```

例：

`LinRegrTConfSlope({1,2,3,4},{3,2,0,-2},0.95)` returns {0.95, 4.302..., 2, -1.7, 0.1732..., -2.445..., -0.954...}

[LinRegrTConfInt]

切片についての線形回帰信頼区間。説明変数データ（X）のリスト、応答変数データ（Y）のリスト、および信頼水準が与えられたときに、以下の値を以下の順で含むリストを返します。

- C：指定された信頼水準
- Critical T：指定された信頼水準に関連付けられた t の値
- DF：自由度
- β_0：線形回帰方程式の切片
- serrInter：切片の標準誤差
- Lower：切片についての信頼区間の下側限界
- Upper：切片についての信頼区間の上側限界

```
LinRegrTConfInt(List1, List2, C-value)
```

例：

`LinRegrTConfInt({1, 2, 3, 4}, {3, 2, 0, - 2},0.95)` は{0.95, 4.302..., 2, 5, 0.474..., 2.959..., 7.040...}を返します

[LinRegrTMean-Resp]

平均応答についての線形回帰信頼区間。説明変数データ（X）のリスト、応答変数データ（Y）のリスト、X 値、および信頼水準が与えられたときに、以下の値を以下の順で含むリストを返します。

- X：指定された X 値
- C：指定された信頼水準
- DF：自由度

- $\hat{Y}$：指定された X 値についての平均応答
- serr $\hat{Y}$：平均応答の標準誤差
- serrInter：切片の標準誤差
- Lower：平均応答についての信頼区間の下側限界
- Upper：平均応答についての信頼区間の上側限界

```
LinRegrTMeanResp(List1, List2, X-value, Cvalue)
```

例：

`LinRegrTMeanResp({1, 2, 3, 4}, {3, 2, 0, -2}, 2.5, 0.95)` は{2.5, 0.95, 4.302..., 2, 0.75, 0.193..., -0.083, 1.583...}を返します

[LinRegrTPredInt]

将来の応答についての線形回帰予測区間。説明変数データ（X）のリスト、応答変数データ（Y）のリスト、将来の X 値、および信頼水準が与えられたときに、以下の値を以下の順で含むリストを返します。

- X：指定された将来の X 値
- C：指定された信頼水準
- DF：自由度
- $\hat{Y}$：指定された将来の X 値についての平均応答
- serr $\hat{Y}$：平均応答の標準誤差
- serrInter：切片の標準誤差
- Lower：平均応答についての予測区間の下側限界
- Upper：平均応答についての予測区間の上側限界

```
LinRegrTPredInt(List1, List2, X-value, Cvalue)
```

例：

`LinRegrTPredInt({1, 2, 3, 4}, {3, 2, 0, -2}, 2.5, 0.95)` は{2.5, 0.95, 4.302..., 2, 0.75, 0.433..., -1.113..., 2.613...}を返します

[LinRegrTTest]

線形回帰の t 検定。説明変数データ（X）のリスト、応答変数データ（Y）のリスト、および AltHyp の値が与えられたときに、以下の値を以下の順で含むリストを返します。

- T：t 値
- P：t 値に関連付けられた確率
- DF：自由度
- β_0：回帰直線の y 切片
- β_1：回帰直線の傾き
- serrLine：回帰直線の標準誤差
- serr $\hat{Y}$：平均応答の標準誤差

- serrSlope：傾きの標準誤差
- serrInter：y 切片の標準誤差
- r：相関係数
- R^2：決定係数

AltHyp の値は次のとおりです。

- $\mu<\mu_0$ の場合は AltHyp=0
- $\mu>\mu_0$ の場合は AltHyp=1
- $\mu\neq\mu_0$ の場合は AltHyp=2

例：

`LinRegrTTest({1,2,3,4}, {3,2,0,-2}, 0)` は{-9.814..., 2, 5, -1.7, 0.387..., 0.173..., 0.474..., -0.989..., 0.979...}を返します

金融アプリケーション関数

金融アプリケーションで使用される一連の関数はすべて、同じ金融アプリケーション変数セットを参照します。これらの変数は、金融アプリケーションの[数値表示]のフィールドに対応します。主な TVM 変数は 5 つで、そのうち 4 つは各関数で必須になります。その理由は、各関数が 5 番目の変数の値を小数点以下 2 桁まで求めて返すためです。`DoFinance` は、この構文規則に対する唯一の例外です。ユーザーに支払われた金額は正の数として入力し、ユーザーがキャッシュフローの一部として支払った金額は負の数として入力します。このほかに、初期値を持つオプションの変数が 3 つあります。以上の変数は、金融アプリケーション関数の引数として次の順序で現れます。

- `NbPmt`：支払い回数
- `IPYR`：年金利
- `PV`：投資またはローンの現在価値
- `PMTV`：支払い額
- `FV`：投資またはローンの将来価値
- `PPYR`：年間の支払い回数（初期値は 12）
- `CPYR`：1 年あたりの複利計算期間の数（初期値は 12）
- `BEG`：支払いを期間の始めと終わりのどちらに行うかの指定。初期値は BEG=0 です（つまり、各期間の終わりに支払い）

引数 `PPYR`、`CPYR`、および `BEG` はオプションです。指定しなかった場合、`PPYR=12`、`CPYR=PPYR`、および `BEG=0` となります。

[CalcFV]

投資またはローンの将来価値について解を求めます。

`CalcFV(NbPmt,IPYR,PV,PMTV[,PPYR,CPYR,BEG])`

例：

`CalcFV(360, 6.5, 150000, -948.10)` は-2.25 を返します

[CalcIPYR]

投資またはローンの年利率について解を求めます。

CalcIPYR(NbPmt,PV,PMTV,FV[,PPYR,CPYR, BEG])

例：

CalcIPYR(360, 150000, -948.10, -2.25)は 6.50 を返します

[CalcNbPmt]

投資またはローンの支払い回数について解を求めます。

CalcNbPmt(IPYR,PV,PMTV,FV[,PPYR,CPYR,BEG])

例：

CalcNbPmt(6.5, 150000, -948.10, -2.25)は 360.00 を返します

[CalcPMT]

投資またはローンの支払い額について解を求めます。

CalcPMT(NbPmt,IPYR,PV,FV[,PPYR,CPYR,BEG])

例：

CalcPMT(360, 6.5, 150000, -2.25)は-948.10 を返します

[CalcPV]

投資またはローンの現在価値について解を求めます。

CalcPV(NbPmt,IPYR,PMTV,FV[,PPYR,CPYR,BEG])

例：

CalcPV(360, 6.5, -948.10, -2.25)は 150000.00 を返します

[DoFinance]

TVM の結果を計算します。変数 TVMVar についての TVM の問題を解きます。変数は、金融アプリケーションのどれかの数値表示変数にする必要があります。金融アプリケーションの[数値表示]で TVMVar が強調表示された状態で 解く をタップしたときと同じ計算を実行します。

DoFinance(TVMVar)

例：

DoFinance(FV)は、金融アプリケーションの[数値表示]で FV が強調表示された状態で 解く をタップしたときと同じ方法で、投資の将来価値を返します。

線形ソルバー アプリケーション関数

線形ソルバーアプリケーションには、2x2 または 3x3 の連立一次方程式を解くことができる柔軟な機能を持つ関数が 3 つあります。

[Solve2x2]

2x2 の連立一次方程式を解きます。

```
Solve2x2(a, b, c, d, e, f)
```

次で表される連立一次方程式を解きます。

ax+by=c

dx+ey=f

[Solve3x3]

3x3 の連立一次方程式を解きます。

```
Solve3x3(a, b, c, d, e, f, g, h, i, j, k, l)
```

次で表される連立一次方程式を解きます。

ax+by+cz=d

ex+fy+gz=h

ix+jy+kz=l

[LinSolve]

連立一次方程式を解きます。行列で表される 2x2 または 3x3 の連立一次方程式を解きます。

```
LinSolve(matrix)
```

例：

`LinSolve([[A, B, C], [D, E,F]])`は、次の連立一次方程式を解きます。

ax+by=c

dx+ey=f

三角ソルバーアプリケーション関数

三角ソルバーアプリケーションには、三角形の連続する 3 つの部分（そのうちの 1 つは辺の長さである必要があります）の入力から完全な三角形を求めることができる一連の関数があります。これらのコマンドの名前には、角度を表す A および辺の長さを表す S が使用されています。これらのコマンドを使用するには、コマンド名で指定された順序で 3 つの情報を入力します。これらのコマンドはすべて、3 つの未知の値（辺の長さや角の角度）を含むリストを返します。

[AAS]

角-角-辺。2 つの角の角度と、最初の角の対辺の長さを引数として取り、2 番目の角の対辺の長さ、3 番目の辺の長さ、および 3 番目の角の角度を（この順に）含むリストを返します。

```
AAS(angle,angle,side)
```

例：

角度モードの `AAS(30, 60, 1)`は{1.732…, 2, 90}を返します

[ASA]

角-辺-角。2 つの角の角度と、間の辺の長さを引数として取り、最初の角の対辺の長さ、2 番目の角の対辺の長さ、および 3 番目の角の角度を（この順に）含むリストを返します。

```
ASA(angle,side,angle)
```

例：

角度モードの `ASA(30, 2, 60)` は{1, 1.732…, 90}を返します

[SAS]

辺-角-辺。2 つの辺の長さと間の角の角度を引数として取り、3 番目の辺の長さ、3 番目の辺の対角の角度、および 2 番目の辺の対角の角度を含むリストを返します。

```
SAS(side,angle,side)
```

例：

角度モードの `SAS(2, 60, 1)` は{1.732…, 30, 90}を返します

[SSA]

辺-辺-角。2 つの辺の長さとその間の角以外の角の角度を引数として取り、3 番目の辺の長さ、2 番目の辺の対角の角度、および 3 番目の辺の対角の角度を含むリストを返します。注：あいまいな場合、このコマンドは可能性のある 2 つの解のうちの 1 つだけを返します。

```
SSA(side,side,angle)
```

例：

`SSA(1, 2, 30)` は{1.732…, 90, 60}を返します

[SSS]

辺-辺-辺。三角形の 3 辺の長さを引数として取り、それらの対角の角度を順番に返します。

```
SSS(side,side,side)
```

例：

角度モードの `SSS(3, 4, 5)` は{36.8…, 53.1…, 90}を返します

[DoSolve]

三角ソルバーアプリケーションの現在の問題を解きます。正しい解を得るためには、三角ソルバーアプリケーションに十分なデータを入力する必要があります。。つまり、少なくとも 3 つの値を入力する必要があり、そのうちの 1 つは辺の長さである必要があります。[数値表示]の未知の値を、その表示画面に表示されている順序（左から右、上から下）で含むリストを返します。

```
DoSolve()
```

1 次エクスプローラー関数

[SolveForSlope]

傾きを求めます。2 つの点の座標(x_1, y_1)および(x_2, y_2)を入力として受け取り、その 2 点を含む直線の傾きを返します。

SolveForSlope(x_1,y_1,x_2,y_2)

例：

SolveForSlope(3,4,2,2) は 2 を返します

[SolveForYIntercept]

y 切片を求めます。点の座標(x, y)および傾き m を入力として受け取り、与えられた点を含み、与えられた傾きを持つ直線の y 切片を返します。

SolveForYIntercept(x, y, m)

例：

SolveForYIntercept(2,3,-1) は 5 を返します

2 次エクスプローラー関数

[解く]

2 次方程式を解きます。2 次方程式 $ax^2+bx+c=0$ の係数が与えられたときに、その実数解を返します。

SOLVE(a, b, c)

例：

SOLVE(1,0,-4) は{-2, 2}を返します

[DELTA]

判別式。2 次方程式 $ax^2+bx+c=0$ の係数が与えられたときに、2 次式の判別式の値を返します。

DELTA(a, b, c)

例：

DELTA(1,0,-4) は 16 を返します

共通アプリケーション関数

各アプリケーションに固有のアプリケーション関数に加え、次のアプリケーションに共通する関数が 3 つあります。これらの関数では引数として 0 から 9 の整数が使用されますが、この整数はそのアプリケーションの記号表示変数の 1 つに対応しています。

- 関数（F0 から F9）
- 解く（E0 から E9）
- 1 変数統計（H1 から H5）
- 2 変数統計（S1 から S5）
- パラメトリック（X0/Y0 から X9/Y9）
- 極座標（R0 から R9）
- 数列（U0 から U9）
- 高度なグラフ作成（V0 から V9）

[CHECK]

チェックを入れます。Digit に対応する記号表示変数にチェックを入れます（つまり選択します）。主にプログラミングで、アプリケーションの[記号表示]の定義を有効にするために使用されます。

```
CHECK(Digit)
```

例：

関数アプリケーションが現在のアプリケーションになっている場合、CHECK(1)は、関数アプリケーションの記号表示変数 F1 にチェックを入れます。その結果、[プロット表示]で F1(X) が描画され、関数アプリケーションの[数値表示]でその関数値の列が表示されます。別のアプリケーションが現在のアプリケーションになっている場合は、Function.CHECK(1) と入力する必要があります。

[UNCHECK]

チェックを外します。Digit に対応する記号表示変数のチェックを外します（つまり選択を解除します）。主にプログラミングで、アプリケーションの[記号表示]の定義を無効にするために使用されます。

```
UNCHECK(Digit)
```

例：

数列アプリケーションが現在のアプリケーションになっている場合、UNCHECK(2) は、数列アプリケーションの記号表示変数 U2 のチェックを外します。その結果、[プロット表示]で U2(N) が描画されなくなり、数列アプリケーションの[数値表示]でその値の列が表示されなくなります。別のアプリケーションが現在のアプリケーションになっている場合は、Sequence.UNCHECK(2) と入力する必要があります。

[ISCHECK]

チェックの有無を検査します。記号表示変数にチェックが入っているかどうかを検査します。変数にチェックが入っている場合は 1、入っていない場合は 0 を返します。

```
ISCHECK(Digit)
```

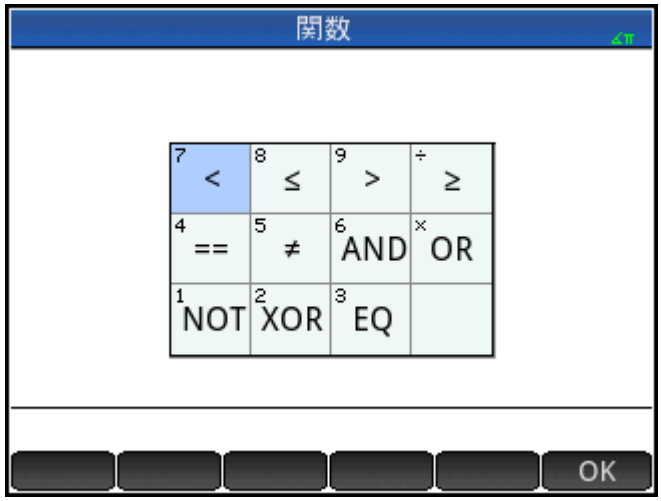
例：

関数アプリケーションが現在のアプリケーションになっている場合、ISCHECK(3) は、関数アプリケーションの[記号表示]で F3 (X) にチェックが入っているかを確認します。

[カタログ]メニュー

[カタログ]メニューには、HP Prime で利用可能な関数およびコマンドがすべて集められています。ただし、このセクションでは、[カタログ]メニューにのみ表示される関数およびコマンドについて説明します。[数学]メニューにも表示される関数およびコマンドについては、337 ページのキーボード関数を参照してください。[CAS]メニューにも表示されるものについては、354 ページの[CAS]メニューを参照してください。

[カタログ]メニュー上の一部のオプションは、関係パレット（ Shift 9 ）からも選択できます。



[!]

階乗。正の整数の階乗を返します。整数でない場合は、! = Γ(x + 1)です。これはガンマ関数を計算します。

`value!`

例：

`6!` は 720 を返します

[%]

y の x パーセントです。(x/100)*y を返します。

`%(x, y)`

例：

`%(20,50)` は 10 を返します

[%TOTAL]

パーセント合計。x に対する y のパーセントです。100*y/x を返します。

```
%TOTAL(x, y)
```

例：

```
%TOTAL(20,50)
```
は 250 を返します

[(]

開きかっこを挿入します。

[*]

乗算記号。2 つの数の積、または 2 つのベクトルのスカラー積を返します。

[+]

加算記号。2 つの数の和を返すか、2 つのリストまたは 2 つの行列の項目ごとの和を返すか、2 つの文字列を連結します。

[–]

減算記号。2 つの数の差を返すか、2 つのリストまたは 2 つの行列の項目ごとの差を返します。

[.*]

行列の項目ごとの乗算。2 つの行列の項目ごとの積を返します。

```
Matrix1.*Matrix2
```

例：

```
[[1,2],[3,4]].*[[3,4],[5,6]]
```
は[[3,8],[15,24]]を返します

[./]

行列の項目ごとの除算。2 つの行列の項目ごとの商を返します。

```
Matrix1 ./ Matrix2
```

[.^]

行列の項目ごとのべき乗。行列を項目ごとにべき乗した結果を返します。

```
Matrix .^ Integer
```

[/]

除算記号。2 つの数の商または 2 つのリストの項目ごとの商を返します。ある行列を正方行列で除算する場合は、正方行列の逆行列を左から乗算した結果を返します。

[:=]

式の評価結果を変数に格納します。グラフィックス変数 G0 から G9 では、:=は使用できません。コマンド BLIT を参照してください。

```
var:=expression
```

例：

A:=3 の場合、変数 A に値 3 が格納されます

[<]

厳密な小なり不等テスト。不等式の左辺が右辺より小さい場合は 1、そうでない場合は 0 を返します。3 つ以上のオブジェクトを比較できます。たとえば、6 < 8 < 11 は（true なので）1 を返し、6 < 8 < 3 は（false なので）0 を返します。

[<=]

小なりまたは等価の不等テスト。不等式の左辺が右辺より小さいか両辺が等しい場合は 1、そうでない場合は 0 を返します。3 つ以上のオブジェクトを比較できます。<についての前述の説明を参照してください。

[<>]

不等テスト。不等式が true の場合は 1、false の場合は 0 を返します。

[=]

等価記号。等式の 2 つの項を結び付けます。

[==]

等価テスト。左辺と右辺が等しい場合は 1、そうでない場合は 0 を返します。

[EQ]

2 つのリストが等しいことをテストします。

例：

EQ({1,2,3},{1,2,3}) は 1 を返します

[>]

厳密な大なり不等テスト。不等式の左辺が右辺より大きい場合は 1、そうでない場合は 0 を返します。3 つ以上のオブジェクトを比較できます。<についての前述の説明を参照してください。

[>=]

大なりまたは等価の不等テスト。不等式の左辺が右辺より大きいか両辺が等しい場合は 1、そうでない場合は 0 を返します。3 つ以上のオブジェクトを比較できます。<についての前述の説明を参照してください。

[^]

べき乗記号。ある数のべき乗を計算するか、行列の整数乗を計算します。

[a2q]

対称行列と、変数のベクトルが与えられたときに、ベクトルの変数を使用した行列の 2 次形式を返します。

```
a2q(Matrix, [Var1, Var2….])
```

例：

`a2q([[1,2],[4,4]],[x,y])` は x^2+6*x*y+4*y^2 を返します

[abcuv]

3 つの多項式 A、B、および C を与えると、A*U+B*V=C となる U および V を返します。最後の引数として変数を指定すると、U および V はその変数について表現されます（必要な場合）。それ以外の場合は x が使用されます

```
abcuv(PolyA, PolyB, PolyC, [Var])
```

例：

`abcuv(x^2+2*x+1,x^2-1,x+1)` は[1/2-1/2]を返します

[additionally]

プログラミングで、変数に関する追加の前提条件を定義する目的で assume と一緒に使用されます。

例：

```
assume(n,integer);
additionally(n>5);
```

[Airy Ai]

エアリー関数 w''-xw=0 の解の Ai 値を返します。

[Airy Bi]

エアリー関数 w''-xw=0 の解の Bi 値を返します。

[algvar]

式で使用されている記号変数名の行列を返します。リストは、元の式を構築するために必要な代数的拡大の順に並べられます。

```
algvar(Expr)
```

例：

`algvar(sqrt(x)+y)` は を返します

[AND]

論理積。左辺と右辺の評価結果がどちらも true の場合は 1、そうでない場合は 0 を返します。

`Expr1` **AND** `Expr2`

例：

`3 +1==4 AND 4 < 5` は 1 を返します

[append]

リストまたはベクトルの末尾に要素を追加します。

`append((List, Element)`

または

`append(Vector, Element)`

例：

`append([1,2,3],4)` は[1,2,3,4]を返します

[apply]

ベクトルまたは行列の要素に関数を適用した結果を含む、ベクトルまたは行列を返します。

`apply(Var→f(Var), Vector)` または `apply(Var→f(Var), Matrix)`

例：

`apply(x→x^3,[1 2 3])` は[1 8 27]を返します

[assume]

プログラミングで、変数に関する前提条件を定義する目的で使用されます。

`assume(Var,Expr)`

例：

`assume(n, integer)`

[basis]

与えられた行列について、行列内のベクトルのセットによって定義される線形部分空間の基底を返します。

`basis(Matrix))`

例：

`basis([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])` は[[-3,0,3],[0,-3,-6]]を返します

[betad]

ベータ確率密度関数。パラメーター α および β での、x におけるベータ分布の確率密度を計算します。

`betad(α, β, x)`

例：

betad(2.2, 1.5, 8)は 1.46143068876 を返します

[betad_cdf]

累積ベータ確率密度関数。パラメーター α および β での、値 x についてのベータ確率密度関数の下側裾確率を返します。オプションのパラメーター x_2 を使うと、ベータ確率密度関数の下側の x と x_2 の間の面積を返します

betad_cdf(α, β, x, [x_2])

例：

betad_cdf(2, 1, 0.2)は 0.04 を返します

betad_cdf(2, 1, 0.2, 0.5)は 0.21 を返します

[betad_icdf]

累積ベータ確率密度関数の逆関数。パラメーター α および β での、x についてのベータ分布の下側裾確率の確率が p になるような値 x を返します。

betad_icdf(α, β, p)

例：

betad_icdf(2,1,0.95)は 0.974679434481 を返します

[bounded_function]

limit コマンドから返される引数で、関数が有界関数であることを示します。

[breakpoint]

プログラミングで、意図的な停止または一時停止ポイントを挿入するために使用されます。

[canonical_form]

次数 2 の 3 項式を標準形で返します。

canonical_form(Trinomial,[Var])

例：

canonical_form(2*x^2-12*x+1)は 2*(x-3)^2- 17 を返します

[cat]

数列内のオブジェクトを評価し、文字列として結合して返します。

cat(Object1, Object2,…)

例：

cat("aaa",c,12*3)は"aaac36"を返します

[Cauchy]

コーシー確率密度関数。パラメーター x_0 および a での、x におけるコーシー分布の確率密度を計算します。初期設定では、x_0 = 0、a = 1 に設定されています。

`cauchy([x0], [a], x)`

例：

`cauchy(0,1,1)` は 0.159154943092 を返し、`cauchy(1)` も同じです

[Cauchy_cdf]

累積コーシー確率密度関数。パラメーター x_0 および a での、値 x についてのコーシー確率密度関数の下側裾確率を返します。オプションのパラメーター x_2 を使うと、コーシー確率密度関数の下側の x と x_2 の間の面積を返します。

`cauchy_cdf(x0, a, x, [x2])`

例：

`cauchy_cdf(0,2,2.1)` は 0.757762116818 を返します

`cauchy_cdf(0,2,2.1,3.1)` は 0.0598570954516 を返します

[Cauchy_icdf]

累積コーシー確率密度関数の逆関数。パラメーター x_0 および a での、x についてのコーシー分布の下側裾確率の確率が p になるような値 x を返します。

`cauchy_icdf(x0, a, p)`

例：

`cauchy_icdf(0, 2, 0.95)` は 12.6275030293 を返します

[cFactor]

複素数体までの範囲で式を因数分解したものを返します（2 つを超える場合はガウス整数に関して）。

`cfactor(Expr)`

例：

`cFactor(x^2*y+y)` は(x+i)*(x-i)*y を返します

[charpoly]

行列の固有多項式の係数を返します。引数が 1 つだけ指定された場合、多項式で使用される変数は x になります。2 番目の引数として変数が指定された場合、返される多項式はその変数に関するものになります。

`charpoly(Matrix,[Var])`

例：

`charpoly([[1,2],[3,4]], z)` は z^2-5*z-2 を返します

[chrem]

2 つのベクトルまたは 2 つのリストのいずれかに含まれる 2 つの整数のセットについての中国式剰余を含むベクトルを返します。

`chrem(List1, List2)` または `chrem(Vector1, Vector2)`

例：

`chrem([2,3],[7,5])` は[-12,35]を返します

[col]

行列および整数 n が与えられたときに、行列の n 番目の列をベクトルとして返します。

`col(Matrix, Integer)`

例：

`col` $\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},2\right)$ は[2,5,8]を返します

[colDim]

行列の列の数を返します。

`colDim(Matrix)`

例：

`colDim` は 3 を返します

[comDenom]

有理分数の和を 1 つの有理分数として書き換えます。1 つの有理分数の分母は元の式の有理分数の共分母となります。2 番目の引数として変数が指定された場合、分子および分母はその変数に従って作成されます。

`comDenom(Expr,[Var])`

例：

`comDenom(1/x+1/y^2+1)` は(x*y^2+x+y^2)/ (x*y^2)を返します

[companion]

多項式のコンパニオン行列を返します。

`companion(Poly,Var)`

例：

`companion(x^2+5x-7,x)` は $\left(\begin{bmatrix}0&7\\1&-5\end{bmatrix}\right)$ を返します

[compare]

2 つのオブジェクトを比較し、type(Obj1)<type(Obj2)の場合または type(Obj1)=type(Obj2)かつ Obj1<Obj2 の場合に 1 を返します。それ以外の場合は 0 を返します。

compare(Obj1, Obj2)

例：

compare(1,2) は 1 を返します

[complexroot]

多項式と実数を 2 つの引数として指定し、行列を返します。行列の各行には、多項式の複素根と多重度か、そのような根を含む区間とその多重度のいずれかが含まれます。区間は多くの場合、複素根が存在する複素平面内の長方形領域を定義します。

3 番目と 4 番目の引数として 2 つの複素数を追加した場合も、既に説明したように 2 つの引数についての行列が返されますが、2 つの複素数によって作成される対角線で定義される長方形領域に含まれる根についてのものに限定されます。

complexroot(Poly, Real, [Complex1], [Complex2])

例：

complexroot(x^3+8, 0.01) は $\begin{bmatrix} -2 & 1 \\ \left[\frac{1017-1782\cdot i}{1024} \;\; \frac{1026-1773\cdot i}{1024}\right] & 1 \\ \left[\frac{1395+378\cdot i}{512-512\cdot i} \;\; \frac{-189+702\cdot i}{256+256\cdot i}\right] & 1 \end{bmatrix}$ を返します

この行列は、x= – 2 に複素根が 1 つ存在し、別の根が 2 行目のベクトルの 2 つの値の間に存在し、3 番目の根が 3 行目のベクトルの 2 つの値の間に存在していることを示しています。

[contains]

リストまたはベクトルと、ある要素が与えられたときに、リストまたはベクトル内でその要素が最初に出現した場所のインデックスを返します。その要素がリストまたはベクトルに含まれていない場合は 0 を返します。

contains(List, Element) または contains(Vector, Element)

例：

contains({0,1,2,3},2) は 3 を返します

[CopyVar]

評価を行わずに最初の変数を 2 番目の変数にコピーします。

CopyVar(Var1,Var2)

[correlation]

リストまたは行列の要素の相関を返します。

correlation(List) または correlation(Matrix)

例：

correlation $\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 4 & 7 \end{bmatrix}$ は $\frac{33}{6\cdot\sqrt{31}}$ を返します

[count]

この関数には 2 つの用法がありますが、最初の引数は常に変数から式への写像になります。式が変数の関数である場合、この関数がベクトルまたは行列（2 番目の引数）の各要素に適用され、結果の合計が返されます。式がブール値テストの場合、ベクトルまたは行列の各要素がテストされ、テストに合格した要素の数が返されます。

`count(Var → Function, Matrix)` または `count(Var → Test, Matrix)`

例：

`count(x→x²,[1 2 3])` は 14 を返します

`count(x→ x>1, [1 2 3])` は 2 を返します

[covariance]

リストまたは行列の要素の共分散を返します。

`covariance(List)` または `covariance(Matrix)`

例：

`covariance` $\left(\begin{bmatrix}1 & 2\\ 1 & 1\\ 4 & 7\end{bmatrix}\right)$ は $\frac{11}{3}$ を返します

[covariance_correlation]

リストまたは行列の要素の共分散と相関の両方を含むベクトルを返します。

`covariance_correlation(List)` または

`covariance_correlation(Matrix)`

例：

`covariance_correlation` $\left(\begin{bmatrix}1 & 2\\ 1 & 1\\ 4 & 7\end{bmatrix}\right)$ は $\begin{bmatrix}\frac{11}{3} & \frac{33}{6\cdot\sqrt{31}}\end{bmatrix}$ を返します

[cpartfrac]

複素数体での有理分数の部分分数分解の結果を返します。

`cpartfrac(RatFrac)`

例：

`cpartfrac` $\left(\frac{x}{4-x^2}\right)$ は $-\frac{\frac{1}{2}}{x-2}-\frac{\frac{1}{2}}{x+2}$ を返します

[crationalroot]

多重度を示すことなく、多項式の複素有理数の根のリストを返します。

`crationalroot(Poly)`

例：

crationalroot(2*x^3+(-5-7*i)*x^2+ (-4+14*i)*x+8-4*i)は $\left[\frac{3+i}{2}\ 2\cdot i\ 1+i\right]$ を返します

[cumSum]

リストまたはベクトルのいずれかを引数として受け取り、要素が元の引数の累積和であるリストまたはベクトルを返します。

cumSum(List)または cumSum(Vector)

例：

cumSum([0,1,2,3,4])は[0,1,3,6,10]を返します

[DateAdd]

NbDays を Date に足し、その結果の日付を YYYY.MMDD フォーマットで返します。

[DATEADD(Date, NbDays)]

例：

DATEADD(20081228, 559)は 2010.0710 を返します

[Day of the week]

YYYY.MMDD 形式の日付を与えると、その日付と関連付けられた曜日を表す 1（月曜日）から 7（日曜日）の間の数字を返します。

DAYOFWEEK(Date)

例：

DAYOFWEEK(2006.1228)は 4（木曜日を表す）を返します

[DeltaDays]

YYYY.MMDD 形式で表現された 2 つの日付の間の日数を計算します。

DELTADAYS(Date1, Date2)

例：

DELTADAYS(2008.1228,2010.0710)は 559 を返します

[delcols]

行列および整数 n が与えられたときに、行列から n 番目の列を削除し、その結果を返します。単一の整数の代わりに 2 つの整数で決まる区間が使用された場合、その区間内の列をすべて削除し、その結果を返します。

delcols(Matrix, Integer)または delcols(Matrix, Intg1..Intg2)

例：

delcols $\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},2\right)$ は $\begin{bmatrix}1&3\\4&6\\7&9\end{bmatrix}$ を返します

[delrows]

行列および整数 n が与えられたときに、行列から n 番目の行を削除し、その結果を返します。単一の整数の代わりに 2 つの整数で決まる区間が使用された場合、その区間内の行をすべて削除し、その結果を返します。

`delrows(Matrix, Integer)` または `delrows(Matrix, Intg1..Intg2)`

例：

`delrows(`$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$`,2..3)` は $\begin{bmatrix}1&2&3\end{bmatrix}$ を返します

[deltalist]

元のリスト内の連続する項の間の差を含むリストを返します。

`deltalist(Lst)`

例：

`deltalist([1,4,8,9])` は[3,4,1]を返します

[deltalist]

元のリスト内の連続する項の間の差を含むリストを返します。

`deltalist(Lst)`

例：

`deltalist([1,4,8,9])` は[3,4,1]を返します

[Dirac]

ある実数に対するディラック デルタ関数の値を返します。

`Dirac(Real)`

例：

`Dirac(1)` は 0 を返します

[e]

数学定数 e（オイラー数）を入力します。

[egcd]

2 つの多項式 A および B が与えられたときに、次の条件を満たす 3 つの多項式 U、V、および D を返します。

`U(x)*A(x)+V(x)*B(x)=D(x)`

ここで、`D(x)=GCD(A(x),B(x))`（多項式 A および B の最大公約数）です。

多項式は記号形式で指定するか、降べきの順の係数のリストとして指定できます。

3 番目の引数がなければ、多項式は x の式と見なされます。3 番目の引数に変数が指定された場合、多項式はその変数に関する式になります。

`egcd(PolyA, PolyB, [Var])` または `egcd(ListA, ListB, [Var])`

例：

`egcd((x-1)^2,x^3-1)` は[-x-2,1,3*x-3]を返します

[eigenvals]

行列の固有値の数列を返します。

`eigenvals(Matrix)`

例：

`eigenvals` $\left(\begin{bmatrix}-2 & -2 & 1\\ -2 & 1 & -2\\ 1 & -2 & -2\end{bmatrix}\right)$ は[3 -3 3]を返します

[eigenvects]

対角化可能行列の固有ベクトルを返します。

`eigenvects(Matrix)`

例：

`eigenvects` $\left(\begin{bmatrix}-2 & -2 & 1\\ -2 & 1 & -2\\ 1 & -2 & -2\end{bmatrix}\right)$ は $\begin{bmatrix}1 & -3 & -3\\ -2 & 0 & -3\\ 1 & 3 & -3\end{bmatrix}$ を返します

[eigVl]

固有値を計算できる場合、行列に関連付けられたジョルダン行列を返します。

[EVAL]

式を評価します。

`eval(Expr)`

例：

`eval(2+3)` は 5 を返します

[evalc]

複素数の式を `real+i*imag` の形式で記述して返します。

`evalc(Expr)`

例：

`cpartfrac` $\left(\frac{1}{x+y\cdot i}\right)$ は $\frac{x}{x^2+y^2}-\frac{i\cdot y}{x^2+y^2}$ を返します

[evalf]

式と有効桁数を与えると、その与えられた有効桁数までの式の数値評価を返します。式のみ指定した場合は、CAS 設定に基づく数値評価を返します。

`evalf(Expr,[Integer])`

例：

`evalf(2/3)` は 0.666666666667 を返します

[even]

ある整数が偶数かどうかを検査します。そうである場合は 1 を、そうでない場合は 0 を返します。

例：

`even(1251)` は 0 を返します

[exact]

小数表現を分数表現または実数表現に変換します。

`exact(Expr)`

例：

`exact(1.4141)` は 14141/10000 を返します

[EXP]

数学定数 e の（式）乗に対する解を返します。

`exp(Expr)`

例：

`exp(0)` は 1 を返します

[exponential]

離散型指数確率密度関数。パラメーター k での、x における指数分布の確率密度を計算します。

`exponential(x, k)`

例：

`exponential(2.1,0.5)` は 0.734869273133 を返します

[exponential_cdf]

累積指数確率密度関数。パラメーター k での、値 x についての指数確率密度関数の下側裾確率を返します。オプションのパラメーター x_2 を使うと、指数確率密度関数の下側の x と x_2 の間の面積を返します。

`exponential_cdf(k, x, [x2])`

例：

`exponential_cdf(4.2, 0.5)` は 0.877543571747 を返します

`exponential_cdf(4.2, 0.5, 3)` は 0.122453056238 を返します

[exponential_icdf]

累積指数確率密度関数の逆関数。パラメーター k での、x についての指数分布の下側裾確率の確率が p になるような値 x を返します。

`exponential_icdf(k, p)`

例：

`exponential_icdf(4.2,0.95)` は 0.713269588941 を返します

[exponential_regression]

点のセットが与えられたときに、その点のセットに最もフィットする指数関数 y=b*a^x の係数 a および b を含むベクトルを返します。点は、2 つのリストの要素として指定することも、行列の行として指定することもできます。

`exponential_regression(Matrix)` または `exponential_regression(List1, List2)`

例：

`exponential_regression` $\left(\begin{bmatrix}1.0 & 2.0\\ 0.0 & 1.0\\ 4.0 & 7.0\end{bmatrix}\right)$ は 1.60092225473,1.10008339351 を返します

[EXPR]

文字列を数字または式に解析し、評価された結果を返します。

`EXPR(String)`

例：

`expr("2+3")` は 5 を返します

変数 X の値が 90 である場合、 `expr("X+10")` は 100 を返します

[ezgcd]

EZ GCD アルゴリズムを使用して、少なくとも 2 つの変数を持つ 2 つの多項式の最大公約数を返します。

`ezgcd(Poly1,Poly2)`

例：

`ezgcd(x^2-2*x-x*y+2*y,x^2-y^2)` は x-y を返します

[f2nd]

これ以上約分できない形式の有理分数の分子と分母で構成されるベクトルを返します。

`f2nd(RatFrac)`

例：

`f2nd` $\left(\frac{x}{x\cdot\sqrt{x}}\right)$ は $\left[1\ \sqrt{x}\right]$ を返します

[factorial]

整数の階乗を返すか、非整数に対するガンマ関数の解を返します。n が整数の場合は、factorial(n)=n! です。a が整数でない実数の場合は、factorial(a)=a! = Gamma(a + 1)です。

`factorial(Integer)` または `factorial(Real)`

例：

`factorial(4)` は 24 を返します

`factorial(1.2)` は 1.10180249088 を返します

[float]

`FLOAT_DOM` または `float` は、`assume` コマンドのオプションです。これは、`type` コマンドから返される名前でもあります。

[fMax]

x の式が与えられたときに、式の値が最大となる x の値を返します。x の式および変数が与えられたときに、式の値が最大となる変数の値を返します。

`fMax(Expr,[Var])`

例：

`fMax(-x^2+2*x+1,x)` は 1 を返します

[fMin]

x の式が与えられたときに、式の値が最小となる x の値を返します。x の式および変数が与えられたときに、式の値が最小となる変数の値を返します。

`fMin(Expr,[Var])`

例：

`fMin(x^2-2*x+1,x)` は 1 を返します

[format]

実数を、指定された形式（f=float（浮動小数点）、s=scientific（関数）、e=engineering（工学））の文字列として返します。

`format(Real, String)`

例：

`format(9.3456,"s3")` は 9.35 を返します

[Fourier a_n]

n 番目のフーリエ係数 a_n=2/T*∫(f(x)*cos(2*pi*n*x/T),a,a+T)を返します。

[Fourier b_n]

n 番目のフーリエ係数 b_n=2/T*∫(f(x)*sin(2*pi*n*x/T),a,a+T)を返します。

[Fourier c_n]

n 番目のフーリエ係数 c_n=1/T*∫(f(x)*exp(-2*i*pi*n*x/T),a,a+T)を返します。

[fracmod]

整数 n（分数を表す）および整数 p（法）が与えられたときに、n=a/b(mod p)を満たす分数 a/b を返します。

```
fracmod(Integern, Integerp)
```

例：

`fracmod(41,121)` は 2/3 を返します

[froot]

有理多項式の根および極を含むベクトルを返します。それぞれの根または極の後に、その多重度が続きます。

```
froot(RatPoly)
```

例：

froot $\left(\frac{x^5-2\cdot x^4+x^3}{x-3}\right)$ は[0 3 1 2 3 -1]を返します

[fsolve]

方程式または方程式系の数値解を返します。オプションの 3 番目の引数を使えば、解の推測を指定したり、解が存在すると予想される区間を指定したりできます。オプションの 4 番目の引数を使えば、ソルバーによって使用される反復アルゴリズムの名前を指定することができます。bisection_solver、newton_solver、または newtonj_solver を指定できます。

```
fsolve(Expr,Var,[Guess or Interval],[Method])
```

例：

`fsolve(cos(x)=x,x,-1..1,bisection_solver)` は[0.739085133215]を返します

[function_diff]

関数の導関数を（写像として）返します。

```
function_diff(Fnc)
```

例：

`function_diff(sin)` は(_x)→cos(_x)を返します

[gammad]

ガンマ確率密度関数。パラメーター a および t での、x におけるガンマ分布の確率密度を計算します。

```
gammad(a, t, x)
```

例：

`gammad(2.2,1.5,0.8)` は 0.510330619114 を返します

[gammad_cdf]

累積ガンマ分布関数。パラメーター a および t での、値 x についてのガンマ確率密度関数の下側裾確率を返します。オプションの 4 番目の引数 x_2 を使うと、2 つの x 値の間の面積を返します。

gammad_cdf(a,t,x,[x_2])

例：

gammad_cdf(2,1,2.96) は 0.794797087996 を返します

gammad_cdf(2,1,2.96,4) は 0.11362471756 を返します

[gamma_icdf]

累積ガンマ分布関数の逆関数。パラメーター a および t での、x についてのガンマ分布の下側裾確率の確率が p になるような値 x を返します。

gammad_icdf(a,t,p)

例：

gammad_icdf(2,1,0.95) は 4.74386451839 を返します

[gauss]

式および変数のベクトルが与えられたときに、ガウス アルゴリズムを使用して、ベクトルで与えられた変数の 2 乗の和または差として記述された二次形式の式を返します。

gauss(Expr,VectVar)

例：

gauss(x^2+2*a*x*y,[x,y]) は(a*y+x)^2+(- y^2)*a^2 を返します

[GF]

p^n の要素を持つ特性 p のガロア体を作成します。

GF(Integerp, Integern)

例：

GF(5,9) は GF(5,k^9-k^8+2*k^7+2*k^5-k^2+2*k- 2,[k,K,g],undef)を返します

[gramschmidt]

部分ベクトル空間の基底と、この部分ベクトル空間上のスカラー積を定義する関数が与えられたときに、その関数の正規直交基底を返します。

gramschmidt(Vector, Function)

例：

gramschmidt $\left([1\ 1+x],(p,q)\to\int_{-1}^{1}p\cdot q\,dx\right)$ は $\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1+x-1}{\frac{\sqrt{6}}{3}}\end{bmatrix}$ を返します

[hadamard]

行列のアマダール境界、または 2 つの行列の要素ごとの乗算による行列。

`hadamard(Matrix,[Matrix])`

例：

`hadamard([[1,2],[3,4]])` は 5√5 を返します

`hadamard([[1,2],[3,4]],[[3,4],[5,6]])` は[[3,8],[15,24]]を返します

[halftan2hypexp]

式の正弦、余弦、および正接を半角正接を使って書き換え、双曲線正弦、双曲線余弦、および双曲線正接を自然指数を使って書き換えた結果を返します。

`halftan_hyp2exp(ExprTrig)`

例：

`halftan_hyp2exp(sin(x)+sinh(x))` は $\frac{2\cdot\tan\left(\frac{x}{2}\right)}{\tan\left(\frac{x}{2}\right)^2+1}+\frac{\exp(x)-\frac{1}{\exp(x)}}{2}$ を返します

[halt]

プログラミングで、ステップ単位のデバッグ モードに移行するために使用されます。

[hamdist]

2 つの整数のハミング距離を返します。

`hamdist(Integer1, Integer2)`

例：

`hamdist(0x12,0x38)` は 3 を返します

[has]

変数が式に含まれている場合は 1 を返し、そうでない場合は 0 を返します。

`has(Expr,Var)`

例：

`has(x+y,x)` は 1 を返します

[head]

与えられたベクトル、シーケンス、または文字列の最初の要素を返します。

`head(Vector)` または `head(String)` または `head(Obj1, Obj2,…)`

例：

`head(1,2,3)` は 1 を返します

[Heaviside]

与えられた実数に対するヘビサイド関数の値を返します (つまり、x>=0 の場合は 1、x<0 の場合は 0)。

`Heaviside(Real)`

例：

`Heaviside(1)` は 1 を返します

[horner]

ホーナー法で計算された多項式 P(a)の値を返します。多項式は記号の式または係数のベクトルとして指定できます。

`horner(Polynomial,Real)`

例：

`horner(x^2+1,2)` は 5 を返します

`horner([1,0,1],2)` は 5 を返します

[hyp2exp]

双曲線関数項を指数関数として書き換えた式を返します。

`hyp2exp(Expr)`

例：

`hyp2exp(cosh(x))` は $\frac{\exp(x)+\frac{1}{\exp(x)}}{2}$ を返します

[iabcuv]

3 つの整数 a、b、c について au+bv=c を満たす[u,v]を返します。解が存在するためには、c が a および b の最大公約数の倍数になっている必要があることに注意してください。

`iabcuv(Intgra, Intgrb, Intgrc)`

例：

`iabcuv(21,28,7)` は[-1,1]を返します

[ibasis]

与えられた 2 つの行列について、これらを 2 つのベクトル空間と解釈し、それらの交差の基底ベクトルを返します。

`ibasis(Matrix1, Matrix2)`

例：

`ibasis` $\left(\begin{bmatrix}1&0&0\\0&1&0\end{bmatrix},\begin{bmatrix}1&1&1\\0&0&1\end{bmatrix}\right)$ は[-1, -1, 0]を返します

[icontent]

多項式の整数係数の最大公約数を返します。

icontent(Poly,[Var])

例：

icontent(24x^3+6x^2-12x+18)は 6 を返します

[id]

引数に対する恒等関数の解を含むベクトルを返します。

id(Object1, [Object2,…])

例：

id([1 2], 3, 4)は[[1 2] 3 4]を返します

[identity]

整数 n が与えられたときに、次元 n の恒等行列を返します。

identity(Integer)

例：

identity(3)は $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ を返します

[iegcd]

2 つの整数の拡張最大公約数を返します。

iegcd(Integer1, Integer2)

例：

iegcd(14, 21)は[-1, 1, 7]を返します

[igcd]

2 つの整数、2 つの有理数、またはいくつかの変数を含む 2 つの多項式の最大公約数を返します。

igcd(Integer1, Integer2)または igcd(Ratnl1, Ratnl2)または igcd(Poly1, Poly2)

例：

igcd(24, 36)は 12 を返します

igcd(2/3,3/4)は 1/12 を返します

[image]

行列の線形適用の像。

image(Matrix)

例：

image([[1,2],[3,6]])は[1,3]を返します

[interval2center]

区間の中心を返します。

`interval2center(Interval)`

例：

`interval2center(2..5)` は 7/2 を返します

[inv]

式または行列の逆数（逆行列）を返します。

`inv(Expr)` または `inv(Matrix)`

例：

`inv(9/5)` は 5/9 を返します

[iPart]

実数から小数部を除去して返すか、リスト内の各実数から小数部を除去して返します。

`iPart(Real)` または `iPart(List)`

例：

`iPart(4.3)` は 4 を返します

[iquorem]

2 つの整数に関するユークリッド除法の商と剰余を返します。

`iquorem(Integer1, Integer2)`

例：

`iquorem(63, 23)` は[2, 17]を返します

[jacobi_symbol]

行列の線形応用の核を返します。

`jacobi_symbol(Integer1, Integer2)`

例：

`jacobi_symbol(132,5)` は-1 を返します

[ker]

与えられた整数のヤコビ記号を返します。

`ker(Matrix)`

例：

`ker([[1 2], [3 6]])` は[2 1]を返します

[laplacian]

変数のベクトルに関する式のラプラシアンを返します。

`laplacian(Expr, Vector)`

例：

`laplacian(exp(z)*cos(x*y),[x,y,z])` は – x^2*cos(x*y)*exp(z)- y^2*cos(x*y)*exp(z) +cos(x*y)*exp(z)を返します

[latex]

評価した CAS 式を Latex フォーマットで返します。

`latex(Expr)`

例：

`latex(1/2)` は"\frac{1}{2}"を返します

`latex((x^4-1)/(x^2+3)` は"\frac{(x^{4}-1)}{(x^{2}+3)}"を返します

[lcoeff]

多項式の最高次数の項の係数を返します。多項式は、記号形式で表現することも、リストとして表現することもできます。

`lcoeff(Poly)` または `lcoeff(List)` または `lcoeff(Vector)`

例：

`lcoeff(-2*x^3+x^2+7*x)` は-2 を返します

[legendre_symbol]

単一の整数 n が指定された場合、次数 n のルジャンドル多項式を返します。2 つの整数が指定された場合、最初の整数を次数に持つルジャンドル多項式を使って、2 番目の整数のルジャンドル記号を返します。

`legendre_symbol(Integer1, [Integer2])`

例：

`legendre(4)` は 35*x^4/8+-15*x^2/4+3/8 を返しますが、legendre(4,2)は簡略化後の 443/8 を返します

[length]

リスト、文字列、またはオブジェクトのセットの長さを返します。

`length(List)` または `length(String)` または `length(Object1, Object2,…)`

例：

`length([1,2,3])` は 3 を返します

[lgcd]

リストまたはベクトルに含まれているか、引数として直接入力された整数または多項式のセットの最大公約数を返します。

`lgcd(List)`または`lgcd(Vector)`または`lgcd(Integer1, Integer2, …)`または`lgcd(Poly1, Poly2, …)`

例：

`lgcd([45,75,20,15])`は 5 を返します

[lin]

式の指数を線形化して返します。

`lin(Expr)`

例：

`lin((exp(x)^3+exp(x))^2)`は exp(6*x)+2*exp(4*x)+exp(2*x)を返します

[linear_interpolate]

2 行からなる行列で定義される折れ線から規則的な標本を取得します。

`linear_interpolate(Matrix,Xmin,Xmax,Xstep)`

例：

`linear_interpolate([[1,2,6,9],[3,4,6,7]],1,9, 1)`は[[1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0], [3.0,4.0,4.5,5.0,5.5,6.0,6.33333333333,6.6666 6666667,7.0]]を返します

[linear_regression]

点のセットが与えられたときに、その点のセットに最もフィットする一次関数 y=a*x+b の係数 a および b を含むベクトルを返します。点は、2 つのリストの要素として指定することも、行列の行として指定することもできます。

`linear_regression(Matrix)`または`linear_regression(List1, List2)`

例：

`linear_regression` $\left(\begin{bmatrix}1.0 & 2.0\\ 0.0 & 1.0\\ 4.0 & 7.0\end{bmatrix}\right)$ は[1.53…, 0.769…]を返します

[LineHorz]

[幾何]アプリケーションの[記号表示]で使用します。実数 a または実数 a に評価される式を与えると、水平な線 y=a を引きます。

`LineHorz(Exp)`または`LineHorz(Real)`

例：

`LineHorz(-1)`は方程式 y = -1 となる線を引きます。

[LineTan]

Var=Value における f(Var)の接線を引きます。

`LineTan(f(Var), [Var], Value)`

例：

`LineTan(x² - x, 1)` は直線 y=x-1 を引きます。つまり、x=1 における y= x² − x への接線です

[LineVert]

[幾何]アプリケーションの[記号表示]で使用します。実数 a または実数 a に評価される式を与えると、垂直な線 x=a を引きます。

`LineVert(Expr)` または `LineVert(Real)`

例：

`LineVert(2)` は方程式 x=2 となる線を引きます

[list2mat]

リストを n 個の項を含むいくつかの行に分割することで、n 列の行列を返します。リストの要素数が n で割り切れない場合、行列はゼロで補完されます。

`list2mat(List, Integer)`

例：

`list2mat({1,8,4,9},1)` は $\begin{bmatrix}1\\8\\4\\9\end{bmatrix}$ を返します

[lname]

式の変数のリストを返します。

`lname(Expr)`

例：

`lname(exp(x)*2*sin(y))` は[x,y]を返します

[lnexpand]

対数式の展開形式を返します。

`lnexpand(Expr)`

例：

`lnexpand(ln(3*x))` は ln(3)+ln(x)を返します

[logarithmic_regression]

点のセットが与えられたときに、その点のセットに最もフィットする自然対数関数 y=a*ln(x)+b の係数 a および b を含むベクトルを返します。点は、2 つのリストの要素として指定することも、行列の行として指定することもできます。

logarithmic_regression(Matrix)または logarithmic_regression(List1, List2)

例：

logarithmic_regression $\begin{bmatrix} 1.0 & 1.0 \\ 2.0 & 4.0 \\ 3.0 & 9.0 \\ 4.0 & 9.0 \end{bmatrix}$ は[6.3299…, 0.7207…]を返します

[logb]

a の底 b の対数を返します。

logb(a,b)

例：

logb(5,2) は ln(5)/ln(2)（約 2.32192809489）を返します

[logistic_regression]

y、y'、C、y'max、xmax、および R を返します。ここで、y は y(x0)=y0 を満たすロジスティク関数（y'/y=a*y+b の解）であり、[y'(x0),y'(x0+1)...]はリスト L の要素で形成される直線の最適な近似です。

logistic_regression(Lst(L),Real(x0),Real(y0))

例：

logistic_regression([0.0,1.0,2.0,3.0,4.0],0.0 ,1.0)は[-17.77/(1+exp(-0.496893925384*x+2.82232341488+3.14159265359* i)),-2.48542227469/(1+cosh(- 0.496893925384*x+2.82232341488+3.14159265359* i))]を返します

[lu]

数値的な行列 A について、PA=LU を満たす置換 P、L、および U を返します。

lu(Matrix)

例：

lu([1 2],[3 4])は[[1 2] [[1 0],[3 1]] [[1 2], [0 -2]]]を返します

[lvar]

式を与えると、変数を使用する式（変数自体の出現も含む）の関数リストを返します。

lvar(Expr)

例：

lvar(e^(x)*2*sin(y) + ln(x))は[e^(x) sin(y) ln(x)]を返します

[map]

この関数には 2 つの用法がありますが、2 番目の引数は常に変数から式への写像になります。式が変数の関数である場合、その関数がベクトルまたは行列（最初の引数）の各要素に適用され、結果のベクトルまたは行列が返されます。式がブール値テストの場合、ベクトルまたは行列の各要素がテストされ、結果がベクトルまたは行列として返されます。各テストでは 0（失敗）または 1（成功）が返されます。

map(Matrix, Var → Function)または map(Matrix, Var → Test)

例：

map([1 2 3], x→x^3)は[1 8 27]を返します

map([1 2 3], x→ x>1)は[0 1 1]を返します

[mat2list]

行列の要素を含むベクトルを返します。

mat2list(Matrix)

例：

mat2list([[1 8],[4 9]])は[1 8 4 9]を返します

[matpow]

行列および整数 n が与えられたときに、ジョルダン標準形を使って行列の n 乗を返します。

matpow(Matrix, Integer)

例：

matpow([[1,2],[3,4]],n)は[[(sqrt(33)- 3)*((sqrt(33)+5)/2)^n*-6/(-12*sqrt(33))+(- (sqrt(33))-3)*((-(sqrt(33))+5)/2)^n*6/(- 12*sqrt(33)),(sqrt(33)-3)*((sqrt(33)+5)/ 2)^n*(-(sqrt(33))-3)/(-12*sqrt(33))+(-(sqrt(33))-3)*((-(sqrt(33))+5)/2)^n*(- (sqrt(33))+3)/(- 12*sqrt(33))],[6*((sqrt(33)+5)/2)^n*-6/(- 12*sqrt(33))+6*((-(sqrt(33))+5)/2)^n*6/(- 12*sqrt(33)),6*((sqrt(33)+5)/2)^n*(- (sqrt(33))-3)/(-12*sqrt(33))+6*((-(sqrt(33))+5)/2)^n*(-(sqrt(33))+3)/(- 12*sqrt(33))]]を返します

[matrix]

2 つの整数 p および整数 q が与えられたときに、ゼロが設定された p 行 q 列の行列を作成します。値が 3 番目の引数として指定された場合、その値が設定された行列を返します。j および k を使った写像が指定された場合は、その写像に基づいて行列を設定します（j が現在の行、k が現在の列）。この関数は、apply コマンドと一緒に使用することもできます。

matrix(p, q, [Value or Mapping(j,k)])

例：

matrix(1,3,5)は[5 5 5]を返します

[MAXREAL]

HP Prime 電卓のホーム表示および[CAS 表示]で表示できる最大の実数を返します。[CAS 表示]では MAXREAL=1.79769313486*10^308、ホーム表示では MAXREAL=9.99999999999E499 になります

[mean]

リストの算術平均を返します（オプションのリストは重みのリストとして使用）。引数が行列の場合、列の平均を返します。

mean(List1, [List2])または mean(Matrix)

例：

`mean([1,2,3],[1,2,3])` は 7/3 を返します

[median]

リストの中央値を返します（オプションのリストは重みのリストとして使用）。引数が行列の場合、列の中央値を返します。

`median(List1, [List2])` または `median(Matrix)`

例：

`median([1,2,3,5,10,4])` は 3.5 を返します

[member]

リストまたはベクトルと、ある要素が与えられたときに、リストまたはベクトル内でその要素が最初に出現した場所のインデックスを返します。要素がリストまたはベクトルに含まれていない場合は 0 を返します。contains に似ていますが、引数の順序として要素が先になっている点が異なります。

`member(( Element, List)` または `contains(Element, Vector)`

例：

`member(2, {0,1,2,3})` は 3 を返します

[MEMORY]

メモリおよびストレージ容量を表す整数が含まれたリストか、メモリ（n=1）またはストレージ容量（n=2）を表す単一の整数のどちらかを返します。

MEMORY()

MEMORY(n)

[MINREAL]

HP Prime 電卓のホーム表示および[CAS 表示]で表示できる最小の実数（ゼロに近い）を返します。

[CAS 表示]では、MINREAL=2.22507385851*10^{-308} です。

ホーム表示では、MINREAL=1 E-499 です

[modgcd]

モジュール アルゴリズムを使用して、2 つの多項式の最大公約数を返します。

`modgcd(Poly1,Poly2)`

例：

`modgcd(x^4-1,(x-1)^2)` は x-1 を返します

[mRow]

式、行列、および整数 n が与えられたときに、行列の行 n に式を乗算します。

`mRow(Expr, Matrix, Integer)`

例：

$\mathrm{mRow}\left(12, \begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}, 1\right)$は$\begin{bmatrix}12 & 24\\ 3 & 4\\ 5 & 6\end{bmatrix}$を返します

[mult_c_conjugate]

与えられた複素数の式に複素数の分母が含まれている場合、その分母の共役複素数を分子と分母の両方に乗算した後、式を返します。与えられた複素数の式に複素数の分母が含まれていない場合、分子の共役複素数を分子と分母の両方に乗算した後、式を返します。

```
mult_c_conjugate(Expr)
```

例：

$\mathrm{mult_c_conjugate}\left(\frac{1}{3+2\cdot i}\right)$は$\frac{1\cdot(3+2\cdot -i)}{(3+2\cdot i)\cdot(3+2\cdot -i)}$を返します

[mult_conjugate]

分子または分母に平方根が含まれている式を受け取ります。分母に平方根が含まれている場合、分母の共役数を分子と分母の両方に乗算した後、式を返します。分母に平方根が含まれていない場合、分子の共役数を分子と分母の両方に乗算した後、式を返します。

```
mult_conjugate(Expr)
```

例：

$\mathrm{mult_conjugate}(\sqrt{3}-\sqrt{2})$は$\frac{(\sqrt{3}-\sqrt{2})\cdot(\sqrt{3}+\sqrt{2})}{\sqrt{3}+\sqrt{2}}$を返します

[nDeriv]

式、微分の変数、および実数 h が与えられたときに、`f’(x)=(f(x+h)-f(x+h))/(2*h)`を使用して式の導関数の近似値を返します。

3 番目の引数がない場合、h の値は 0.001 に設定されます。3 番目の引数が実数の場合、それが h の値になります。3 番目の引数が変数の場合、上記の式で h の代わりにその変数を使用したものが返されます。

`nDeriv(Expr,Var, Real)`または`nDeriv(Expr, Var1, Var2)`

例：

`nDeriv(f(x),x,h)`は(f(x+h)-(f(x-h)))*0.5/h を返します

[NEG]

単項マイナス。負の符号を入力します。

[negbinomial]

負の二項確率密度関数。パラメーター n および k での、x における負の二項分布の確率密度を計算します。

```
negbinomial(n, k, x)
```

例：

`negbinomial(4, 2, 0.6)` は 0.20736 を返します

[negbinomial_cdf]

負の二項分布についての累積確率密度関数。パラメーター n および k での、値 x についての負の二項確率密度関数の下側裾確率を返します。オプションのパラメーター x_2 を使うと、負の二項確率密度関数の下側の x と x_2 の間の面積を返します。

`negbinomial_cdf(n, k, x, [x2])`

例：

`negbinomial_cdf(4, 0.5, 2)` は 0.34375 を返します

`negbinomial_cdf(4, 0.5, 2, 3)` は 0.15625 を返します

[negbinomial_icdf]

負の二項分布についての累積確率密度関数の逆関数。パラメーター n および k での、x についての負の二項分布の下側裾確率の確率が p になるような値 x を返します。

`negbinomial_icdf(n, k, p)`

例：

`negbinomial_icdf(4, 0.5, 0.7)` は 5 を返します

[newton]

ニュートン法を使用して関数の根を求めます。Guess で開始して Integer（整数）回の反復計算を行います。初期設定では、Integer は 20 です。

`newton(Expr,Var, [Guess],[Integer])`

例：

`newton(3-x^2,x,2)` は 1.73205080757 を返します

[normal]

式を展開された既約形式で返します。

`normal(Expr)`

例：

`normal(2*x*2)` は 4*x を返します

[normalize]

ベクトルが与えられた場合、それを L_2 ノルムで除算した結果を返します（L2 ノルムはベクトルの座標の 2 乗和の平方根）。

複素数が与えられた場合、その複素数を絶対値で除算した結果を返します。

`normalize(Vector)` または `normalize(Complex)`

例：

normalize(3+4*i) は(3+4*i)/5 を返します

[NOT]

ブール式の論理否定を返します。

not(Expr)

[odd]

与えられた整数が奇数の場合は 1 を返し、そうでない場合は 0 を返します。

odd(Integer)

例：

odd(6) は 0 を返します

[OR]

論理和。いずれかの辺または両辺の評価結果が true の場合は 1、それ以外の場合は 0 を返します。

Expr1 または Expr2

例：

3 +1==4 OR 8 < 5 は 1 を返します

[order_size]

級数展開の剰余（0 項）を返します。a>0 の場合、limit(x^a*order_size(x),x=0)=0。

order_size(Expr)

[pa2b2]

4 を法として 1 と合同な素数の整数 n を受け取り、a^2+b^2=n を満たす[a,b]を返します。

pa2b2(Integer)

例：

pa2b2(17) は[4 1]を返します

[pade]

式のパデ近似を返します。つまり、P の次数<p で、P/Q=Expr mod x^(n+1)または mod N となる有理分数 P/Q です。

pade(Expr, Var, Integern, Integerp)

例：

pade(exp(x), x, 5, 3) は $\frac{-3 \cdot x^2 - 24 \cdot x - 60}{x^3 - 9 \cdot x^2 + 36 \cdot x - 60}$ を返します

[part]

式の n 番目の部分式を返します。

part(Expr, Integer)

例：

part(sin(x)+cos(x),1) は sin(x)を返します

part(sin(x)+cos(x),2) は cos(x)を返します

[peval]

係数のベクトルで定義される多項式と実数 n が与えられたときに、その値において多項式を評価します。

peval(Vector, Value)

例：

peval([1,0,-2],1) は-1 を返します

[PI]

π を挿入します。

[PIECEWISE]

区分定義関数を定義するために使用されます。条件および式から成るペアを引数として受け取ります。これらの各ペアは、区分関数のサブ関数、およびそのサブ関数が有効になる定義域を定義します。

$$\text{PIECEWISE}\begin{cases}\text{Case1 } if \text{ Test1}\\ \text{Case2 } if \text{ Test2}\\ \cdots\end{cases}$$

例：

$$\text{PIECEWISE}\begin{cases}-x \text{ if } x<0\\ x^2 \text{ if } x\geq 0\end{cases}$$

[エントリ]設定が[テキストブック]に設定されていない場合は構文が変わることに注意してください。

PIECEWISE(Case1, Test1, ...[Casen, Testn])

[plotinequation]

2 変数を持つ不等式の解のグラフを表示します。

plotinequation(Expr,[x=xrange,y=yrange],[xstep],[ystep])

[polar_point]

極形式の点の半径および角度が与えられたときに、複素形式の直交座標を使用して点を返します。

polar_point(Radius, Angle)

例：

polar_point(2, π/3) は点 $\left(2\cdot\left(\frac{1}{2}+\frac{i\cdot\sqrt{3}}{2}\right)\right)$ を返します

[pole]

円および直線が与えられたときに、その直線が円の極線になるような点を返します。

`pole(Crcle,Line)`

例：

`pole(circle(0, 1), line(1+i, 2))` は point(1/2,1/2)を返します

[POLYCOEF]

ベクトル引数またはリスト引数に指定された根を持つ多項式の係数を返します。

`POLYCOEF(Vector)` または `POLYCOEF(List)`

例：

`POLYCOEF({-1, 1})` は{1, 0, -1}を返します

[POLYEVAL]

係数のベクトルまたはリスト、および値が指定されたときに、それらの係数によって指定される多項式を指定された値で評価します。

`POLYEVAL(Vector, Value)` または `POLYEVAL(List, Value)`

例：

`POLYEVAL({1,0,-1},3)` は 8 を返します

[polygon]

リストの各要素を頂点に持つ多角形を描画します。

`polygon(Point1, Point2, …, Pointn)`

例：

`polygon(GA,GB,GD)` は ΔABD を描画します

[polygonplot]

幾何アプリケーションの[記号表示]で使用されます。n × m 行列が与えられたときに、点(xk, yk)を描画して各点を接続します。ここで xk は行 k 列 1 の要素、yk は行 k 列 j の要素です（k が 1 から n 行まで変化する間 j は固定）。したがって、それぞれの列のペアによって独自の図が生成され、m − 1 個の図が得られます。

`polygonplot(Matrix)`

例：

`polygonplot` $\left(\begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 1 \\ -1 & 2 & 3 \end{bmatrix}\right)$ は 2 つの図を描画します（それぞれ 3 点が線分で結ばれる）。

[polygonscatterplot]

幾何アプリケーションの[記号表示]で使用されます。n × m 行列が与えられたときに、点(xk, yk)を描画して各点を接続します。ここで xk は行 k 列 1 の要素、yk は行 k 列 j の要素です（k が 1 から n 行まで

変化する間 j は固定)。したがって、それぞれの列のペアによって独自の図が生成され、m – 個の図が得られます。

```
polygonscatterplot(Matrix)
```

例：

`polygonscatterplot` $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 0 & 1 \\ -1 & 2 & 3 \end{pmatrix}$ は 2 つの図を描画します（それぞれ 3 点が線分で結ばれる)。

[polynomial_regression]

2 つのリストで定義される点のセット、および正の整数 n が与えられたときに、与えられた点に最も良く当てはまる n 次多項式 $y = a_n*x^n + a_{n-1}x^{n-1} + \ldots a_1*x + a_0$ の係数（a_n、$a_{n-1} \ldots a_0$）を含むベクトルを返します。

```
polynomial_regression(List1, List2, Integer)
```

例：

`polynomial_regression({1, 2, 3, 4}, {1, 4, 9, 16},3)` は[0 1 0 0]を返します

[POLYROOT]

係数のベクトルとして与えられた多項式のゼロ点を返します。

```
POLYROOT(Vector)
```

例：

`POLYROOT([1 0 -1])` は{-1, 1}を返します

[potential]

その勾配が、ベクトルおよび変数ベクトルによって定義されるベクトル場である関数を返します。

```
potential(Vector1, Vector2)
```

例：

`potential([2*x*y+3,x^2-4*z,-4*y],[x,y,z])` は x2*y+3*x-4*y*z を返します

[power_regression]

2 つのリストで定義される点のセットが与えられたときに、与えられた点に最も当てはまる単項式 y=b*x^m の係数 m および b を含むベクトルを返します。

```
power_regression(List1, List2)
```

例：

`power_regression({1, 2, 3, 4}, {1, 4, 9, 16})` は[2 1]を返します

[powerpc]

円および点が与えられたときに、実数 d2 – r2（d は点から円の中心まで距離、r は円の半径）を返します。

```
powerpc(Circle, Point)
```

例：

powerpc(circle(0,1+i),3+i) は 8 を返します

[prepend]

リストまたはベクトルの先頭に要素を追加します。

prepend(List, Element) または prepend(Vector, Element)

例：

prepend([1,2],3) は[3,1,2]を返します

[primpart]

多項式をその係数の最大公約数で除算した結果を返します。

primpart(Poly,[Var])

例：

primpart(2x^2+10x+6) は x^2+5*x+3 を返します

[product]

最初の引数として式が指定された場合、与えられたステップで式の変数が最小値から最大値まで変化するときの解の積を返します。ステップが指定されない場合、ステップは 1 として解釈されます。

最初の引数としてリストが指定された場合、リストの値の積が返されます。

最初の引数として行列が指定された場合、行列の要素ごとの積が返されます。

product(Expr, Var, Min, Max, Step) または product(List) または product(Matrix)

例：

product(n,n,1,10,2) は 945 を返します

[propfrac]

分数または有理分数 A/B を Q+R/B に簡略化して返します（R<B、または R の次数が B の次数より小さい）。

propfrac(Fraction) または propfrac(RatFrac)

例：

propfrac(28/12) は 2+1/3 を返します

[ptayl]

多項式 P と値 a が与えられたときに、P(x)=Q(x − a)を満たすテイラー多項式 Q を返します。

ptayl(Poly, Value, [Var])

例：

ptayl(x^2+2*x+1,1) は x^2+4*x+4 を返します

[purge]

[CAS 表示]での変数名の割り当てを解除します。

たとえば、f が定義されている場合、purge(f)はその定義を削除し、f を記号状態に戻します。

`purge(Var)`

[Q2a]

二次形式と変数ベクトルが与えられたときに、与えられた変数についての二次形式の行列を返します。

`q2a(Expr, Vector)`

例：

`q2a(x^2+2*x*y+2*y^2,[x,y])` は $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ を返します

[quantile]

リストまたはベクトル、および 0 から 1 までの間の分位値が与えられたときに、リストまたはベクトルの要素の対応する分位数を返します。

`quantile(List, Value)` または `quantile(Vector, Value)`

例：

`quantile([0,1,3,4,2,5,6],0.25)` は 1 を返します

[quartile1]

リストまたはベクトルが与えられた場合、リストまたはベクトルの要素の第 1 分位数を返します。行列が与えられた場合、行列の列の第 1 分位数を返します。

`quartile1(List)` または `quartile1(Vector)` または `quartile1(Matrix)`

例：

`quartile1([1,2,3,5,10,4])` は 2 を返します

[quartile3]

リストまたはベクトルが与えられた場合、リストまたはベクトルの要素の第 3 分位数を返します。行列が与えられた場合、行列の列の第 3 分位数を返します。

`quartile3(List)` または `quartile3(Vector)` または `quartile3(Matrix)`

例：

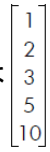
`quartile3([1,2,3,5,10,4])` は 5 を返します

[quartiles]

リストまたはベクトルの要素の最小値、第 1 分位数、中央値、第 3 分位数、および最大値を含む行列を返します。行列が引数として指定された場合、行列の列の、5 つの数値の集計結果を返します。

`quartiles(List)` または `quartiles(Vector)` または `quartiles(Matrix)`

例：

`quartiles([1,2,3,5,10,4])` は  を返します

[quorem]

2 つの多項式（記号形式で直接表現されたか係数のベクトルとして表現されたもの）のユークリッド除法の商および剰余を返します。多項式が係数のベクトルとして表現された場合、このコマンドは類似の商ベクトルおよび剰余ベクトルを返します。

`quorem(Poly1, Poly2)` または `quorem(Vector1, Vector2)`

例：

`quorem(x^3+2*x^2+3*x+4,-x+2)` は[-x^2-4*x- 11, 26]を返します

`quorem([1,2,3,4],[-1,2])` は[[-1, -4, -11] [26]]を返します

[QUOTE]

式を評価しないまま返します。

`quote(Expr)`

[randbinomial]

それぞれ成功確率が p の試行 n 回での二項分布に従う乱数を返します。

`randbinomial(n, p)`

例：

`randbinomial(10, 0.4)` は 0 と 10 の間の整数を返します

[randchisquare]

自由度 n でのカイ 2 乗分布による乱数を返します。

`randchisquare(n)`

例：

`randchisquare(5)` は自由度 5 でのカイ 2 乗分布から正の実数を返します

[randexp]

正の実数が与えられたときに、実数 a>0 での指数分布に従うランダムな実数を返します。

`randexp(Real)`

[randfisher]

分子の自由度 n、分母の自由度 d の F 分布から乱数を返します。

`randfisher(n, d)`

例：

`randfisher(5, 2)` は分子の自由度 5、分母の自由度 2 の F 分布から実数を返します

[randgeometric]

成功確率が p の幾何分布から乱数を返します。

`randgeometric(p)`

例：

`randgeometric(0.4)` は成功確率 0.4 の幾何分布から正の整数を返します

[randperm]

正の整数が与えられたときに、[0,1,2,...,n – 1]のランダムな順列を返します。

`randperm(Intg(n))`

例：

`randperm(4)` は、ベクトル[0 1 2 3]の要素のランダムな順列を返します

[randpoisson]

パラメーター k でのポワソン分布から乱数を返します。

`randpoisson(k)`

例：

`randpoisson(5.4)`

[randstudent]

自由度 n のスチューデントの t 分布から乱数を返します。

`randstudent(n)`

例：

`randstudent(5)`

[randvector]

1 つの整数 n が指定された場合、-99 から 99 の範囲で均一に分布するランダム整数を含むサイズ n のベクトルを返します。オプションの 2 番目の整数 m が指定された場合、(0, m]の範囲の整数が設定されたベクトルを返します。オプションの区間が第 2 引数として指定された場合、その区間内の実数をベクトルに設定します。

`randvector(n, [m または p..q])`

[ranm]

1 つの整数 n が指定された場合、[-99, 99]の範囲で均一に分布するランダム整数を含むサイズ n のベクトルを返します。2 つの整数 n および m が指定された場合、nxm 行列を返します。最後の引数として区間が指定された場合、その区間に限定されたランダムな実数を要素に持つベクトルまたは行列を返します。

[ratnormal]

これ以上約分できない有理分数として式を書き換えます。

`ratnormal(Expr)`

例：

`ratnormal` $\left(\frac{x^2-1}{x^3-1}\right)$ は $\frac{x+1}{x^2+x+1}$ を返します

[rectangular_coordinates]

点の極座標を含むベクトルが与えられたときに、点の直交座標を含むベクトルを返します。

`rectangular_coordinates(Vector)`

例：

`rectangular_coordinates([1, π/4])` は $\left[\frac{\sqrt{2}}{2}\ \frac{\sqrt{2}}{2}\right]$ を返します

[reduced_conic]

円錐式を指定し、以下の項目を持つベクトルを返します。

- 円錐の原点
- 円錐が退化した基礎の行列
- 0 または 1（円錐が縮退の場合は 0）
- 円錐の誘導方程式
- 円錐のパラメトリック方程式のベクトル

`reduced_conic(Expr, [Vector])`

例：

`reduced_conic(x^2+2*x-2*y+1)` は次を返します

$$\left[\left[-1\ \ 0\right]\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}1\,y^2+2\cdot x\left[-1+-i\cdot\left(-\frac{1}{2}\cdot x\cdot x+i\cdot x\right)\ x-4\,4\,0.1\,x^2+2\cdot x-2\cdot y+1-1+(-i)\cdot\left(\frac{-1}{2}\cdot x\cdot x+(i)\cdot x\right)\right]\right]$$

[ref]

行列についてガウスの消去法を実行します。

`ref(Matrix)`

例：

`ref` $\begin{bmatrix}3 & 1 & -2\\ 3 & 2 & 2\end{bmatrix}$ は $\begin{bmatrix}1 & \frac{1}{3} & \frac{-2}{3}\\ 0 & 1 & 2\end{bmatrix}$ を返します

[remove]

ベクトルまたはリストが与えられたときに、出現した Value を削除するか、Test が真になる値を削除し、結果のベクトルまたはリストを返します。

remove(Value, List)または remove(Test, List)

例：

remove(5,{1,2,5,6,7,5})は{1,2,6,7}を返します

remove(x→x≥5, [1 2 5 6 7 5])は[1 2]を返します

[reorder]

式と、変数のベクトルが与えられたときに、ベクトルに指定された順序に従って式の変数を並べ替えます。

reorder(Expr, Vector)

例：

reorder($x^2+2*x+y^2$,[y,x])は y^2+x^2+2*x を返します

[residue]

ある値での式の留数を返します。

residue(Expr, Var, Value)

例：

residue(1/z,z,0)は 1 を返します

[restart]

すべての変数を除去します。

restart(NULL)

[resultant]

2 つの多項式の終結式（つまり、シルベスター行列の行列式）を返します。

resultant(Poly1, Poly2, Var)

例：

resultant(x^3+x+1, x^2-x-2,x)は-11 を返します

[revlist]

リストまたはベクトルの要素の順序を反転します。

revlist(List)または revlist(Vector)

例：

revlist([1,2,3])は[3,2,1]を返します

[romberg]

ロンベルグの方法を使用して、定積分の近似値を返します。

romberg(Expr, Var, Val1, Val2)

例：

`romberg(exp(x^2),x,0,1)` は 1.46265174591 を返します

[row]

行列とおよび整数 n が指定された場合、行列の行 n を返します。行列および区間が指定された場合、その区間で示される行列の行を含むベクトルを返します。

`row(Matrix, Integer)` または `row(Matrix, Interval)`

例：

$\mathrm{row}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},2\right)$ は[4 5 6]を返します

[rowAdd]

行列および 2 つの整数が与えられたときに、与えられた行列の 2 番目の整数で指定される行を、2 つの整数で指定される行の和で置き換えることによって得られる行列を返します。

`rowAdd(Matrix, Integer1, Integer2)`

例：

$\mathrm{rowAdd}\left(\begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix},1,2\right)$ は $\begin{bmatrix}1&2\\4&6\\5&6\end{bmatrix}$ を返します

[rowDim]

行列の行の数を返します。

`rowDim(Matrix)`

例：

$\mathrm{rowDim}\left(\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}\right)$ は 2 を返します

[rowSwap]

行列および 2 つの整数が与えられたときに、与えられた行列の 2 つの整数で指定される 2 つの行を交換することによって得られる行列を返します。

`rowSwap(Matrix,Integer1,Integer2)`

例：

$\mathrm{rowSwap}\left(\begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix},1,2\right)$ は $\begin{bmatrix}3&4\\1&2\\5&6\end{bmatrix}$ を返します

[rsolve]

再帰関係を定義する式、変数、および初期条件が与えられた場合、再帰数列の閉形式解（可能な場合）を返します。上記の性質を持つ複数の項目をそれぞれ含む 3 つのリストが与えられた場合、再帰数列の系の解を求めます。

`rsolve(Expr, Var, Condition) or rsolve(List1, List2, List3)`

例：

`rsolve(u(n+1)=2*u(n)+n,u(n),u(0)=1)` は[-n+2*2^n-1]を返します

[select]

変数が 1 つのテスト式とリストまたはベクトルが与えられたときに、リストまたはベクトルの各要素をテストし、テストに合格した要素を含むリストまたはベクトルを返します。

`select(Test, List) or select(Test, Vector)`

例：

`select(x→x>=5,[1,2,6,7])` は[6,7]を返します

[seq]

1 つの式、ある区間で定義された変数、およびステップ値が与えられたときに、与えられた区間内で与えられたステップを使用して式が評価された場合に得られる数列を含むベクトルを返します。ステップが指定されない場合、使用されるステップは 1 です。

`seq(Expr, Var=Interval, [Step])`

例：

`seq(2^k,k=0..8)` は[1,2,4,8,16,32,64,128,256]を返します

[seqsolve]

rsolve に類似します。n または直前の項（x）の一方またはその両方に関する再帰関係を定義する式、変数のベクトル、および x に関する初期条件（0 番目の項）が与えられた場合、再帰数列の閉形式解（可能な場合）を返します。上記の性質を持つ複数の項目をそれぞれ含む 3 つのリストが与えられた場合、再帰数列の系の解を求めます。

`seqsolve(Expr, Vector, Condition)` または `seqsolve(List1, List2, List3)`

例：

`seqsolve(2x+n,[x,n],1)` は-n-1+2*2^n を返します

[shift]

リストまたはベクトルおよび整数 n を与えると、そのリストまたはベクトルの要素を n だけ左方向（n>0 の場合）または右方向（n<0 の場合）に動かします。整数が指定されない場合は初期設定で n=-1 が使用され、すべての要素を左方向に 1 つだけ動かします。

片側でリストから脱落した要素は、反対側で 0 に置換されます。

1 番目の整数および 2 番目の整数 n を与えると、1 番目の整数をビット単位で左方向（n>0 の場合）または右方向（n<0 の場合）に n ビットだけ動かします。

shift(list, integer)または shift(vector, integer)または shift(integer1, integer2)

例：

shift({1,2,3},2)は{3, 0, 0}を返します

[shift_phase]

三角関数に pi/2 の位相シフトを適用した結果を返します。

shift_phase(Expr)

例：

shift_phase(sin(x))は-cos((pi+2*x)/2)を返します

[signature]

順列の符号を返します。

signature(Vector)

例：

signature([2 1 4 5 3])は－1 を返します

[simult]

行列形式で与えられた 1 つの一次方程式系またはいくつかの一次方程式系の解を返します。1 つの一次方程式系の場合、係数の行列および定数の列行列を受け取り、解の列行列を返します。

simult(Matrix1, Matrix2)

例：

simult $\left(\begin{bmatrix}3 & 1\\ 3 & 2\end{bmatrix}, \begin{bmatrix}-2\\ 2\end{bmatrix}\right)$ は $\begin{bmatrix}-2\\ 4\end{bmatrix}$ を返します

[sincos]

式に含まれる複素指数関数を正弦および余弦の項で書き換えた結果を返します。

sincos(Expr)

例：

sincos(exp(i*x))は cos(x)+(i)*sin(x)を返します

[spline]

2 つのリストまたはベクトル（一方は X 値、他方は Y 値）と、変数および整数の次数が与えられたときに、2 つのリストによって指定される点を通る自然スプラインを返します。スプラインの多項式は、指定された変数に関するもので、指定された次数を使用します。

spline(ListX, ListY, Var, Integer)または spline(VectorX, VectorY, Var, Integer)

例：

spline({0,1,2},{1,3,0},x,3) は次を返します

$$\left[\frac{-5}{4}\cdot x^3+\frac{13}{4}\cdot x+1 \quad \frac{5}{4}\cdot(x-1)^3+\frac{-15}{4}\cdot(x-1)^2-\frac{1}{2}\cdot(x-1)+3\right]$$

[sqrfree]

引数を因数分解した結果を返します（同じ指数を持つ項を集めます）。

sqrfree(Expr)

例：

sqrfree((x-2)^7*(x+2)^7*(x^4-2*x^2+1)) は(x^2-1)^2*(x^2-4)^7 を返します

[sqrt]

式の平方根を返します。

sqrt(Expr)

例：

sqrt(50) は 5*sqrt(2)を返します

[srand]

整数を返し、乱数を生成する CAS ベースの関数のために乱数の数列を初期化します。

srand または srand(Integer)

[stddev]

リストの要素の標準偏差、または行列の列の標準偏差のリストを返します。オプションの 2 番目のリストは重みのリストです。

stddev(List1, [List2]) または stddev(Vector1, [Vector2]) または stddev(Matrix)

例：

stddev({1,2,3}) は $\frac{\sqrt{6}}{3}$ を返します

[stddevp]

リストの要素の母標準偏差、または行列の列の母標準偏差のリストを返します。オプションの 2 番目のリストは重みのリストです。

stddevp(List1, [List2]) または stddevp(Vector1, [Vector2]) または stddevp(Matrix)

例：

stddevp({1,2,3}) は 1 を返します

[sto]

実数または文字列を変数に格納します。

sto((Real または Str),Var)

[sturmseq]

多項式または有理分数のスツルム数列を返します。

`sturmseq(Poly,[Var])`

例：

`sturmseq(x^3-1,x)` は[1 [[1 0 0 -1] [3 0 0] 9] 1]を返します

[subMat]

4 つの整数で対角が定義される部分行列を行列から抽出します。最初の 2 つの整数が部分行列の最初の要素の行および列を定義し、最後の 2 つの整数が最後の要素の行および列を定義します。

`subMat(Matrix, Int1, Int2, Int3, Int4)`

例：

$\texttt{subMat}\left(\begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix},2,1,3,2\right)$ は $\begin{bmatrix}3&4\\5&6\end{bmatrix}$ を返します

[suppress]

リストおよび要素が与えられたときに、リスト内で最初に出現したその要素を削除し（ある場合）、結果を返します。

`suppress(List, Element)`

例：

`suppress([0 1 2 3 2],2)` は[0 1 3 2]を返します

[surd]

式および整数 n が与えられたときに、式の 1/n 乗を返します。

`surd(Expr, Integer)`

例：

`surd(8,3)` は-2 を返します

[sylvester]

2 つの多項式のシルベスター行列を返します。

`sylvester(Poly1, Poly2, Var)`

例：

$\texttt{sylvester}(x^2-1,x^3-1,x)$ は $\begin{bmatrix}1&0&-1&0&0\\0&1&0&-1&0\\0&0&1&0&-1\\1&0&0&-1&0\\0&1&0&0&-1\end{bmatrix}$ を返します

[table]

添え字が文字列または実数の配列を定義します。

`table(SeqEqual(index_name=element_value))`

[tail]

リスト、文字列、または一連のオブジェクトが与えられたときに、最初の要素を削除した結果を含むベクトルを返します。

`tail(List)` または `tail(Vector)` または `tail(String)` または `tail(Obj1, Obj2,…)`

例：

`tail([3 2 4 1 0])` は[2 4 1 0]を返します

[tan2cossin2]

式の tan(x)を(1 – cos(2*x))/ sin(2*x)で書き換えた結果を返します。

`tan2cossin2(Expr)`

例：

`tan2cossin2(tan(x))` は(1-cos(2*x))/sin(2*x)を返します

[tan2sincos2]

tan(x)を sin(2*x)/(1+cos(2*x))で書き換えた式を返します。

`tan2sincos2(Expr)`

例：

`tan2sincos2(tan(x))` は sin(2*x)/(1+cos(2*x)を返します

[transpose]

行列を転置した結果（共役なし）を返します。

`transpose(Matrix)`

例：

`transpose` $\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は $\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$ を返します

[trunc]

値または値のリストと整数 n が与えられたときに、小数点以下 n 桁までに切り捨てた値またはリストを返します。n が指定されない場合は 0 と見なされます。複素数を受け入れます。

`trunc(Real, Integer)` または `trunc(List, Integer)`

例：

`trunc(4.3)` は 4 を返します

[tsimplify]

式の超越関数を複素指数関数に書き換えた結果を返します。

`tsimplify(Expr)`

例：

`tsimplify(exp(2*x)+exp(x))` は exp(x)^2+exp(x)を返します

[type]

式のタイプ（リストや文字列など）を返します。

`type(Expr)`

例：

`type("abc")` は DOM_STRING を返します

[unapply]

式および変数で定義される関数を返します。

`unapply(Expr,Var)`

例：

`unapply(2*x^2,x)` は(x)→2*x^2 を返します

[uniform]

離散型一様確率密度関数。パラメーター a および b での、x における一様分布の確率密度を計算します。

`uniform(a, b, x)`

例：

`uniform(1.2, 3.5, 3)` は 0.434782608696 を返します

[uniform_cdf]

累積一様確率密度関数。パラメーター a および b での、値 x についての一様確率密度関数の下側裾確率を返します。オプションのパラメーター x_2 を使うと、一様確率密度関数の下側の x と x_2 の間の面積を返します。

`uniform_cdf(a, b, x, [x2])`

例：

`uniform_cdf(1.2, 3.5, 3)` は 0.782608695652 を返します

`uniform_cdf(1.2, 3.5, 2, 3)` は 0.434782608696 を返します

[uniform_icdf]

累積一様確率密度関数の逆関数。パラメーター a および b での、x についての一様分布の下側裾確率の確率が p になるような値 x を返します。

uniform_icdf(a, b, p)

例：

uniform_icdf(3.2, 5.7, 0.48)は 4.4 を返します

[UNION]

入力をすべての重複が削除されたリストに結合します。

例：

UNION({1,2,3}, {2,4,8}, 10)は{1, 2, 3, 4, 8, 10}を返します

[valuation]

多項式の評価（最低次数の項の次数）を返します。多項式のみが引数として指定された場合、返される評価は x に対するものになります。変数が 2 番目の引数として指定された場合、その変数に対して評価が実行されます。

valuation(Poly,[Var])

例：

valuation(x^4+x^3)は 3 を返します

[variance]

リストの分散、または行列の列の分散のリストを返します。オプションの 2 番目のリストは重みのリストです。

variance(List1, [List2])または variance(Matrix)

例：

variance({3, 4, 2})は 2/3 を返します

[vpotential]

ベクトル V および変数のベクトルが与えられたときに、curl(U)=V となるベクトル U を返します。

vpotential(Vector1, Vector2)

例：

vpotential([2*x*y+3,x2-4*z,-2*y*z],[x,y,z])は $\left[0 \quad -2\cdot x\cdot y\cdot z \quad 4\cdot x\cdot z-\frac{1}{3}\cdot x^3+3\cdot y\right]$ を返します

[VERSION]

システムのさまざまなコンポーネントのバージョン番号を含む文字列を返します。HP Prime 電卓の[バージョン情報]ヘルプ ページに表示されるものと同じです。整数 n を与えると、特定のコンポーネントのバージョン番号のみを返します。コンポーネントは以下の整数で識別されます。

- **[1]**—ソフトウェア バージョン
- **[2]**—ハードウェア バージョン
- **[3]**—CAS バージョン

- **[4]**—製品のシリアル番号
- **[5]**—オペレーティング システムのバージョン

[weibull]

ワイブル確率密度関数。パラメーター k、n、および t での、x におけるワイブル分布の確率密度を計算します。初期設定では t=0 です。

```
weibull(k, n, [t], x)
```

例：

`weibull(2.1, 1.2, 1.3)` は 0.58544681204 を返し、`weibull(2.1, 1.2, 0, 1.3)` も同じです

[weibull_cdf]

ワイブル分布についての累積確率密度関数。パラメーター k、n、および t での、値 x についてのワイブル確率密度関数の下側裾確率を返します。初期設定では t=0 です。オプションのパラメーター x_2 を使うと、ワイブル確率密度関数の下側の x と x_2 の間の面積を返します。

```
weibull_cdf(k, n, [t], x, [x2])
```

例：

`weibull_cdf(2.1, 1.2, 1.9)` は 0.927548261801 を返します

`weibull_cdf(2.1, 1.2, 0, 1.9)` は 0.927548261801 を返します

`weibull_cdf(2.1, 1.2, 1, 1.9)` は 0.421055367782 を返します

[weibull_icdf]

ワイブル分布についての累積確率密度関数の逆関数。パラメーター k、n、および t での、x についてのワイブル分布の下側裾確率の確率が p になるような値 x を返します。初期設定では t=0 です。

```
weibull_icdf(k, n, [t], x)
```

例：

`weibull_icdf(4.2, 1.3, 0.95)` は 1.68809330364 を返します

`weibull_icdf(4.2, 1.3, 0, 0.95)` は 1.68809330364 を返します

[when]

条件文を導入するために使用されます。

[XOR]

排他論理和。最初の式が true で 2 番目の式が false の場合、または最初の式が false で 2 番目の式が true の場合に 1 を返します。それ以外の場合は 0 を返します。

```
Expr1 XOR Expr2
```

例：

`0 XOR 1` は 1 を返します

[zip]

2 つのリストまたはベクトルの要素に 2 変数関数を適用し、結果をベクトルとして返します。初期値を指定しなかった場合、ベクトルの長さは、2 つのリストの長さの短い方になります。デフォルト値を指定した場合、短い方のリストが初期値でパディングされます。

`zip(‘function’ List1, List2, Default)` または `zip(‘function’ , Vector1, Vector2, Default)`

例：

`zip('+',[a,b,c,d], [1,2,3,4])` は[a+1 b+2 c+3 d+4]を返します

[ztrans]

数列の z 変換。

`ztrans(Expr,[Var],[ZtransVar])`

例：

`ztrans(a^n,n,z)` は – z/(a-z)を返します

[|]

[カタログ]メニューおよびテンプレート メニューに含まれるこの条件コマンドには、変数の宣言に関連するいくつかの用途があります。ある用途では、式に含まれる 1 つ以上の変数を値で置き換えるために使用されます。また、変数の定義域を定義するために使用することもできます。

`Expr|Var=Val` または `Expr|{Var1=Val1, Var2=Val2…Varn=Valn}` または `Expr|Var>n` または `Expr|Var<n` など

例：

`(X+Y)|{X=2, Y=6}` は 8 を返します

`int((1-x)^p|p>0,x,0,1)` は((-x+1)^(p+1))/(-p-1)を返します

[2]

式の 2 乗を返します。

`(Expr)²`

[π]

pi を挿入します。

[∂]

偏導関数式のテンプレートを挿入します。

[Σ]

総和式のテンプレートを挿入します。

[−]

マイナス符号を挿入します。

[√]

平方根記号を挿入します。

[∫]

式の積分を返します。

引数として 1 つの式を使用すると、このコマンドは x についての不定積分を返します。

オプションで 3 つの追加の引数を使用すると、積分の変数および定積分の境界を指定できます。

例：

`int(1/x)` は ln(abs(x))を返します

`int(sin(x),x,0,π)` は 2 を返します

`int(1/(1-x^4),x,2,3))` は-1/4*(2*atan(2)+ln(3))+1/4*(2*atan(3)-ln(2)+ln(4))を返します

[≠]

不等テスト。左辺と右辺が等しくない場合は 1 を返し、等しい場合は 0 を返します。

[≤]

小なりまたは等価の不等テスト。不等式の左辺が右辺より小さいか両辺が等しい場合は 1、そうでない場合は 0 を返します。

[≥]

大なりまたは等価の不等テスト。不等式の左辺が右辺より大きいか両辺が等しい場合は 1、そうでない場合は 0 を返します。

[▶]

式を評価した後、結果を変数 var に格納します。グラフィックス G0 から G9 では、?は使用できません。コマンド BLIT を参照してください。

```
expression ▶ var
```

[i]

虚数 *i* を挿入します。

[-1]

式の逆数を返します。

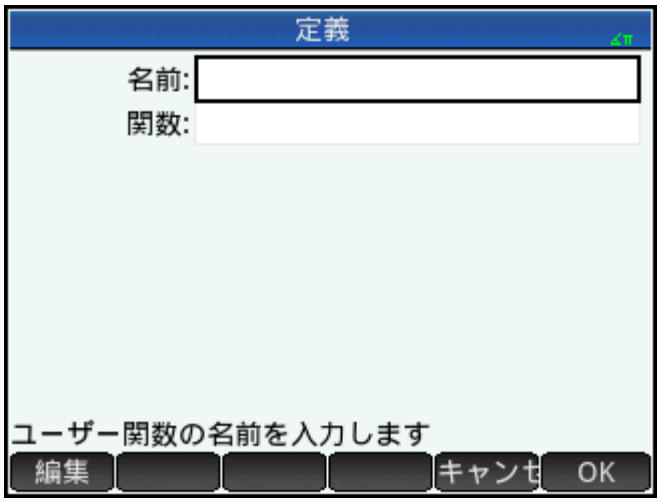
`(Expr)^-1`

ユーザー独自の関数の作成

ユーザー独自の関数を作成するには、プログラムを記述するか（第 5 章を参照）、より単純な DEFINE 機能を使用します。ユーザー自身が作成した関数は、[ユーザー]メニュー（ツールボックス メニューの 1 つ）に表示されます。

関数 SINCOS(A,B)=SIN(A)+COS(B)+C を作成するとします。

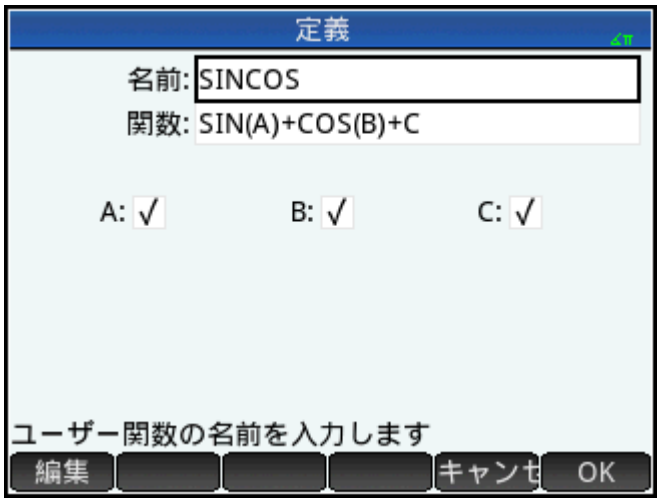
1. Shift xtθn（定義）を押します。



2. **[名前]**フィールドに関数の名前（SINCOS など）を入力し、 OK をタップします。

3. **[関数]**フィールドに関数を入力します。 SIN ALPHA A ▶ + COS ALPHA B ▶

ALPHA C OK



関数の下に、その定義に使用した変数ごとに新しいフィールドが表示されます。どのフィールドを関数の入力引数にし、どのフィールドを関数内で値が入力されないグローバル変数にするかを決める必要があります。この例では、A および B を入力変数にするため、新しい関数は 2 つの引数を取ることになります。C の値は、グローバル変数 C（初期値は 0）から提供されます。

4. A および B が選択され、C が選択されていないことを確認します。

5. OK をタップします。

   関数を実行するには、ホーム表示の入力行に関数を入力するか、[ユーザー]メニューから関数を選択します。パラメータとして選択した各変数に値を入力します。この例では、A および B をパラメータとして選択しました。したがって、SINCOS(0.5, 0.75)のように入力します。C=0 でラジアン モードの場合、これは 1.211… を返します。

23 変数

変数は、データを格納する名前付きのオブジェクトです。変数に格納されたデータは、後で使用されるか、HP Prime システムの設定を制御するために使用されます。変数には次の 4 種類がありますが、そのすべてが、 変数 を押して表示される**[変数]**メニューに含まれています。

- ホーム変数
- CAS 変数
- アプリケーション変数
- ユーザー変数

ホーム変数およびアプリケーション変数は、名前がすべて予約されています。また、型を持っています。つまり、これらの変数には、特定の種類のオブジェクトのみを格納できます。たとえば、ホーム変数 A には実数のみを格納できます。ホーム変数を使えば、行列、リスト、実数などユーザーにとって重要なデータを格納できます。アプリケーション変数は、アプリケーションにデータを格納したり、アプリケーション設定を変更したりするために使用します。アプリケーションのユーザー インターフェイス経由で同じタスクを実行することも可能ですが、アプリケーション変数を使用した方が、ホーム表示またはプログラム内からすばやくタスクを実行できます。たとえば、ホーム表示から式 `SIN(X)` を Function（関数）アプリケーション変数 `F1` に格納することも、Function（関数）アプリケーションを開いて `F1(X)` に移動し、そのフィールドに `SIN(X)` を入力することもできます。

CAS 変数およびユーザー変数はユーザーが作成できる変数で、特定の型を持ちません。また、任意の長さの名前を付けることができます。したがって、CAS 変数が `t` および `bt` のとき、`diff(t2,t)` は `2*t` を返し、`diff((bt)2, bt)` は `2*bt` を返します。さらに `2*bt` を評価しても、オブジェクトが `bt` に格納されない限り、`2*bt` が返されるだけです。たとえば、`bt:={1,2,3}`と入力した場合、`diff((bt)2, bt)`と入力しても CAS は `2*bt` を返します。しかし、その結果を（`EVAL` コマンドを使って）評価すると、CAS は`{2,4,6}`を返すようになります。

ユーザー変数は、ユーザーによって明示的に作成されます。ユーザー変数は、プログラム内で作成するか、ホーム表示で値を代入して作成します。プログラム内で作成されたユーザー変数は、ローカルとして宣言されるか、グローバルとしてエクスポートされます。値の代入によって作成されたユーザー変数やプログラムからエクスポートされた変数は、[変数]の[ユーザー]メニューに表示されます。ローカル変数は、そのプログラム内でのみ存在します。

以下の各セクションでは、変数の作成、変数へのオブジェクトの格納、変数の内容の取得など、変数に関するさまざまな手順について説明します。この章の他のセクションでは、すべてのホーム変数およびアプリケーション変数の名前を表に記載しています。

変数の操作

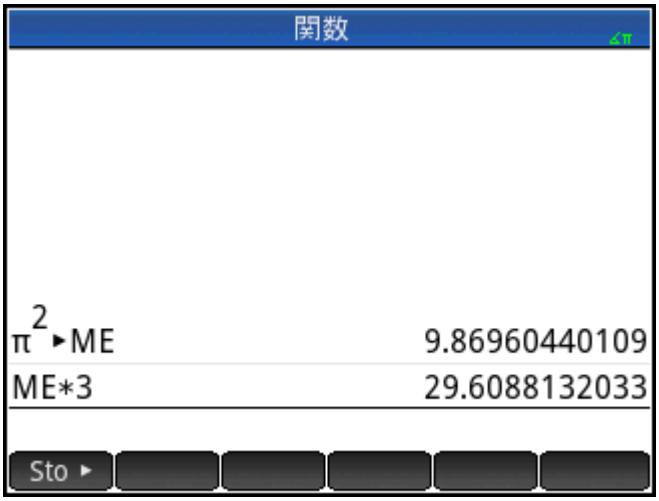
ホーム変数の操作

例 1：ホーム変数 A に π^2 を代入し、5*A を計算します。

1. Settings を押して、ホーム表示を表示します。
2. A に π² を代入します。

   Shift 3 x^2 Sto ▸ ALPHA 変数 Enter

3. A に 5 を乗じます。5 × 変数 Enter



この例は、ホーム実数変数 A ~ Z だけでなく、任意のホーム変数を格納および使用するための手順を示しています。格納するオブジェクトを正しい型のホーム変数に対応付けることが重要になります。詳しくは、463 ページのホーム変数を参照してください。

ユーザー変数の操作

例 2：ME という名前の変数を作成し、π² を代入します。

1. Settings を押して、ホーム表示を表示します。
2. π² を ME に代入します。

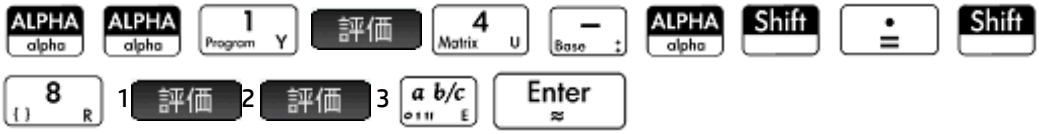
   Shift 3 x^2 Sto ▸ ALPHA +/− ALPHA $a\ b/c$ Enter

3. ME という名前の変数を作成するかどうかを尋ねるメッセージが表示されます。OK または Enter を押して操作を確定します。

   以降の計算ではこの変数を使用できます。たとえば、ME*3 は 29.6… を返します。

例 3：代入演算子を使って変数にオブジェクトを格納することもできます（Name:=Object）。この例では、ユーザー変数 YOU に{1,2,3}を格納します。

1. 代入演算子:=使用して変数にリストを代入します。



2. YOU という名前の変数を作成するかどうか尋ねるメッセージが表示されます。 OK をタップするか Enter ≈ を押して、操作を確定します。

   変数 YOU が作成され、リス{1,2,3}がこの変数に格納されます。以降の計算ではこの変数を使用できます。たとえば、YOU+60 は{61,62,63}を返します。

アプリケーション変数の操作

ホーム変数やユーザー変数に値を代入できるのとまったく同じように、アプリ変数にも値を代入できます。ホーム設定は[ホーム設定]画面（ Shift Settings ）で変更できます。しかし、ホーム表示でその設定を表す変数に値を代入することによって、ホーム設定を変更することもできます。たとえば、ホーム表示で Base:=0 Enter ≈ と入力すると、（整数ベース用の）ホーム設定フィールド**[整数]**が binary に設定されます。値が 1 の場合は octal、2 の場合は decimal、3 の場合は hex に設定されます。別の例：ホーム表示で HAngle:=1 Enter ≈ と入力すると、角度単位の設定をラジアンから度に変更できます。

HAngle:=0 Enter ≈ と入力すると、設定がラジアンに戻ります。

変数がホーム、アプリケーション、ユーザーのどれであっても、ホーム表示で変数名を入力して Enter ≈ を押すことで、変数に代入された値を確認できます。変数の名前を 1 文字ずつ入力することも、 変数 を押して[変数]メニューから変数を選択することもできます。

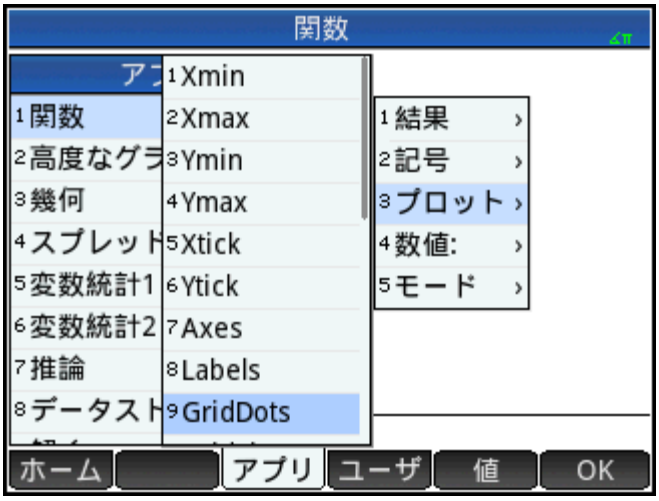
[変数]メニューに関する追加情報

[変数]メニューには、4 つの変数メニューのほかにトグルも含まれています。**[変数]**メニューから変数を選択したときに名前ではなく値が取得されるようにしたい場合は、 値 をタップします。メニュー ボタンのラベルの隣に白色のドットが表示されます。これは、このメニューがアクティブで、変数の選択時に名前でなく値が返されることを示します。

ホーム変数およびアプリケーション変数について、各変数の目的に関するヘルプを表示するには、**[変数]**メニューを使用します。目的の変数を選択して、 Help User を押します。たとえば、Function（関数）アプリケーション変数 GridDots（グリッド ドット）に関するヘルプを表示するには、以下の操作を行います。

1. 変数 を押して、**[変数]**メニューを開きます。

2. アプリ をタップして、アプリケーション変数メニューを開きます（ホーム変数のヘルプを見る場合は、 ホーム をタップします）。



3. カーソル キーを使って目的の変数に移動します。

4. Help User を押すと、その変数に関するヘルプが表示されます。

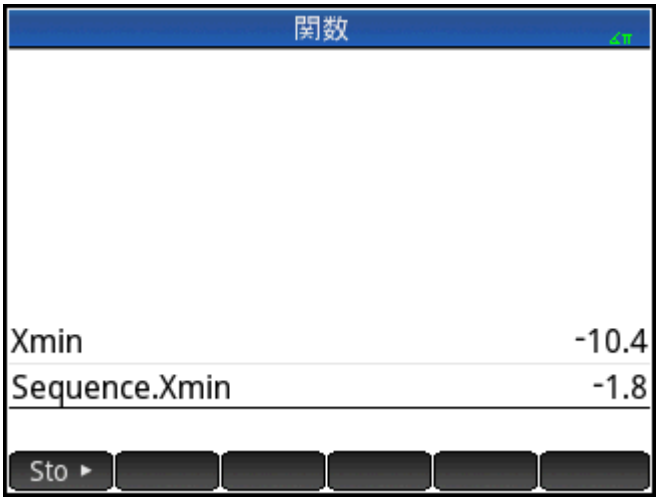
5. OK をタップして終了するか、 Esc Clear を押して**[変数]**のサブメニューに戻ります。

変数の修飾

一部のアプリケーション変数名は複数のアプリケーションで共有されています。たとえば、Function（関数）アプリケーションには名前が `Xmin` の変数がありますが、Polar（極座標）アプリケーション、Parametric（パラメトリック）アプリケーション、Sequence（数列）アプリケーション、Solve（解く）アプリケーションにもあります。これらの変数は、名前が同じであっても、通常は異なる値を保持します。複数のアプリケーションで使用されている変数の内容を、ホーム表示で名前のみを入力して取得しようとすると、現在のアプリケーションにおけるそのバージョンの変数の内容が取得されます。たとえば、Function（関数）アプリケーションがアクティブな状態でホーム表示から `Xmin` と入力した場合、Function（関数）アプリケーションの `Xmin` の値が得られます。たとえば、Sequence（数列）ア

プリケーションの Xmin の値が必要であれば、変数名を修飾する必要があります。`Sequence.Xmin` と入力すると、Sequence（数列）アプリケーションの Xmin の値が取得されます。

以下の図では、Function（関数）アプリケーションの `Xmin` の値（－10.4…）が最初に取得されました。修飾された変数を 2 番目に入力したことで、`Xmin` の値（－1.8）が取得されています。



次の構文が必要であることに注意してください：`app_name.variable_name`。

このアプリケーションは、18 種類の HP アプリケーションのどれか、または組み込みアプリケーションに基づいてユーザーが作成したアプリケーションになります。アプリケーション変数の名前は、次のアプリケーション変数表に記載されている名前と一致させる必要があります。スペースはアプリケーション名に使用できないので、アンダースコア文字で表す必要があります（ Shift ＿ ）。

ヒント： Σ や σ などの非標準文字を、変数名として入力するには、特殊記号パレット（ Shift 9 ）または文字メニュー（ Shift 変数 ）から文字を選択します。

ホーム変数

ホーム変数にアクセスするには、 変数 を押して ホーム をタップします。

カテゴリ	名前
Real（実数）	A ~ Z および θ 例：7.45 Sto ▸ A
Complex（複素数）	Z0 ~ Z9 例：2+3×i Sto ▸ Z1 または(2,3) Sto ▸ Z1（複素数の設定による）
List（リスト）	L0 ~ L9 例：{1,2,3} Sto ▸ L1
Matrix（行列）	M0 ~ M9 これらの変数には行列およびベクトルを格納します

カテゴリ	名前
	例：[[1,2],[3,4]] Sto ▸ M1
Graphics（グラフィック）	G0 ~ G9
Settings（設定）	HAngle
	HFormat
	HSeparator
	HDigits
	HComplex
	Entry（エントリ）
	Base（基数）
	Bits（ビット）
	Signed（符号付き）
System（システム）	Date（日付）
	Time（時間）
	Language（言語）
	Notes（注記）
	Programs（プログラム）
	TOff
	HVars
	DelHVars

アプリケーション変数

アプリケーション変数にアクセスするには、[a]を押してタップします。これらの変数は、以下のようにアプリケーション別にグループ化されています。組み込みアプリケーションをカスタマイズした場合、そのアプリケーションは、設定した名前でアプリケーション変数メニューに表示されます。カスタマイズしたアプリケーションの変数にアクセスするには、組み込みアプリケーションの変数にアクセスする場合と同じ手順を使用します。

Function（関数）アプリケーション変数

カテゴリ	名前	
Results（結果）（以下で説明）	SignedArea（符号付き面積）	Root（根）
	Extremum（極値）	Slope（傾き）
	Isect	
Symbolic（記号）	F1	F6
	F2	F7
	F3	F8
	F4	F9

カテゴリ	名前	
	F5	F0
Plot（プロット）	Axes（軸） Cursor（カーソル） GridDots（グリッド ドット） GridLines（グリッド線） Labels（ラベル） Method（方式） Recenter（中央に再表示） Xmax	Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Numeric（数値）	NumStart NumStep NumIndep	NumType NumZoom
Modes（モード）	AAngle ADigits AFilesB ANote AVars DelAVars	AComplex AFiles AFormat AProgram DelAFiles

[結果]の変数

[極値]

Function（関数）アプリケーションの[プロット表示]にある 関数 メニューの Extremum（極値）関数を最後に使用したときの値を含みます。アプリケーション関数 EXTREMUM（極値）の結果は、この変数には格納されません。

[Isect]

Function（関数）アプリケーションの[プロット表示]にある 関数 メニューの Isect 関数を最後に使用したときの値を含みます。アプリケーション関数 ISECT の結果は、この変数には格納されません。

[根]

Function（関数）アプリケーションの[プロット表示]にある 関数 メニューの Root（根）関数を最後に使用したときの値を含みます。アプリケーション関数 Root（根）の結果は、この変数には格納されません。

[符号付き面積]

Function（関数）アプリケーションの[プロット表示]にある 関数 メニューの SignedArea（符号付き面積）関数を最後に使用したときの値を含みます。アプリケーション関数 AREA（面積）の結果は、この変数には格納されません。

[傾き]

Function（関数）アプリケーションの[プロット表示]にある 関数 メニューの Slope（傾き）関数を最後に使用したときの値を含みます。アプリケーション関数 SLOPE（傾き）の結果は、この変数には格納されません。

Geometry（幾何）アプリケーション変数

カテゴリ	名前	
Plot（プロット）	Axes（軸）	GridDots（グリッド ドット）
	GridLines（グリッド線）	Labels（ラベル）
	PixSize（ピクセル サイズ）	ScrollText（テキストのスクロール）
	Xmax	Xmin
	Ymax	Ymin
	XTick	Ytick
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Spreadsheet（スプレッドシート）アプリケーション変数

カテゴリ	名前	
Numeric（数値）	ColWidth	RowHeight
	Row（行）	Col（列）
	Cell（セル）	
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Solve（解く）アプリケーション変数

カテゴリ	名前	
Results（結果）（以下で説明）	SignedArea（符号付き面積）	Root（根）
	Extremum（極値）	Slope（傾き）
	Isect	
Symbolic（記号）	E1	E6
	E2	E7
	E3	E8
	E4	E9
	E5	E0
Plot（プロット）	Axes（軸）	Xmin
	Cursor（カーソル）	Xtick
	GridDots（グリッド ドット）	Xzoom
	GridLines（グリッド線）	Ymax
	Labels（ラベル）	Ymin
	Method（方式）	Ytick
	Recenter（中央に再表示）	Yzoom
	Xmax	
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Advanced Graphing（高度なグラフ作成）アプリケーション変数

カテゴリ	名前	
Symbolic（記号）	V1	V6
	V2	V7
	V3	V8
	V4	V9
	V5	V0
Plot（プロット）	Axes（軸）	Xmin
	Cursor（カーソル）	Xtick
	GridDots（グリッド ドット）	Xzoom

カテゴリ	名前	
	GridLines（グリッド線）	Ymax
	Labels（ラベル）	Ymin
	Recenter（中央に再表示）	Ytick
	Xmax	Yzoom
Numeric（数値）	NumXStart	NumIndep
	NumYStart	NumType
	NumXStep	NumXZoom
	NumYStep	NumYZoom
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Statistics 1Var（1 変数統計）アプリケーション変数

カテゴリ	名前	
Results（結果）（以下で説明）	NbItem	ΣX
	MinVal	ΣX2
	Q1	MeanX
	MedVal	sX
	Q3	σX
	MaxVal	serrX
		ssx
Symbolic（記号）	H1	H4
	H2	H5
	H3	
Plot（プロット）	Axes（軸）	Xmax
	Cursor（カーソル）	Xmin
	GridDots（グリッド ドット）	Xtick
	GridLines（グリッド線）	Xzoom
	Hmin	Ymax
	Hmax	Ymin
	Hwidth	Ytick
	Labels（ラベル）	Yzoom

カテゴリ	名前	
	Recenter（中央に再表示）	
Numeric（数値）	D1	D6
	D2	D7
	D3	D8
	D4	D9
	D5	D0
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

[結果]

[NbItem]

現在の 1 変数分析（H1～H5）のデータ ポイントの数を含みます。

[MinVal]

現在の 1 変数分析（H1～H5）のデータ セットの最小値を含みます。

[Q1]

現在の 1 変数分析（H1～H5）の第 1 四分位数の値を含みます。

[MedVal]

現在の 1 変数分析（H1～H5）の中央値を含みます。

[Q3]

現在の 1 変数分析（H1～H5）の第 3 四分位数の値を含みます。

[MaxVal]

現在の 1 変数分析（H1～H5）の最大値を含みます。

[ΣX]

現在の 1 変数分析（H1～H5）のデータ セットの和を含みます。

[ΣX2]

現在の 1 変数分析（H1～H5）のデータ セットの 2 乗の和を含みます。

[MeanX]

現在の 1 変数分析（H1～H5）のデータ セットの平均を含みます。

[sX]

現在の 1 変数分析（H1～H5）のデータ セットの標本標準偏差を含みます。

[σX]

現在の 1 変数分析（H1～H5）のデータ セットの母標準偏差を含みます。

[serrX]

現在の 1 変数分析（H1～H5）のデータ セットの標準誤差を含みます。

[ssX]

現在の統計分析（H1～H5）の x の偏差平方和を含みます。

Statistics 2Var（2 変数統計）アプリケーション変数

カテゴリ	名前	
Results（結果）（以下で説明）	NbItem	σX
	Corr（相関）	serrX
	CoefDet（係数決定）	ssX
	sCov	MeanY
	σCov	ΣY
	ΣXY	ΣY2
	MeanX	sY
	ΣX	σY
	ΣX2	serrY
	sX	ssY
Symbolic（記号）	S1	S4
	S2	S5
	S3	
Plot（プロット）	Axes（軸）	Xmin
	Cursor（カーソル）	Xtick
	GridDots（グリッド ドット）	Xzoom
	GridLines（グリッド線）	Ymax
	Labels（ラベル）	Ymin
	Recenter（中央に再表示）	Ytick
	Xmax	Yzoom
Numeric（数値）	C1	C6
	C2	C7

カテゴリ	名前	
	C3	C8
	C4	C9
	C5	C0
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

[結果]

[NbItem]

現在の 2 変数分析（S1～S5）のデータ ポイントの数を含みます。

[相関]

要約統計の最新の計算から得られた相関係数を含みます。この値は、選択されたフィット タイプにかかわらず、線形フィットのみに基づきます。

[係数決定]

要約統計の最新の計算から得られた決定係数を含みます。この値は選択されたフィット タイプに基づきます。

[sCov]

現在の 2 変数統計分析（S1～S5）の標本共分散を含みます。

[σCov]

現在の 2 変数統計分析（S1～S5）の母共分散を含みます。

[ΣXY]

現在の 2 変数統計分析（S1～S5）の X·Y 積の和を含みます。

[MeanX]

現在の 2 変数統計分析（S1～S5）の独立値（X）の平均を含みます。

[ΣX]

現在の 2 変数統計分析（S1～S5）の独立値（X）の和を含みます。

[ΣX2]

現在の 2 変数統計分析（S1～S5）の独立値（X）の 2 乗の和を含みます。

[sX]

現在の 2 変数統計分析（S1～S5）の独立値（X）の標本標準偏差を含みます。

[σX]

現在の 2 変数統計分析（S1～S5）の独立値（X）の母標準偏差を含みます。

[serrX]

現在の 2 変数統計分析（S1～S5）の独立値（X）の標準誤差を含みます。

[ssX]

現在の統計分析（S1～S5）の x の偏差平方和を含みます。

[MeanY]

現在の 2 変数統計分析（S1～S5）の従属値（Y）の平均を含みます。

[ΣY]

現在の 2 変数統計分析（S1～S5）の従属値（Y）の和を含みます。

[ΣY2]

現在の 2 変数統計分析（S1～S5）の従属値（Y）の 2 乗の和を含みます。

[sY]

現在の 2 変数統計分析（S1～S5）の従属値（Y）の標本標準偏差を含みます。

[σY]

現在の 2 変数統計分析（S1～S5）の従属値（Y）の母標準偏差を含みます。

[serrY]

現在の 2 変数統計分析（S1～S5）の従属値（Y）の標準誤差を含みます。

[ssY]

現在の統計分析（S1～S5）の y の偏差平方和を含みます。

Inference（推論）アプリケーション変数

カテゴリ	名前	
Results（結果）（以下で説明）	ContribList	ContribMat
	Slope（傾き）	Inter
	Corr（相関）	CoefDet（係数決定）
	serrLine	serrSlope
	serrInter	Yval
	serrY	CritScore
	Result（結果）	CritVal1

カテゴリ	名前	
	TestScore	CritVal2
	TestValue	DF
	Prob（確率）	
Symbolic（記号）	AltHyp	InfType
	Method（方式）	
Numeric（数値）	Alpha	Pooled（プール済み）
	Conf	s1
	ExpList	s2
	Mean1	σ1
	Mean2	σ2
	n1	x1
	n2	x2
	μ0	Xlist
	π0	Ylist
	ObsList	Xval
	ObsMat	
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

[結果]

[係数決定]

決定係数の値を含みます。

[ContribList]

カイ 2 乗適合度検定のカイ 2 乗寄与度のリスト（カテゴリ別）を含みます。

[ContribMat]

カイ 2 乗 2 要因検定のカイ 2 乗寄与度の行列（カテゴリ別）を含みます。

[相関]

相関係数の値を含みます。

[CritScore]

入力された α 値に関連付けられた Z 分布または t 分布の値を含みます。

[CritVal1]

入力された α 水準から計算された負の `TestScore` 値に関連付けられた、実験変数の下側臨界値を含みます。

[CritVal2]

入力された α 水準から計算された正の `TestScore` 値に関連付けられた、実験変数の上側臨界値を含みます。

[DF]

t 検定の自由度を含みます。

[ExpList]

カイ 2 乗適合度検定の期待カウントのリスト（カテゴリ別）を含みます。

[ExpMat]

カイ 2 乗 2 要因検定の期待カウントの行列（カテゴリ別）を含みます。

[Inter]

線形 t 検定または切片信頼区間に対する回帰直線の切片の値を含みます。

[確率]

`TestScore` 値に関連付けられた確率を含みます。

[結果]

仮説検定について、帰無仮説を棄却するかしないかを示す 0 または 1 を含みます。

[serrInter]

線形 t 検定または切片信頼区間に対する切片の標準誤差を含みます。

[serrLine]

線形 t 検定に対する直線の標準誤差を含みます。

[serrSlope]

線形 t 検定または切片信頼区間に対する傾きの標準誤差を含みます。

[serrY]

平均応答信頼区間または将来応答予測区間に対する $\hat{y}$ の標準誤差を含みます。

[傾き]

線形 t 検定または切片信頼区間に対する回帰直線の傾きの値を含みます。

[TestScore]

仮説検証または信頼区間の入力から計算された Z または t 分布値を含みます。

[TestValue]

TestScore に関連付けられた実験変数の値を含みます。

[Yval]

平均応答信頼区間または将来応答予測区間に対する ŷ の値を含みます。

Parametric（パラメトリック）アプリケーション変数

カテゴリ	名前	
Symbolic（記号）	X1	X6
	Y1	Y6
	X2	X7
	Y2	Y7
	X3	X8
	Y3	Y8
	X4	X9
	Y4	Y9
	X5	X0
	Y5	Y0
Plot（プロット）	Axes（軸）	Tstep
	Cursor（カーソル）	Xmax
	GridDots（グリッド ドット）	Xmin
	GridLines（グリッド線）	Xtick
	Labels（ラベル）	Xzoom
	Method（方式）	Ymax
	Recenter（中央に再表示）	Ymin
	Tmin	Ytick
	Tmax	Yzoom
Numeric（数値）	NumStart	NumType
	NumStep	NumZoom
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Polar（極座標）アプリケーション変数

カテゴリ	名前	
Symbolic（記号）	R1	R6
	R2	R7
	R3	R8
	R4	R9
	R5	R0
Plot（プロット）	θmin	Recenter（中央に再表示）
	θmax	Xmax
	θstep	Xmin
	Axes（軸）	Xtick
	Cursor（カーソル）	Xzoom
	GridDots（グリッド ドット）	Ymax
	GridLines（グリッド線）	Ymin
	Labels（ラベル）	Ytick
	Method（方式）	Yzoom
Numeric（数値）	NumStart	NumType
	NumStep	NumZoom
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Finance（金融）アプリケーション変数

カテゴリ	名前	
Numeric（数値）	CPYR	NbPmt
	BEG	PMT
	FV	PPYR
	IPYR	PV
	GSize	
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat

カテゴリ		名前
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Linear Solver（線形ソルバー）アプリケーション変数

カテゴリ		名前
Numeric（数値）	LSystem	LSolution[a]
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

[a] Linear Solver（線形ソルバー）アプリケーションで見つかった最後の解を持つベクトルを含みます。

Triangle Solver（三角ソルバー）アプリケーション変数

カテゴリ		名前
Numeric（数値）	SideA	AngleA
	SideB	AngleB
	SideC	AngleC
	TriType	
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Linear Explorer（1 次エクスプローラー）アプリケーション変数

カテゴリ		名前
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat

カテゴリ		名前
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Quadratic Explorer（2 次エクスプローラー）アプリケーション変数

カテゴリ		名前
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

Trig Explorer（三角法エクスプローラー）アプリケーション変数

カテゴリ		名前
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

[a] Linear Solver（線形ソルバー）アプリケーションで見つかった最後の解を持つベクトルを含みます。
[2]

Sequence（数列）アプリケーション変数

カテゴリ		名前
Symbolic（記号）	U1	U6
	U2	U7
	U3	U8
	U4	U9
	U5	U0
Plot（プロット）	Axes（軸）	Xmax
	Cursor（カーソル）	Xmin
	GridDots（グリッド ドット）	Xtick

カテゴリ	名前	
	GridLines（グリッド線）	Xzoom
	Labels（ラベル）	Ymax
	Nmin	Ymin
	Nmax	Ytick
	Recenter（中央に再表示）	Yzoom
Numeric（数値）	NumIndep	NumType
	NumStart	NumZoom
	NumStep	
Modes（モード）	AAngle	AComplex
	ADigits	AFiles
	AFilesB	AFormat
	ANote	AProgram
	AVars	DelAFiles
	DelAVars	

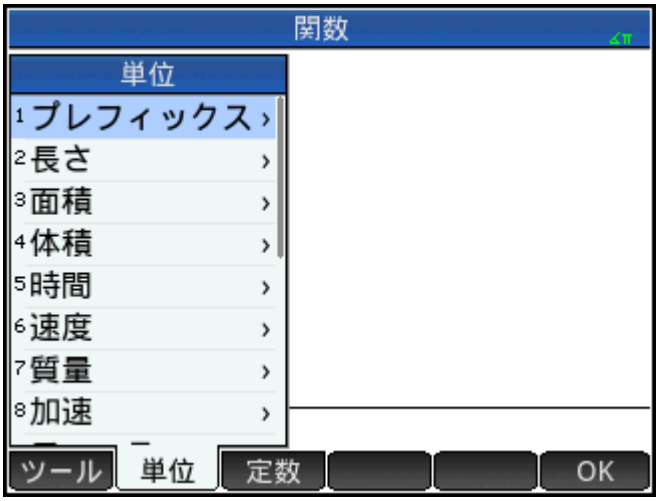
24 単位および定数

単位

インチ、オーム、ベクレルなどの測定単位を使用すると、物理量に正確な大きさを与えることができます。

測定単位は、任意の数値または数値結果に付加できます。単位付きの数値は測定値と呼ばれます。測定値の演算は、単位のない数値の演算と全く同じように実行できます。単位は、以降の演算でも数値とともに維持されます。

単位は、**[単位]**メニューにあります。 Shift （単位）を押し、必要に応じて 単位 をタップします。



このメニューはカテゴリ別に整理されています。各カテゴリが左側に一覧表示され、選択したカテゴリの単位が右側に一覧表示されます。

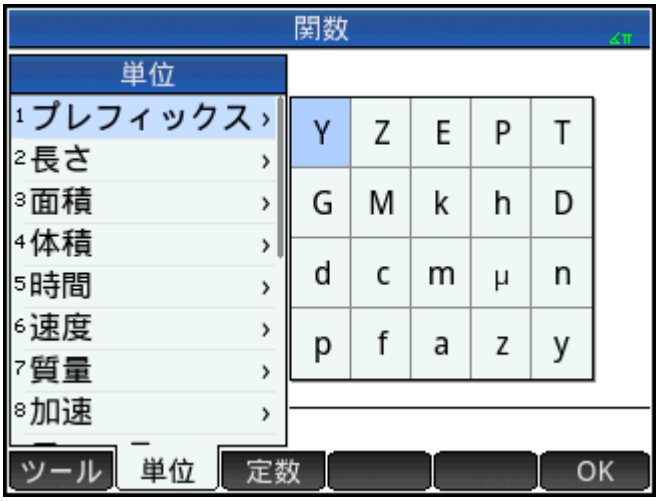
単位のカテゴリ

- [長さ]
- [面積]
- [体積]
- [時間]
- [速度]
- [質量]
- [加速度]
- [力]
- [エネルギー]
- [電源]

- [圧力]
- [温度]
- [電気]
- [光]
- [角度]
- [粘性]
- [放射]

プレフィックス

[単位]メニューには、[プレフィックス]という、単位のカテゴリではないエントリが 1 つ含まれています。このオプションを選択すると、プレフィックスのパレットが表示されます。



Y：ヨッタ	Z：ゼッタ	E: エクサ	P：ペタ	T：テラ
G：ギガ	M：メガ	k：キロ	h：ヘクト	D：デカ
d：デシ	c：センチ	m：ミリ	μ：マイクロ	n：ナノ
p：ピコ	f：フェムト	a：アト	z：ゼプト	y：ヨクト

単位プレフィックスを使用すると、大きな数値や小さな数値を簡単に入力できます。たとえば、光の速度は約 300,000 m/s です。これを計算で使用したい場合は、（プレフィックス パレットからプレフィックス k を選択することで）300_km/と入力できます。

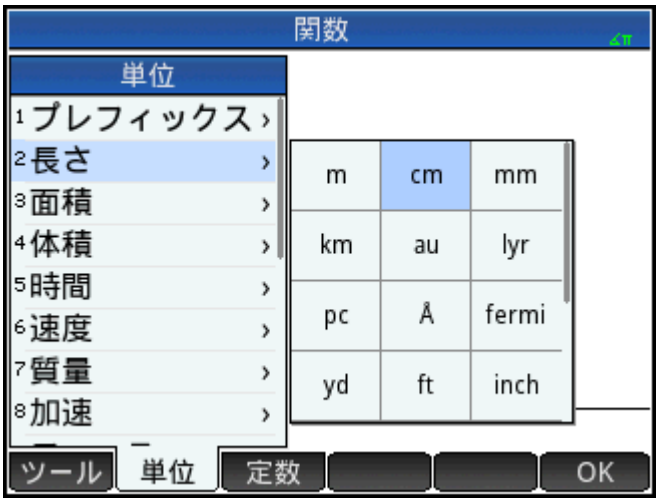
目的のプレフィックスを選択してから、単位を選択するようにしてください。

単位の計算

数値に単位を追加したものが、測定値です。計算式に複数の測定値が含まれていても、すべての測定値の単位が同じカテゴリであれば、計算を実行できます。たとえば、2 つの長さの測定値を加算できます（以下の例で示すように、長さの単位は異なっていてもかまいません）。ただし、たとえば、長さの測定値を体積の測定値に加算することはできません。

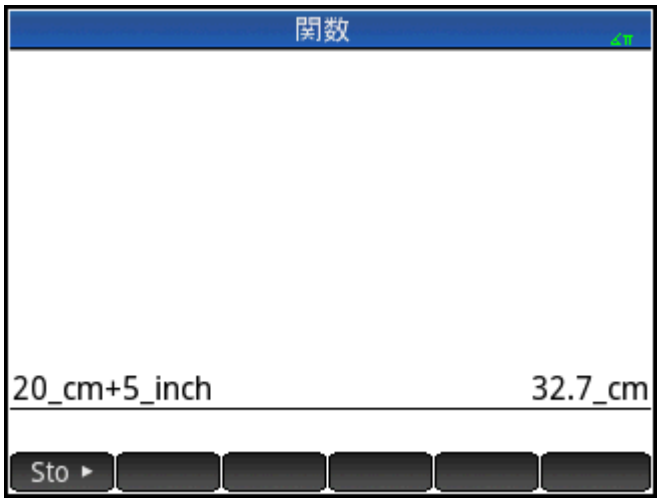
20 センチメートルと 5 インチを加算し、結果をセンチメートルで表示するとします。

1. 結果を cm で得たい場合は、センチメートルの測定値を先に入力します。20 Shift （単位）。**[長さ]**を選択します。**[cm]**を選択します。



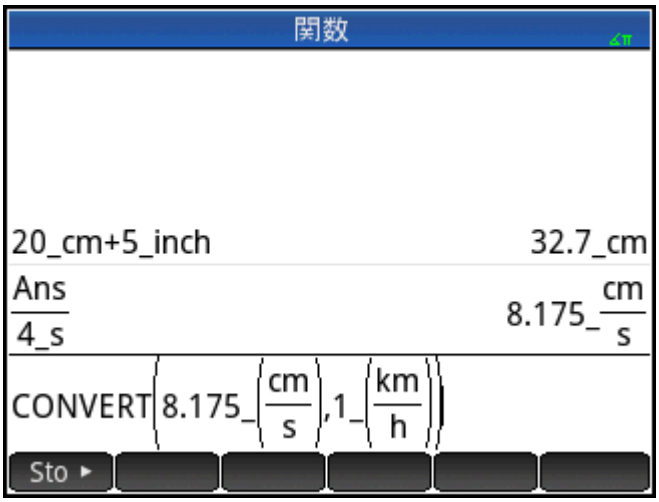
2. 次に 5 インチを加算します。 + Ans ; 5 Shift Units 。**[長さ]**を選択します。**[in]** Enter ≈ を選択します。

結果が 32.7 cm と表示されます。結果をインチで得たい場合は、5 インチを先に入力します。

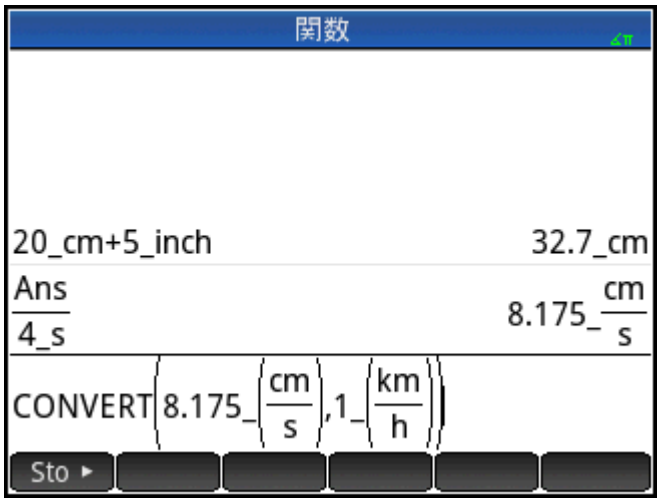


3. 例を続けるため、結果を 4 秒で割ってみます。 ÷ 4 Shift Units 。**[時間]**を選択します。**[s]** Enter を選択します。

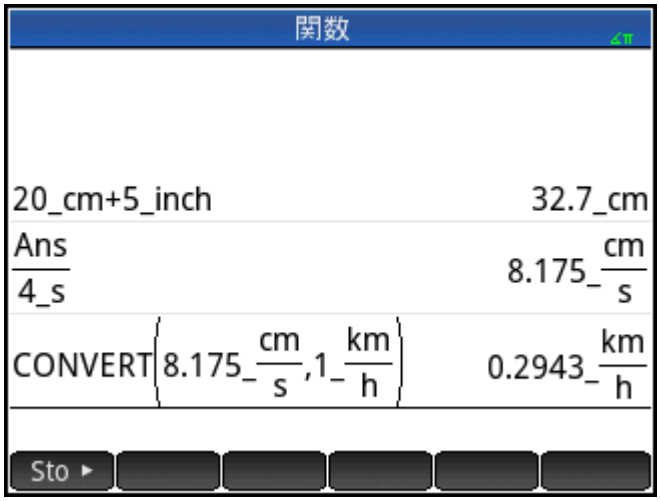
結果が 8.175 cm*s^{-1} と表示されます。



4. 次に、結果をキロメートル毎時に変換します。 Sto ▸ Shift Units 。**[速度]**を選択します。**[km/h]** Enter ≈ を選択します。



結果が 0.2943 キロメートル毎時と表示されます。



このショートカットは、[CAS 表示]では動作しません。

単位ツール

単位を管理および操作するためのツールがいくつか存在しています。これらのツールを使用するには、 Shift Units を押して、 ツール をタップします。

[変換]

ある単位を同じカテゴリの別の単位に変換します。

`CONVERT(5_m,1_ft)` は `16.4041994751_ft` を返します

また、新しい変換計算の 1 番目の引数として最後の答えを使用することもできます。 Shift + Ans ; を押すと、最後の答えが入力行に配置されます。また、履歴から値を選択して コピー をタップし、その値を入力行にコピーすることもできます。 Sto ▸ を測定値とともに使用すると、変換コマンドも呼び出され、保存記号の後の単位に変換されます。

[変換]ツールは、単一の値または値の配列の基数も変換します。

`convert(123,base,8)` は `[3, 7, 1]` を返します

結果は常に反転した数字になるため、この結果は、10 進表記の 123 は 8 進表記で 173 であることを意味します。

`convert([3, 7, 1],base,8)` は `123` を返します

[変換]ツールは、実数または比を連分数に変換するために使用することもできます。

例：

`convert(pi,confrac)` は `[3,7,15,1,292,1,1,1,2]` を返します

[MKSA]

メートル、キログラム、秒、およびアンペアの単位のことです。複雑な単位を、この MKSA 系の基本要素に変換します。

`MKSA(8.175_cm/s)` は `.08175_m/s` を返します

[UFACTOR]

単位を分解して変換します。組立単位を使用した基準を、構成単位で表現される基準に変換します。たとえば、電荷の測定単位であるクーロンは、SI ベースの単位であるアンペアおよび秒から派生した組立単位です （1 C = 1 A * 1 s)。したがって、

`UFACTOR(100_C,1_A))` は `100_A*s` を返します

[USIMPLIFY]

単位を簡略化します。たとえば、1 ジュールは 1 $kg*m^2/s^2$ と定義されます。したがって、

`USIMPLIFY(5_kg*m^2/s^2)` は `5_J` を返します

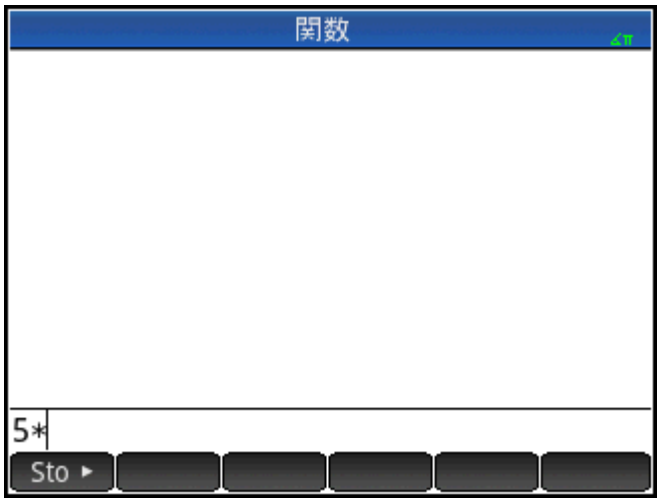
物理定数

数学および物理に関する 34 個の定数の値を（名前または値で）選択し、計算式内で使用できます。これらの定数は、[数学]、[化学]、[物理]、[量子]の 4 つのカテゴリにグループ化されています。これらすべての定数の一覧については、489 ページの定数の一覧を参照してください。

定数を表示するには、 Shift + Units を押してから、 定数 をタップします。

方程式 **E = mc2** の下で、5 単位の質量のポテンシャル エネルギーを求めるとします。

1. 質量および乗算演算子を入力します（5 [×] ）。



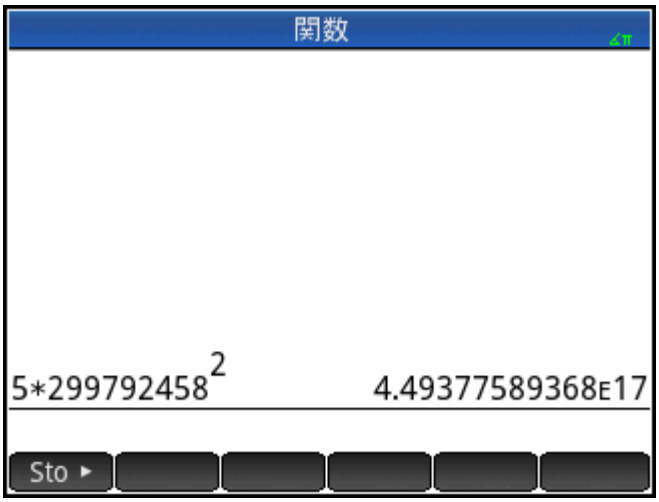
2. 定数メニューを開きます。

[Shift] [Units] 定数

3. **[物理]**を選択します。

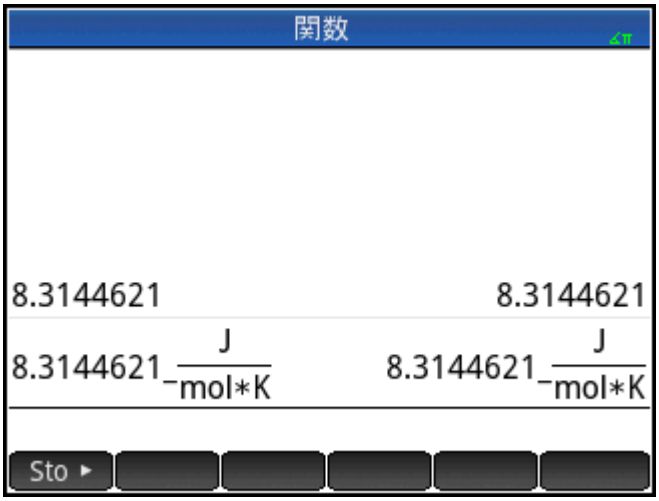
4. **[c: 299792458]**を選択します。

5. 光の速度を 2 乗して、式を評価します。 x^2 Enter



定数の値のみを入力することも、定数およびその単位（存在する場合）を入力することもできます。 値• が画面に表示されている場合は、値がカーソル位置に挿入されます。 値 が画面に表示されている場合は、値および単位がカーソル位置に挿入されます。

前の図では、最初のエントリは、 値• が表示されているときに一般気体定数を選択した場合の結果を示したものです。2 番目のエントリも同じ定数を示していますが、これは、 値 が表示されているときに選択した場合です。



値 をタップすると 値• が表示されます（その逆も同様です）。

定数の一覧

カテゴリ	名前および記号
Math（数学）	e
	MAXREAL
	MINREAL
	π
	i
Chemistry（化学）	アボガドロ：NA
	ボルツマン：k
	モル体積：Vm
	一般気体：R
	標準温度：StdT
	標準圧力：StdP
Physics（物理）	ステファンボルツマン：σ
	光速：c
	誘電率：ϵ_0
	透磁率：μ_0
	重力加速度：g
	重力：G
Quantum（量子）	プランク：h
	ディラック：ħ
	電子電荷：q
	電子の質量：me
	q/me 比率：qme

カテゴリ	名前および記号
	陽子の質量：mp
	mp/me 比率：mpme
	微細構造：α
	磁束量子：ϕ
	ファラデー：F
	リュードベリ：R_∞
	ボーア半径：a_0
	ボーア磁子：μ
	核磁子：μ_N
	光子波長：λ_0
	光子周波数：f_0
	コンプトン波長：λ_c

25 リスト

リストとは、複数の実数、複素数、式、または行列をコンマで区切り、全体を中括弧で囲んだものです。たとえば、リストには、{1,2,3}のように一連の実数を含めることができます。リストを使用すると、関連のあるオブジェクトを簡単にグループ化できます。

リストの操作は、ホーム表示およびプログラム内で実行できます。

L0 ~ L9 という名前の 10 個のリスト変数が使用できるほか、独自のリスト変数名を作成することもできます。これらの変数は、ホーム表示やプログラム内の計算や式で使用できます。[変数]メニュー（ 変数 ）からリスト名を取得するか、キーボードからリスト名を入力してください。

List Catalog（リスト カタログ）で、名前付きのリストを作成、編集、削除、送信、および受信できます（ Shift 7 List Q （リスト））。また、ホーム表示でリスト（名前付きまたは名前なし）を作成して格納することもできます。

リスト変数の動作は、2 変数統計アプリケーションの列 C1～C0、および 1 変数統計アプリケーションの列 D1～D0 と同じです。統計列をリストとして格納したり（その逆も可能です）、統計列に対して任意のリスト関数を使用したり、リスト変数に対して統計関数を使用したりできます。

リスト カタログでのリストの作成

1. List Catalog（リスト カタログ）を開きます。

   Shift 7 List Q （リスト）。

   リスト名の横にリストの要素数が表示されます。

2. 新しいリストに割り当てる名前（L1、L2 など）をタップします。リスト エディターが表示されます。

すでに要素を含んでいるリストを編集するのではなく新しいリストを作成する場合は、必ず要素が 1 つも含まれていないリストを選択するようにしてください。

3. 値ごとに Enter を押して、必要な値をリストに入力します。

値には実数または複素数（または式）を使用できます。式を入力した場合、その式が評価され、結果がリストに挿入されます。

4. 操作が完了したら、Shift 7 List（リスト）を押して List Catalog（リスト カタログ）に戻るか、Settings を押してホーム表示に移動します。

List Catalog（リスト カタログ）のボタンおよびキーは次のとおりです。

ボタンまたはキー	用途
編集	編集するために選択されたリストを開きます。リスト名をタップするだけでも開くことができます

ボタンまたはキー	用途
削除 または Del	選択されたリストの内容を削除します
送信	選択されたリストを別の HP Prime 電卓に送信します（ある場合）
Shift Esc Clear （クリア）	すべてのリストをクリアします
Shift ▲ または ▼	カタログの先頭または末尾に移動します

List（リスト）エディター

List（リスト）エディターは、リストにデータを入力するための特別な環境です。List Catalog（リストカタログ）が開いている状態で List（リスト）エディターを開くには、次の 2 つの方法があります。

- リストを選択して 編集 をタップします。または、
- リストの名前をタップします。

List（リスト）エディター：ボタンおよびキー

リストを開くと、以下のボタンおよびキーが使用可能になります。

ボタンまたはキー	用途
編集	選択されたリスト項目を入力行にコピーします
その他	リストを編集するためのメニュー オプションを開きます
移動	指定したリスト要素にカーソルを移動します。このオプションは、リストが非常に長いときに特に便利です
移動	Enter ≈ を押したときのカーソルの移動方法を設定します。このオプションは、**[下]**、**[右]**、および**[なし]**です
Shift Esc Clear （クリア）	リストからすべての項目をクリアします
Shift ▲ または ▼	リストの先頭または末尾にカーソルを移動します

[リスト]エディター：[その他]メニュー

[リスト]エディターの[その他]メニューには、リストを編集するためのオプションが含まれています。オプションの説明を次の表に示します。

カテゴリ	オプション	説明
[挿入]	[行]	リスト内の現在の行の上に新しい行を挿入します。新しい行には 0 が含まれます
[削除]	[列]	現在のリスト（列）の内容を削除します。特定の要素を削除するには、その要素を選択して Del を押します
[選択]	[行]	現在の行を選択します。選択後、その行をコピーできます
	[列]	現在の列を選択します。選択後、その列をコピーできます
	[ボックス]	開始位置および終了位置を指定して要素を矩形選択するためのダイアログ ボックスを開きます。開始位置となるセルをタップしたまま選択し、指でドラッグして要素を矩形選択することもできます。選択後、その要素をコピーできます
[選択]		選択モードのオン/オフを切り替えます。セルをタップしたまま指でドラッグして、複数のセルを選択することもできます
[スワップ]	[列]	選択した 2 つの列の値を入れ替えます

リストの編集

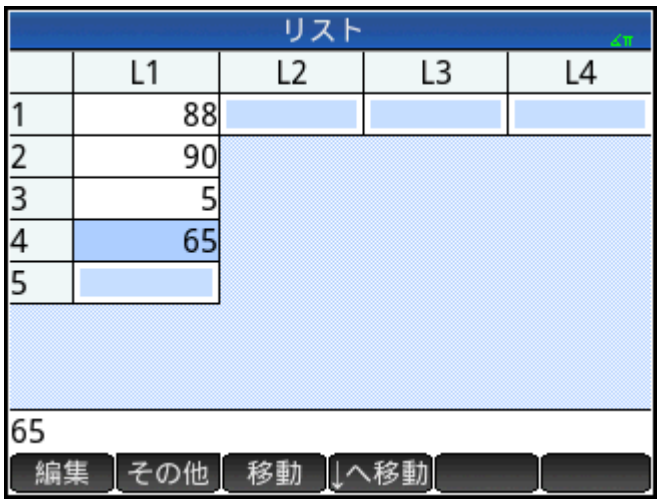
1. List Catalog（リスト カタログ）を開きます。

   Shift 7 List Q （リスト）。

2. リストの名前（**[L1]**や**[L2]**など）をタップします。List（リスト）エディターが表示されます。

3. 編集する要素をタップします（または、編集する要素が強調表示されるまで ▲ または ▼ を押します）。この例では 3 番目の要素を編集し、値が 5 になるようにします。

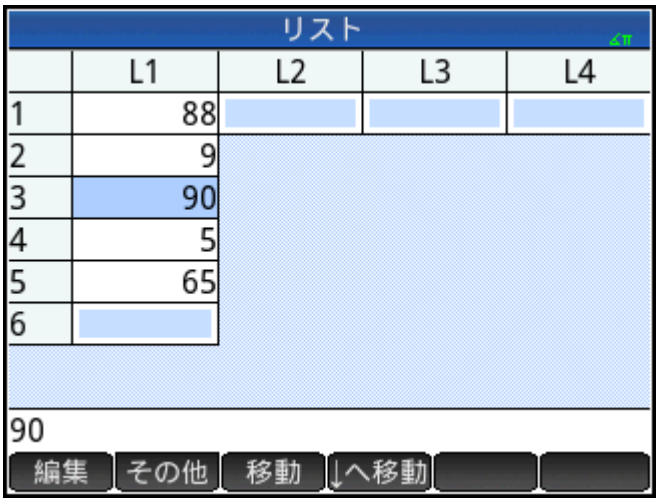
（5 OK ）。



リストへの要素の挿入

次の図に示したリスト L1 の L1(2)に新しい値 9 を挿入するとします。

1. L1(2)を選択します。つまり、リストの 2 番目の要素を選択します。
2. その他 をタップし、**[挿入]**を選択して、**[行]**を選択します。
3. 9 を入力し、 OK をタップします。



リストの削除

リストの削除

List Catalog（リスト カタログ）で、カーソル キーを使用してリストを選択し、 Del を押します。

操作の確認を求められます。 OK をタップするか Enter ≈ を押します。

予約済みのリスト（L0 から L9）のどれかである場合、リストの内容のみが削除されます。リストから単純に内容が消去されます。ユーザーが名前を付けたリスト（L0 から L9 以外のリスト）である場合、リストが完全に削除されます。

すべてのリストの削除

List Catalog（リスト カタログ）で、 Shift Esc Clear （クリア）を押します。

リスト L0 から L9 は内容が削除され、他のすべての名前付きリストは完全に削除されます。

ホーム表示でのリストの操作

リストの入力や操作は、ホーム表示から直接行えます。名前付きのリストでも名前なしのリストでも操作できます。

リストの作成

1. Press Shift 8 {} R （{}）を押します。

   1 組の中かっこが入力行に表示されます。どのリストも必ず中かっこで囲む必要があります。

2. 最初の要素をリストに入力してから、コンマを入力します（[要素] , Eval O）。

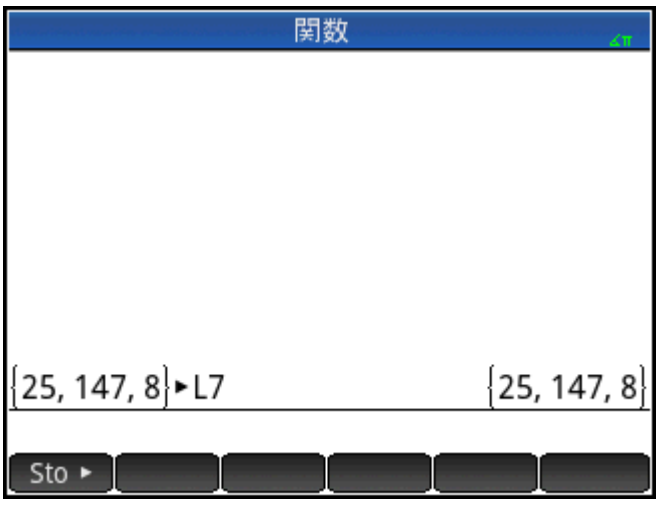
3. 各要素をコンマで区切って追加していきます。

4. 要素の入力が完了したら、 Enter ≈ を押します。リストが履歴に追加されます（要素間の式がすべて評価されます）。

リストの格納

リストを変数に格納できます。リストは、履歴に追加する前に変数に格納することも、履歴から変数にコピーすることもできます。入力行へのリストの入力、または履歴から入力行へのリストのコピーが完了したら、 Sto ▸ をタップしてリストの名前を入力し、 Enter ≈ を押します。ユーザーが使用できる予約済みのリスト変数名は、`L0` から `L9` です。ただし、ユーザー独自のリスト変数名を作成することもできます。

たとえば、リスト{25,147,8}を `L7` に格納するには、以下の操作を行います。



1. 入力行でリストを作成します。
2. ▶ を押して、リストの外側にカーソルを移動させます。
3. Sto ▸ をタップします。
4. 名前を入力します。

   ALPHA alpha x^2 7
5. 操作を完了します（ Enter ≈ ）。

リストの表示

ホーム表示でリストを表示するには、その名前を入力して Enter ≈ を押します。

リストが空の場合は、1 組の空の中かっこが返されます。

1 つの要素の表示

ホーム表示でリストの 1 つの要素を表示するには、*listname (element#)*と入力します。たとえば、L6 が{3,4,5,6}の場合、`L6(2)` Enter ≈ と入力すると、4 が返されます。

1 つの要素の格納

ホーム表示でリストの 1 つの要素に値を格納するには、*value* Sto ▸ *listname (element#)*と入力します。たとえば、148 を L2 の 2 番目の要素として格納するには、`148` Sto ▸ `L2(2)` Enter ≈ と入力します。

リストの参照

L1:={5, "abcde", {1,2,3,4,5}, 11}とします。`L1(1)` は 5 を返し、`L1(2)` は"abcde"を返します。`L1(2, 4)` は 100（d の ASCII コード）を返し、`L1(2,4,1)` は"d"を返します。`L1({2,4})` と入力すると、2 から 4 までのすべての要素を含むサブリストが抽出され、{"abcde", {1,2,3,4,5},11}が返されます。

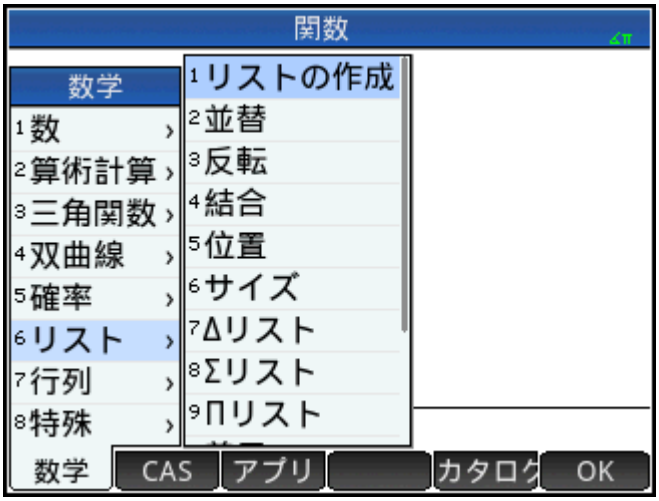
リストの送信

リストは、アプリケーション、プログラム、行列、および注記と同じように、別の電卓や PC に送信できます。

リスト関数

リスト関数は[数学]メニューにあります。リスト関数はホームやプログラム内で使用できます。

関数の名前を入力することも、[数学]メニューの[リスト]カテゴリから関数の名前をコピーすることもできます。



[数学]メニューの左側の列で**[リスト]**カテゴリを選択するには、 Mem 8 6 を押します（**[リスト]**は**[数学]**メニューの 6 番目のカテゴリであるため、6 を押すと**[リスト]**カテゴリに直接移動できます）。関数をタップして選択するか、方向キーを使って関数を選択してから、 OK をタップするか Enter ≈ を押します。

リスト関数はかっこで囲まれています。リスト関数の引数は、`CONCAT(L1,L2)`のようにコンマで区切ります。引数には、リストの変数名または実際のリストで指定できます。たとえば、`REVERSE(L1)`や`REVERSE({1,2,3})`のようになります。

+、−、×、÷などの一般的な演算子は、リストを引数に取ることができます。引数が 2 つあり、その両方がリストである場合は、2 つの要素が 1 組として計算されるため、2 つのリストの長さは同じである必要があります。引数が 2 つあり、一方が実数である場合は、リストの各要素に対して計算が行われます。

例：

`5*{1,2,3}`は`{5,10,15}`を返します。

数値、行列、またはリストを引数に取ることができる一般的な演算子のほかに、リストしか操作できないコマンドもあります。

メニュー形式

初期設定では、リスト関数は、一般的なコマンド名ではなく説明的な名前で[数学]メニューに表示されます。したがって、短縮名 `CONCAT` は**[結合]**と表示され、`POS` は**[位置]**と表示されます。

[数学]メニューに代わりにコマンド名が表示されるようにするには、[ホーム設定]画面の 2 ページ目で**[メニュー表示]**オプションの選択を解除します。

[差異]

2 つのリストの共通ではない要素のリストを返します。

`DIFFERENCE({1,2,3,4}, {1,3,5,7})`は{2,4,5,7}を返します

[交差]

2 つのリストに共通する要素のリストを返します。

`INTERSECT({1,2,3,4}, {1,3,5,7})` は{1,3}を返します

[リストの作成]

次の構文を使用して、新しいリストの一連の要素を計算します。

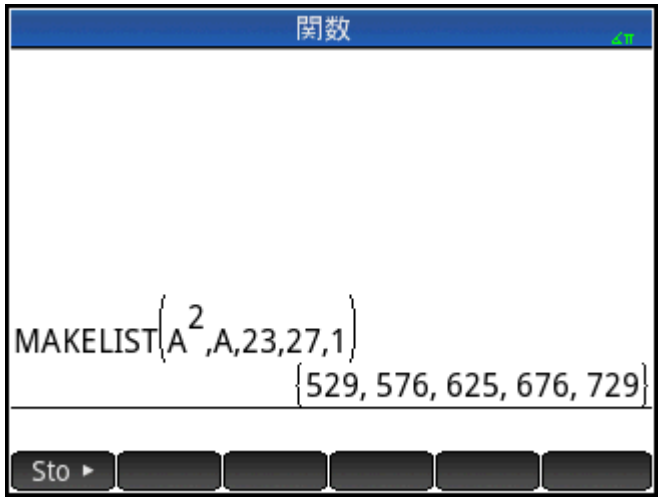
`MAKELIST(expression,variable,begin,end,increment)`

指定された増分ステップで *variable* が *begin* から *end* までの値を取るとき、*variable* に関する *expression* を評価します。

例：

ホームで、23 から 27 までの 2 乗を続けて生成します。

[リスト]を選択します（ Mem B ）。**[リストの作成]**または**[MAKELIST]**を選択し、 ALPHA alpha 変数 x^2 , Eval 0 ALPHA alpha 変数 , Eval 0 23 , Eval 0 27 , Eval 0 1 Enter ≈ と入力します。



[並べ替え]

リストの要素を昇順で並べ替えます。

`SORT(list)`

例：

`SORT({2,5,3})` は`{2,3,5}`を返します

[反転]

リストの要素の順序を反転してリストを作成します。

`REVERSE(list)`

例：

`REVERSE({1,2,3})` は`{3,2,1}`を返します

[結合]

2 つのリストを結合して新しいリストを作成します。

CONCAT(list1,list2)

例：

CONCAT({1,2,3},{4})は{1,2,3,4}を返します。

[位置]

リスト内での要素の位置を返します。要素は、値、変数、または式にすることができます。要素のインスタンスが複数存在する場合、最初に見つかった要素の位置が返されます。指定した要素が見つからなかった場合、値 0 が返されます。

POS(list, element)

例：

POS ({3,7,12,19},12)は 3 を返します

[サイズ]

リストの要素数を返すか、ベクトルまたは行列の次元を含むリストを返します。

SIZE(list)またはSIZE(Vector)またはSIZE(Matrix)

例：

SIZE({1,2,3})は 3 を返します

SIZE([[1 2 3], [4 5 6]])は{2, 3}を返します

[ΔLIST]

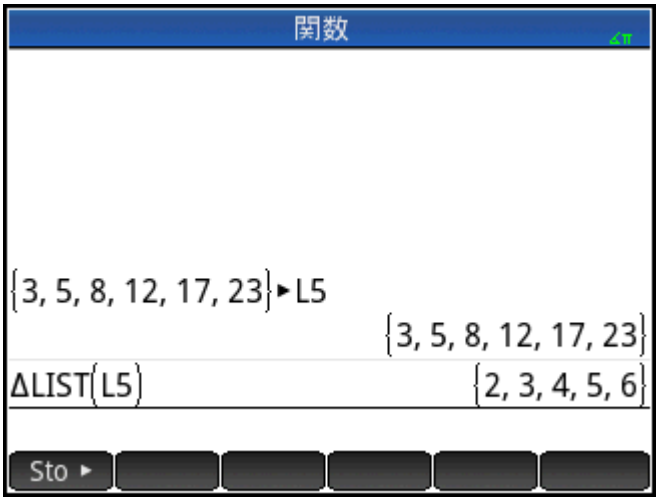
リストの第 1 階差（リスト内の連続する要素の差）で構成される新しいリストを作成します。新しいリストの要素数は、元のリストより 1 つ少なくなります。$\{x_1, x_2, x_3, \ldots x_{n-1}, x_n\}$の階差は、$\{x_2 - x_1, x_3 - x_2, \ldots x_n - x_{n-1}\}$になります。

ΔLIST(list1)

例：

ホーム表示で{3,5,8,12,17,23}を L5 に格納し、このリストの第 1 階差を求めます。

Shift 8 3,5,8,12,17,23 ▶ Sto▸ ALPHA x^2 5 Enter Mem と入力し、**[リスト]**を選択します。Δ**[ListA]**を選択します。 ALPHA x^2 5 Enter と入力します。



[ΣLIST]

リスト内のすべての要素の和を計算します。

`ΣLIST(list)`

例：

`ΣLIST({2,3,4})` は `9` を返します。

[πLIST]

リスト内のすべての要素の積を計算します。

`πLIST(list)`

例：

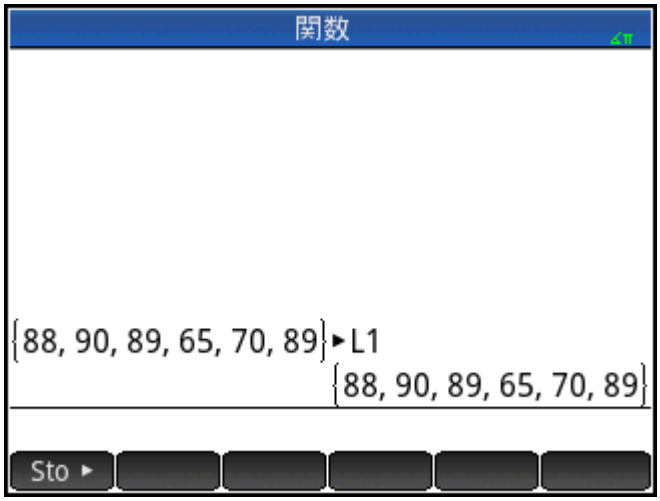
`πLIST({2,3,4})` は `24` を返します。

リストの統計値の計算

統計値（リストの平均値、中央値、最大値、最小値など）を求めるには、リストを作成してデータセットに格納してから、1 変数統計アプリケーションを使用します。

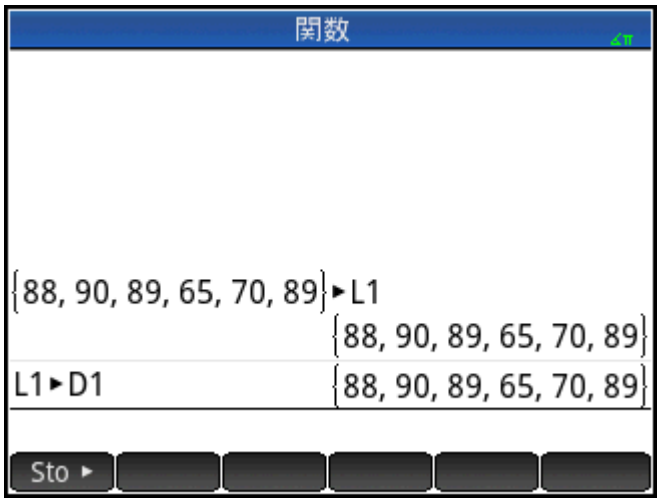
この例では、リスト L1 の要素（88、90、89、65、70、および 89）の平均値、中央値、最大値、および最小値を、1 変数統計アプリケーションを使って求めます。

1. ホーム表示で、L1 を作成します。



2. ホーム表示で、L1 を D1 に格納します。

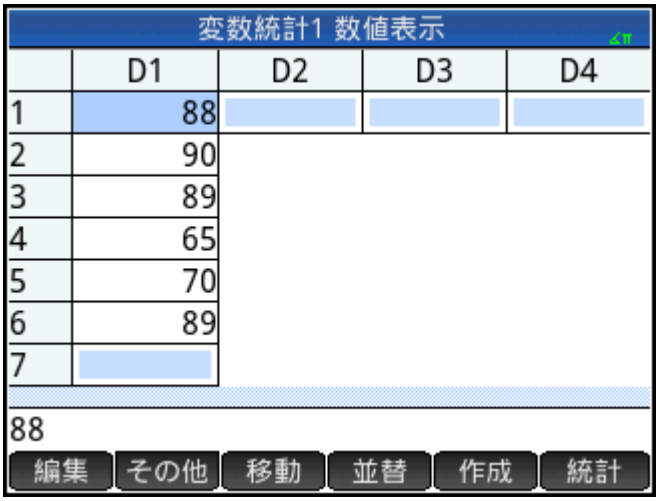
ALPHA alpha | x^2 | 1 | Sto ▸ | ALPHA alpha | xtθn Define D | 1 | Enter ≈



これで、1 変数統計アプリケーションの[数値表示]にリスト データを表示できます。

3. 1 変数統計アプリケーションを起動します。

   Apps Info **[1 変数統計]**を選択し、データ セット D1 にリストの要素が含まれていることを確認します。



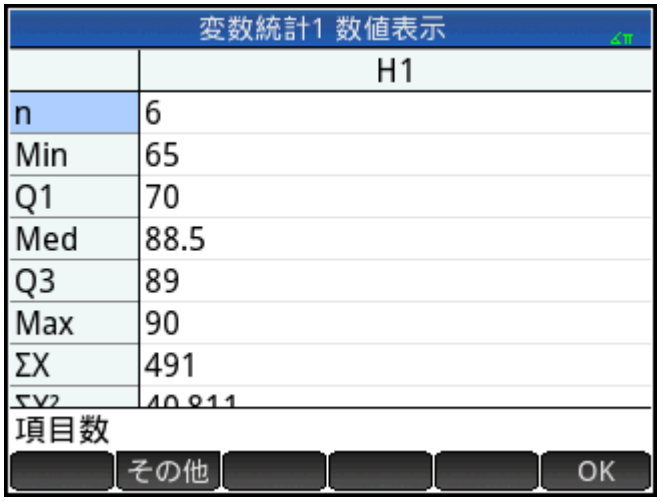
4. [記号表示]で、統計を求めるデータ セットを指定します。

   Symb Setup

初期設定では、H1 は D1 のデータを使用するため、[記号表示]で追加の操作を行う必要はありません。ただし、対象のデータが D2 など D1 以外の列に含まれていた場合は、目的のデータ列をここで指定する必要があります。

5. 統計を計算します。

Num Setup 統計



6. 完了したら、 OK をタップします。

26 行列

ホーム表示、CAS、またはプログラム内で、行列やベクトルを作成、編集、および操作できます。ホームまたは CAS から行列を直接入力することも、Matrix（行列）エディターを使用することもできます。

ベクトル

ベクトルは 1 次元の配列です。これらは 1 つの行だけで構成されます。ベクトルは、[1 2 3]のように 1 重かっこで表されます。ベクトルには、実数ベクトルまたは[1+2*i 7+3*i]のような複素数ベクトルがあります。

行列

行列は 2 次元の配列です。これらは、少なくとも 2 つの行および 1 つの列で構成されます。行列には、次のように実数と複素数の任意の組み合わせを含めることができます。

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ または $\begin{bmatrix} 1+2i \\ 3-4i \\ 7 \end{bmatrix}$

行列変数

M0 ~ M9 という名前の 10 個の予約済み行列変数が使用できます。ただし、ユーザーが定義した変数名に行列を保存することもできます。これらの行列変数は、ホーム表示や[CAS 表示]、またはプログラム内の計算で使用できます。行列名は、Vars[変数]メニューから取得することも、単純にキーボードから入力することもできます。

行列の作成および格納

行列カタログには、予約済み行列変数 M0 から M9、およびホーム表示または[CAS 表示]で（グローバル行列変数の場合はプログラム内で）ユーザーによって作成されたすべての行列変数が含まれます。

行列名を選択したら、Matrix（行列）エディターで行列を作成、編集、および削除できます。また、行列を別の HP Prime に送信することもできます。

Matrix Catalog（行列カタログ）を開くには、 Shift 4 Matrix （行列）を押します。

Matrix Catalog（行列カタログ）では、行列名の横に行列のサイズが表示されます（空の行列は 1*1 と表示されます）。ベクトルの横には行列の要素数が表示されます。

また、ホーム表示で行列（名前付きまたは名前なし）を作成して格納することもできます。たとえば、次のコマンド

```
POLYROOT([1,0,-1,0])▶M1
```

は、長さ 3 の複素数ベクトルの根を変数 M1 に格納します。したがって、M1 には、0、1、および－1 の 3 つの根が格納されます。

Matrix Catalog（行列カタログ）：ボタンおよびキー

Matrix Catalog（行列カタログ）で利用可能なボタンおよびキーは次のとおりです。

ボタンまたはキー	用途
編集	編集するために選択された行列を開きます
削除 または Del	選択された行列の内容を削除します
ベクトル	選択された行列を 1 次元のベクトルに変更します
送信	選択された行列を別の HP Prime 電卓に送信します（ある場合）
Shift Esc Clear （クリア）	予約済み行列変数 M0 から M9 の内容をクリアし、ユーザーによって名前が付けられた行列をすべて削除します

行列の操作

Matrix（行列）エディターを開く

行列を作成および編集するには、Matrix Catalog（行列カタログ）に移動して、行列をタップします（カーソル キーを使用して行列を選択してから 編集 を押すこともできます）。Matrix（行列）エディターが開きます。

Matrix（行列）エディター：ボタンおよびキー

[行列]エディターで利用可能なボタンおよびキーは次のとおりです。

ボタンまたはキー	用途
編集	要素を編集できる入力行に、選択した要素をコピーします。この項目は、行列またはベクトル内の要素が選択されているときにのみ表示されます
その他	編集オプションのメニューを表示します
移動	指定した行列要素にカーソルを移動します。このオプションは、行列が非常に大きいときに特に便利です

ボタンまたはキー	用途
移動	Enter ≈ を押したときのカーソルの移動方法を設定します。このオプションは、**[下]**、**[右]**、および**[なし]**です
Shift Esc Clear （クリア）	選択された行、選択された列、または行列全体を削除します（選択するよう求められます）
Shift ▲ ▼ ◀ ▶	カーソルを最初の行、最後の行、最初の列、最後の列にそれぞれ移動させます

[行列]エディター：[その他]メニュー

[行列]エディターの[その他]メニューに含まれるオプションは[リスト]エディターの[その他]メニューとほぼ同じですが、行列編集用の追加オプションがある点が異なります。オプションの説明を次の表に示します。

カテゴリ	オプション	説明
[挿入]	[行]	行列内の現在の行の上に新しい行を挿入します。新しい行には 0 が含まれます
	[列]	行列内の現在の列の左に新しい列を挿入します。新しい列には 0 が含まれます
[削除]	[行]	行列の現在の行を削除します
	[列]	行列内の現在の列を削除します
	[すべて]	行列の内容を削除します
[選択]	[行]	現在の行を選択します。選択後、その行をコピーできます
	[列]	現在の列を選択します。選択後、その列をコピーできます
	[ボックス]	開始位置および終了位置を指定して要素を矩形選択するためのダイアログ ボックスを開きます。開始位置となるセルをタップしたまま選択し、指でドラッグして要素を矩形選択することもできます。選択後、その要素をコピーできます
[選択]		選択モードのオン/オフを切り替えます。セルをタップしたまま指でドラッグして、複数のセルを選択することもできます
[スワップ]	[行]	選択した 2 つの行の値を入れ替えます
	[列]	選択した 2 つの列の値を入れ替えます

[行列]エディターでの行列の作成

1. Matrix Catalog（行列カタログ）を開きます。

   Shift 4 Matrix U （行列）

2. ベクトルを作成する場合は、使用する行列が選択されるまで ▲ または ▼ を押し、

   ベクトル をタップしてから Enter ≈ を押します。以下の手順 4 に進みます。

3. 行列を作成する場合は、行列の名前（M0 から M9）をタップするか、使用する行列が選択されるまで ▲ または ▼ を押してから、 Enter ≈ を押します。

   空の行列が表示され、名前の横に `1*1` というサイズが表示されます。

4. 行列の要素ごとに、数値または式を入力してから OK をタップするか Enter ≈ を押します。

   複素数は、(*a*, *b*)（ここで、*a* は実部、*b* は虚部）という複素形式で入力できます。また、*a*+*bi* の形式で入力することもできます。

5. 初期設定では、要素の入力が完了すると、同じ行の次の列にカーソルが移動します。別の行または列に移動するには、カーソル キーを使用します。また、カーソルが自動的に移動する方向を変更するには、 移動 をタップします。 移動 ボタンを使用すると、以下のオプションが切り替わります。

   - →へ移動 : Enter ≈ を押すと、カーソルが現在のセルの右側のセルに移動します。
   - ↓へ移動 : Enter ≈ を押すと、カーソルが現在のセルの下側のセルに移動します。
   - 移動 : Enter ≈ を押しても、カーソルは現在のセルにとどまります。

6. 操作が完了したら、 Shift 4 Matrix U （行列）を押して Matrix Catalog（行列カタログ）に戻るか、 Settings を押してホーム表示に戻ります。行列の入力内容が自動的に保存されます。

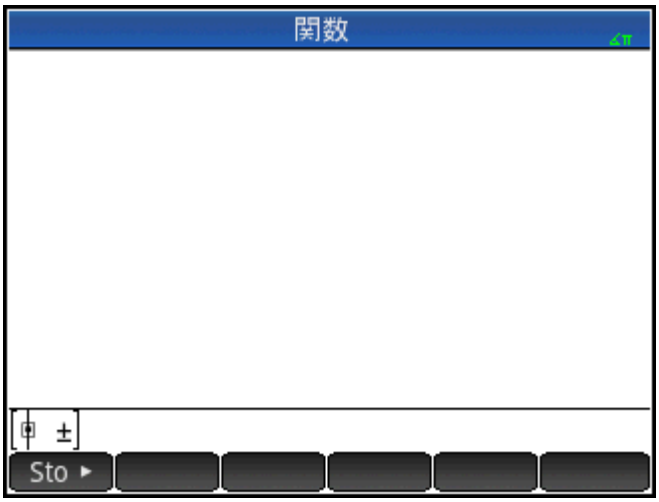
ホーム表示での行列の操作

リストの入力や操作は、ホーム表示から直接行えます。名前付きの行列でも名前なしの行列でも操作できます。

ホーム表示または[CAS 表示]で、直接ベクトルまたは行列を入力行に入力します。

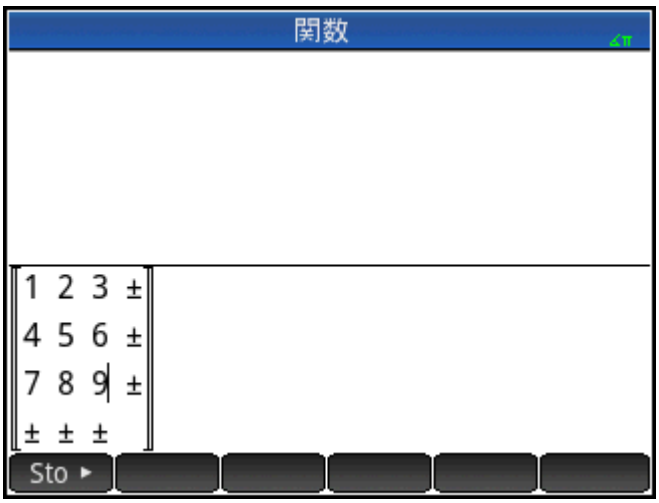
1. Shift 5 [ ] V を押すとベクトルを開始します。次に、 Shift 5 [ ] V を再び押すと行列を開始します。または、 Units C を押してテンプレート メニューを開き、ベクトル テンプレートまたはどれかの行列テンプレートを選択することもできます。以下の図は、ベクトルが開始され、

最初の値を入力するための暗い正方形のプレースホルダーが表示されている状態を示しています。

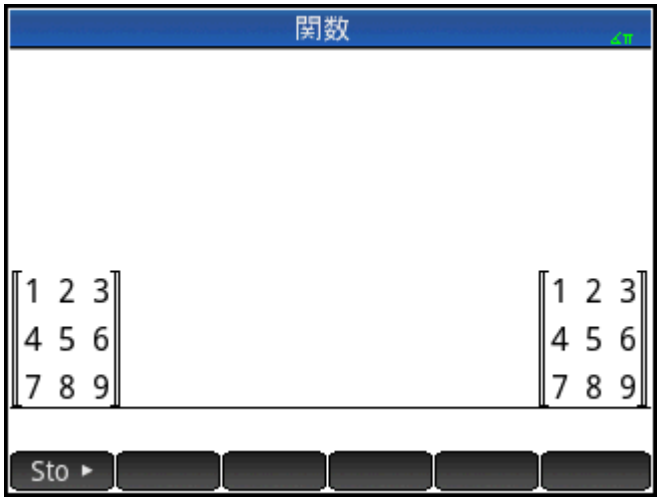


2. 正方形に値を入力します。次に、 ▶ を押して同じ行に 2 番目の値を入力するか、 + Ans ; を押して行を追加します。ユーザーが値を入力するたびに行列が拡張され、必要に応じて行および列が追加されます。

3. いつでも行列を拡張し、列および行を必要に応じて追加することができます。また、行全体または列全体を削除することもできます。行または列の末尾にある±記号の位置にカーソルを移動します。次に、 + Ans ; を押して新しい行または列を挿入するか、 − Base ; を押して行または列を削除します。 Del を押して行または列を削除することもできます。上の図では、 Del を押すと、行列の 2 番目の行が削除されます。



4. 操作が完了したら、 Enter を押すと、行列が履歴に表示されます。その後、行列を使用したり行列に名前を付けたりできます。



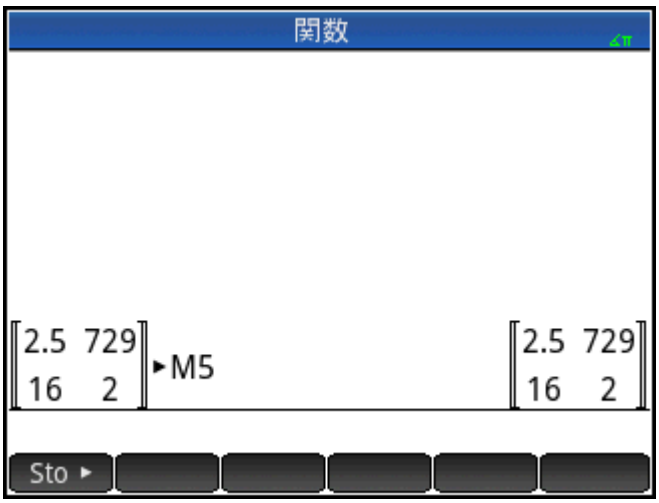
行列の格納

ベクトルまたは行列を変数に格納できます。ベクトルまたは行列は、履歴に追加する前に格納することも、履歴からコピーすることもできます。入力行へのベクトルまたは行列の入力、または履歴から入力行へベクトルまたは行列のコピーが完了したら、 Sto ▸ をタップしてその名前を入力し、 Enter を押します。ベクトルおよび行列用に予約されている変数名は、M0 から M9 です。常に独自の変数名を使用して、ベクトルや行列を格納できます。新しい変数は、**[変数]**メニューの ユーザ の下に表示されます。

以下の画面は、行列

$$\begin{bmatrix} 2.5 & 729 \\ 16 & 2 \end{bmatrix}$$

が M5 に格納されていることを示しています。行列の要素として式（5/2 など）を入力することが可能で、式は入力時に評価されます。



以下の図は、ベクトル[1 2 3]をユーザー変数 M25 に格納する様子を示したものです。ユーザー独自の変数を作成するかどうか確認するよう求められます。 OK をタップして処理を進めるか、キャンセをタップしてキャンセルします。

OK をタップすると、新しい行列が M25 という名前で格納されます。この変数は、**[変数]**メニューの[ユーザー]セクションに表示されます。また、Matrix Catalog（行列カタログ）にも新しい行列が表示されます。

行列の表示

ホーム表示で、ベクトルまたは行列の名前を入力し、 Enter ≈ を押します。ベクトルまたは行列が空の場合は、2 重角かっこで囲まれたゼロが返されます。

1 つの要素の表示

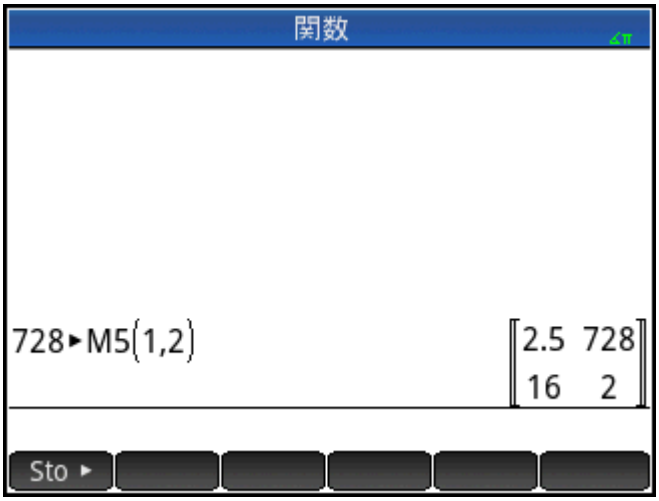
ホーム表示で、「matrixname(row,column)」と入力します。たとえば、`M2` が`[[3,4],[5,6]]`の場合、`M2(1,2)` Enter ≈ は `4` を返します。

1 つの要素の格納

ホーム表示で、値を入力して Sto ▸ をタップしてから、「*matrixname(row,column)*」と入力します。

たとえば、M5 の 1 行目の 2 列目にある要素を 728 に変更し、その結果の行列を表示するには、以下の操作を行います。

728 Sto ▸ ALPHA alpha +/− 5 ( ) 1 , Eval 2 Enter



行列のサイズを超える行または列に要素を格納しようとした場合、格納できるように行列のサイズが変更されます。中間のセルにはすべてゼロが設定されます。

行列の参照

`M1(1,2)` は、行列 M1 の 1 行目の 2 列目の値を返します。`M1(1)` は、M1 の 1 行目をベクトルとして返します。ホーム表示では、`M1(-1)` は M1 の 1 列目をベクトルとして返します。CAS 表示では、このコマンドに負の引数を使用することはできません。

`M1({1,2})` は、M1 の最初の 2 行を返します。`M1({1,1},{2,2})` は、1 行目の 1 列目の要素から 2 行目の 2 列目の要素までの部分行列を抽出します。M1 がベクトルの場合、`M1({1,3})` は、最初の 3 つの要素を含む部分ベクトルを抽出します。

行列の送信

行列は、アプリ、プログラム、リスト、および注記と同じように、電卓間で送信できます。手順について詳しくは、「データの共有」を参照してください。

行列の算術計算

算術関数（+、 −、×、÷、およびべき乗）では、行列の引数を使用できます。除算では、除数の逆行列を左から乗算します。行列自体を入力することも、格納済み行列変数の名前を入力することもできます。行列は実数または複素数にすることができます。

以降の例で使用するために、[[1,2],[3,4]]を M1 に、[[5,6],[7,8]]を M2 に格納します。

1. 最初の行列を選択します。

   Shift 4 Matrix （行列）

2. 行列の要素を入力します。

→へ移動 1 Enter 2 Enter 3 Enter 4 Enter

3. 最初の行列を選択します。

Shift 4 Matrix U （行列）

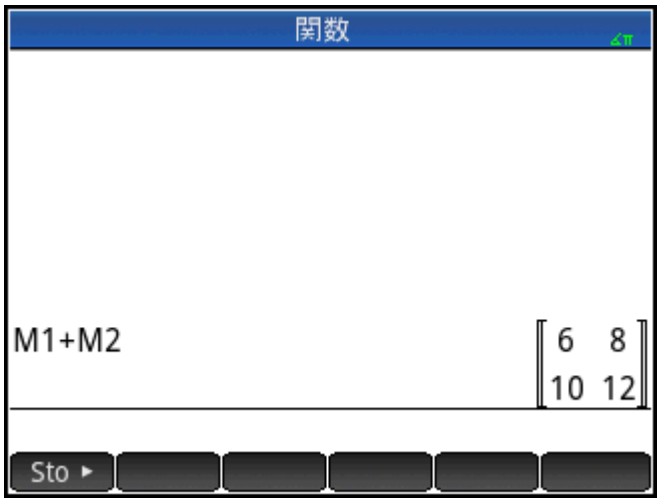
[M2]をタップまたは選択して、 Enter を押します。

4. 行列の要素を入力します。

5 Enter 6 Enter 7 Enter 8 Enter

5. ホーム表示で、作成したばかりの 2 つの行列を加算します。

Settings ALPHA alpha +/− 1 + Ans ; ALPHA alpha +/− 2 Enter ≈

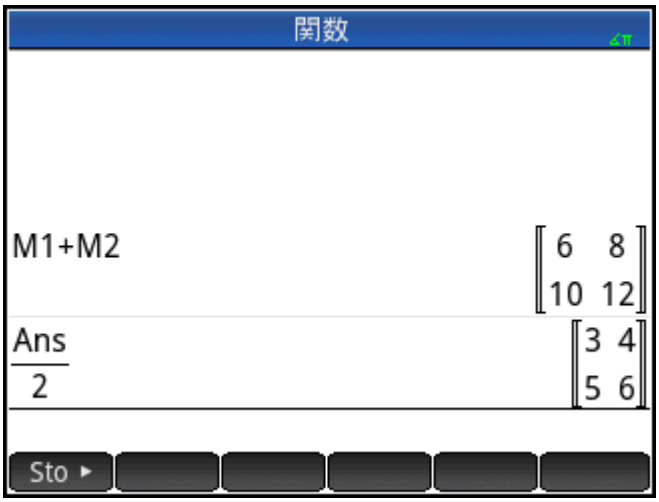


スカラーによる乗算および除算

スカラーによる除算では、まず行列を入力してから、演算子、スカラーの順に入力します。乗算では、オペランドの順序を気にする必要はありません。

行列およびスカラーは、実数または複素数にすることができます。たとえば、前の例の結果を 2 で除算するには、次のキーを押します。

÷ 2 Enter ≈

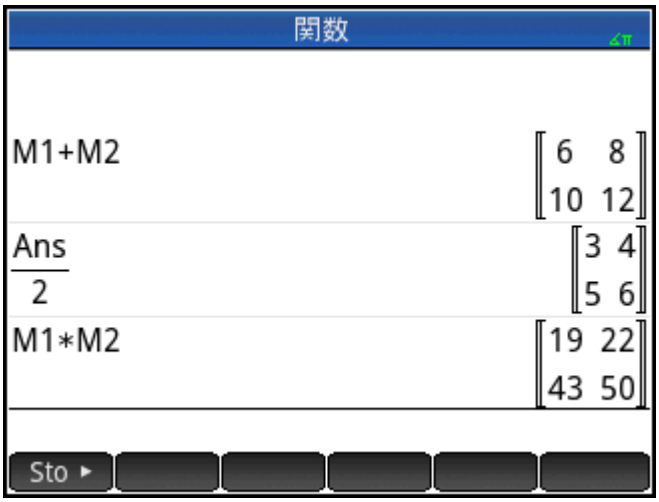


2 つの行列の乗算

前の例で作成した 2 つの行列を乗算するには、次のキーを押します。

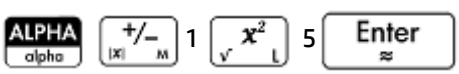
ALPHA alpha +/− 1 × ALPHA alpha +/− 2 Enter ≈

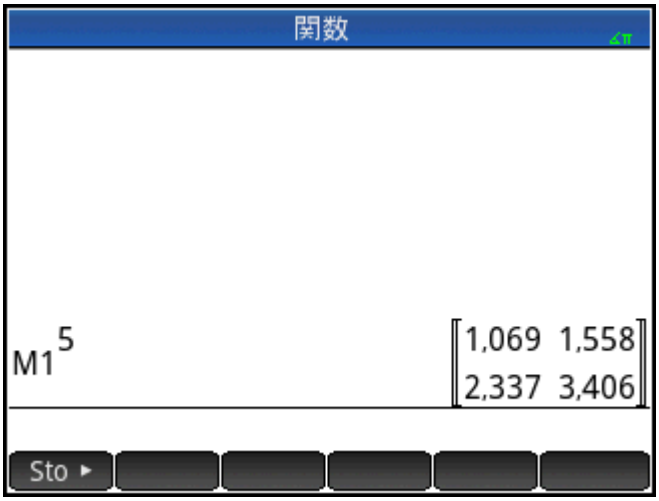
行列をベクトルで乗算するには、行列を入力してから、ベクトルを入力します。ベクトルの要素数が行列の列数と等しくなっている必要があります。



行列のべき乗の計算

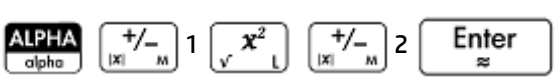
行列のべき乗は、整数乗であればどのようなものでも可能です。次の例は、先ほど作成した行列 M1 を 5 乗した結果を示したものです。





行列は、最初に変数として格納していなくてもべき乗できます。

行列では、負のべき乗も実行できます。この場合、結果は 1/ [matrix]^ABS(power)に相当します。次の例では、M1 の – 2 乗を計算しています。



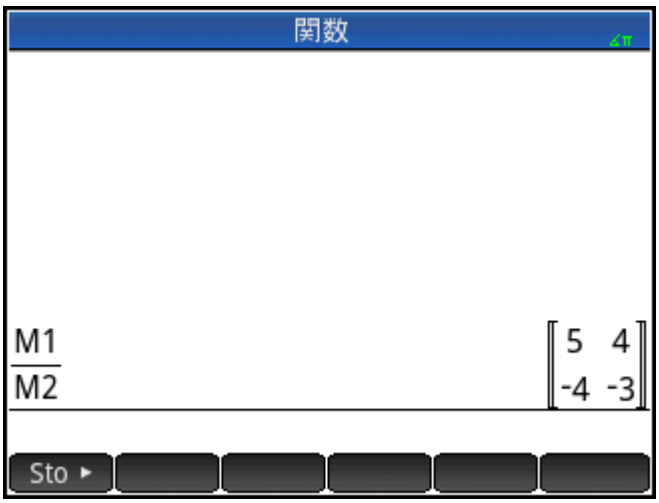
正方行列による除算

行列またはベクトルを正方行列で除算する場合、被除数の行数（ベクトルの場合は要素数）を除数の行数と等しくする必要があります。

この演算は数学的な除算ではありません。これは、除数の逆行列を左から乗算したものです。M1/M2 は $M2^{-1} * M1$ に相当します。

前の例で作成した 2 つの行列を除算するには、次のキーを押します。

ALPHA alpha +/− 1 ÷ ALPHA alpha +/− 2



行列の逆行列の計算

ホーム表示で正方行列の逆行列を求めるには、行列（またはその変数名）を入力し、Shift ÷ Enter ≈ を押します。[数学]メニューの[行列]カテゴリにある INVERSE コマンドを使用することもできます。

各要素の符号の反転

行列の各要素の符号を変更するには、 +/− を押し、行列名を入力して Enter を押します。

連立一次方程式の計算

行列を使用して、次のような連立一次方程式を解くことができます。

2x+3y+4z=5

x+y-z=7

4x-y+2z=1

この例では行列 M1 および行列 M2 を使用しますが、利用可能なすべての行列変数名を使用できます。

この例では行列 M1 および行列 M2 を使用しますが、利用可能なすべての行列変数名を使用できます。

1. Matrix Catalog（行列カタログ）を開いて M1 をクリアし、ベクトルの作成を選択してから、Matrix（行列）エディターを開きます。

   Shift 4 Matrix U （ ▲ または ▼ を押して M1 を選択） Del OK ベクトル

   Enter

2. 連立一次方程式の 3 つの定数を含むベクトルを作成します。

5 Enter ≈ 7 Enter ≈ 1 Enter ≈

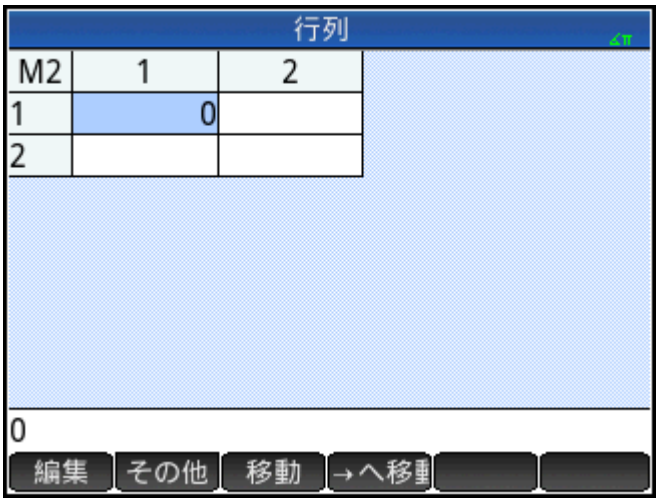
3. Matrix Catalog（行列カタログ）に戻ります。

Shift 4 Matrix U

M1 のサイズが 3 と表示されているはずです。

4. M2 を選択してクリアし、Matrix（行列）エディターをもう一度開きます。

（ ▼ または ▲ を押して M2 を選択） Del OK Enter



5. 方程式の係数を入力します。

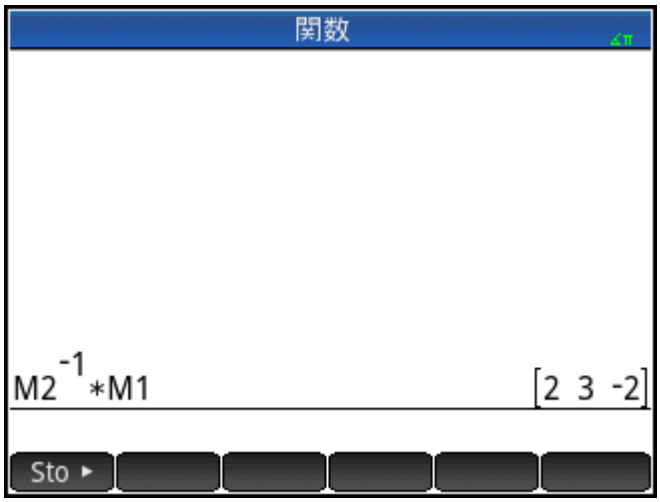
2 Enter 3 Enter （セル R1, C3 をタップ）4 Enter 1 Enter 1 Enter

+/− 1 Enter 4 Enter +/− 1 Enter 2 Enter

6. ホーム表示に戻り、係数行列の逆行列を左から定数ベクトルに乗算します。

Settings ALPHA alpha +/− 2 Shift ÷ × ALPHA alpha +/− 1 Enter

結果は、解を含むベクトルになります（x = 2、y = 3、z = −2）。



別の方法として、RREF 関数を使用することもできます（「RREF」セクションを参照してください）。

行列の関数およびコマンド

関数

関数は任意のアプリケーションやホーム表示で使用できます。関数は、[数学]メニューの[行列]カテゴリの下に一覧表示されます。おもにホーム表示の数式で使用できますが、プログラム内でも使用できます。

関数は常に結果を生成して表示します。行列変数などの格納された変数は、一切変更しません。

関数の引数はかっこで囲み、コンマで区切ります（例：CROSS(*vector1*,*vector2*)）。行列入力は、行列変数名（M1 など）か、実際の行列データを角かっこで囲んだものになります（例：CROSS(M1,[1 2])）。

メニュー形式

初期設定では、行列関数は、一般的なコマンド名ではなく説明的な名前で[数学]メニューに表示されます。したがって、短縮名 TRN は**[転置]**と表示され、DET は**[行列式]**と表示されます。

[数学]メニューに代わりにコマンド名が表示されるようにするには、[ホーム設定]画面の 2 ページ目で**[メニュー表示]**オプションの選択を解除します。

コマンド

行列コマンドが行列関数と異なる点は、結果を返さないことです。このため、行列関数は、行列コマンドが使用できない式で使用できます。行列コマンドは、行列を使用するプログラムをサポートするように設計されています。

行列コマンドは、プログラム エディターのコマンド メニューの[行列]カテゴリに一覧表示されます。また、[カタログ]メニュー（ツールボックス メニューの 1 つ）にも一覧表示されます。Settings を押してカタログをタップすると、コマンドのカタログが表示されます。行列関数については、この章の

以下の各セクションで説明します。行列コマンドについては、プログラミングの章で説明します (544 ページを参照してください)。

引数の表記法

- *row#*または *column#*で、行数（一番上からカウント、1 から開始）または列数（一番左からカウント、1 から開始）を指定します。
- 引数 *matrix* は、ベクトルまたは行列を表します。

行列関数

行列関数は、[数学]メニューの[行列]カテゴリから利用できます。 **[行列]**を選択します。関数を選択します。

[行列]

[転置]

行列を転置します。複素数の行列の場合、TRN は共役転置行列を求めます。

```
TRN(matrix)
```

例：

$\mathrm{TRN}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は $\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$ を返します

[行列式]

正方行列の行列式。

```
DET(matrix)
```

例：

$\mathrm{DET}\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$ は -2 を返します

[RREF]

行既約階段形。長方形行列を RREF に変更します。

```
RREF(matrix)
```

例：

$\mathrm{RREF}\left(\begin{bmatrix}1 & -2 & 1\\ 3 & 4 & -1\end{bmatrix}\right)$ は $\begin{bmatrix}1 & 0 & 0.2\\ 0 & 1 & -0.4\end{bmatrix}$ を返します

[作成]

[作成]

rows × columns のサイズの行列を作成します（expression を使って各要素を計算します）。式に変数 I および J が含まれている場合、各要素の計算時に、I は現在の行番号に、J は現在の列番号にそれぞれ置き換えられます。また、行数および列数の代わりに要素数を指定してベクトルを作成することもできます。

```
MAKEMAT(expression, rows, columns)
```

```
MAKEMAT(expression, elements)
```

例：

`MAKEMAT(0,3,3)` は 3 × 3 ゼロ行列[[0,0,0],[0,0,0],[0,0,0]]を返します。

`MAKEMAT(√2,2,3)` は 2 × 3 行列[[√2,√2,√2],[√2,√2,√2]]を返します。

`MAKEMAT(I+J-1,2,3)` は 2 × 3 行列[[1,2,3],[2,3,4]]を返します

上の例で、各要素が行番号と列番号の和から 1 を引いたものになっていることに注意してください。

[恒等]

恒等行列。対角要素が 1、対角要素以外が 0 であるような、size × size 次元の正方行列を作成します。

```
IDENMAT(size)
```

[乱数]

2 つの整数 n と m および行列名が指定された場合、-99 から 99 の範囲で均一に分布するランダム整数を含む n x m 行列を作成し、指定された行列名に格納します。整数が 1 つだけ指定された場合は、ランダム整数が設定された、その長さのベクトルを返します。オプションの追加の整数ペアが指定された場合、それらの整数で定義される区間に限定された乱数を含む行列を返します。

```
randMat([MatrixName],n,[m], [lower, upper})
```

例：

`RANDMAT(M1,2,2)` は、ランダム整数を含む 2x2 行列を返し、それを M1 に格納します。

[ジョルダン]

対角が expr、その上が 1、他のすべてが 0 の n x n 正方行列を返します。

```
JordanBlock(Expr,n)
```

例：

`JordanBlock(7,3)` は $\begin{bmatrix} 7 & 1 & 0 \\ 0 & 7 & 1 \\ 0 & 0 & 7 \end{bmatrix}$ を返します

[ヒルベルト]

正の整数 n が与えられたときに、第 n 次のヒルベルト行列を返します。行列の各要素は式 1/(j+k-1)で与えられます（j は行番号、k は列番号）。

hilbert(n)

例：

[CAS 表示]で、hilbert(4)は $\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$ を返します。

[等長]

適切な要素によって指定される等長の行列。

mkisom(vector,sign(1 または-1))

例：

[CAS 表示]で、mkisom([1,2],1)は $\begin{bmatrix} \cos(1) & -\sin(1) \\ \sin(1) & \cos(1) \end{bmatrix}$ を返します。

[ファンデルモンド]

ファンデルモンド行列を返します。ベクトル[n1, n2 … nj]が与えられたときに、1 行目が[$(n1)^0$, $(n1)^1$, $(n1)^2$, …,$(n1)^{j-1}$]となる行列を返します。2 行目は、[$(n2)^0$, $(n2)^1$, $(n2)^2$, …,$(n2)^{j-1}$]となります。

vandermonde(vector)

例：

vandermonde([1 3 5])は $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \end{bmatrix}$ を返します

[基本]

[ノルム]

行列のフロベニウス ノルムを返します。

|matrix|

例：

$\left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \right\|$ は 5.47722557505 を返します

[行ノルム]

行ノルム。行のすべての要素の絶対値の合計の（すべての行での）最大値を求めます。

ROWNORM(matrix)

例：

ROWNORM$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 7 を返します

[列ノルム]

列ノルム。列のすべての要素の絶対値の合計の（すべての列での）最大値を求めます。

```
COLNORM(matrix)
```

例：

COLNORM$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 6 を返します

[スペクトル ノルム]

正方行列のスペクトル ノルム。

```
SPECNORM(matrix)
```

例：

SPECNORM$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 5.46498570422 を返します

[スペクトル半径]

正方行列のスペクトル半径。

```
SPECRAD(matrix)
```

例：

SPECRAD(matrix)$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 5.37228132327 を返します

[条件]

条件数。正方行列の 1 ノルム（列ノルム）を求めます。

```
COND(matrix)
```

例：

COND$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 21 を返します

[階数]

長方形行列の階数。

```
RANK(matrix)
```

例：

RANK$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 2 を返します

[ピボット]

行列、行番号 n、および列番号 m が与えられたときに、ガウスの消去法を使用して、列 m にゼロを持つ行列を返しますが、列 m および行 n の要素はピボットとして保持されます。

```
pivot(matrix,n,m)
```

例：

$\mathrm{pivot}\left(\begin{bmatrix}1 & 2\\3 & 4\\5 & 6\end{bmatrix}, 1, 1\right)$ は $\begin{bmatrix}1 & 2\\0 & -2\\0 & -4\end{bmatrix}$ を返します

[トレース]

正方行列のトレースを求めます。トレースは対角要素の和に等しくなります（固有値の和にも等しくなります）。

```
TRACE(matrix)
```

例：

$\mathrm{TRACE}\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は 5 を返します

[高度]

[固有値]

行列の固有値をベクトル形式で表示します。

```
EIGENVAL(matrix)
```

例：

$\mathrm{EIGENVAL}\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は[5.37228... − 0.37228...]を返します

[固有ベクトル]

正方行列の固有ベクトルおよび固有値。2 つの配列のリストを表示します。1 つ目には固有ベクトルが含まれ、2 つ目には固有値が含まれます。

```
EIGENVV(matrix)
```

例：

$\mathrm{EIGENVV}\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)$ は次の行列を返します。

$$\left\{\begin{bmatrix}0.4159\ldots & -0.8369\ldots\\0.9093\ldots & 0.5742\ldots\end{bmatrix}, \begin{bmatrix}5.3722\ldots & 0\\0 & -0.3722\ldots\end{bmatrix}\right\}$$

[ジョルダン]

通過行列と行列のジョルダン標準形によって作成されたリストを返します。

jordan(matrix)

例：

jordan は $\left[\begin{bmatrix}\sqrt{2} & -\sqrt{2}\\ 1 & 1\end{bmatrix}, \begin{bmatrix}\sqrt{2} & 0\\ 0 & -\sqrt{2}\end{bmatrix}\right]$ を返します

[対角]

リストが指定された場合、対角要素がリストの要素で他の要素がすべてゼロであるような行列を返します。行列が指定された場合、その対角要素を含むベクトルを返します。

diag(list) または diag(matrix)

例：

diag$\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は [1 4] を返します

[コレスキー]

数値的な対称行列 A について、A=L*tran(L)を満たす行列 L を返します。

cholesky(matrix)$\left(\begin{bmatrix}3 & 1\\ 1 & 4\end{bmatrix}\right)$

例：

[CAS 表示]で、cholesky は簡略化後の $\left(\begin{bmatrix}\sqrt{3} & 0\\ \frac{\sqrt{3}}{3} & \frac{\sqrt{33}}{3}\end{bmatrix}\right)$ を返します

[エルミート]

Z の係数を持つ行列のエルミート標準形。U は Z で可逆、B は上三角行列で、B=U*A となるような U、B を返します。

ihermite(Mtrx(A))

例：

ihermite$\left(\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}\right)$ は $\left[\begin{bmatrix}-3 & 1 & 0\\ 4 & -1 & 0\\ -1 & 2 & -1\end{bmatrix}, \begin{bmatrix}1 & -1 & -3\\ 0 & 3 & 6\\ 0 & 0 & 0\end{bmatrix}\right]$ を返します

[ヘッセンベルグ]

行列のヘッセンベルグ形式への簡略化。B=inv(P)*A*P を満たす[P,B]を返します。

hessenberg(Mtrx(A))

例：

[CAS 表示]で、hessenberg $\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right)$ は $\begin{bmatrix}\begin{bmatrix}1&0&0\end{bmatrix} & \begin{bmatrix}0&\frac{4}{7}&1\end{bmatrix} & \begin{bmatrix}0&1&0\end{bmatrix}\\ \begin{bmatrix}1&\frac{29}{7}&2\end{bmatrix} & \begin{bmatrix}7&\frac{39}{7}&8\end{bmatrix} & \begin{bmatrix}0&\frac{278}{49}&\frac{3}{7}\end{bmatrix}\end{bmatrix}$ を返します

[スミス]

Z の係数を持つ行列のスミス標準形。U および V は Z で可逆、B は対角行列で、B[i,i]は B[i+1,i+1]を分割し、B=U*A*V であるような U、B、V を返します

```
ismith(Mtrx(A))
```

例：

ismith $\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right)$ は $\begin{bmatrix}\begin{bmatrix}1&0&0\\4&-1&0\\-1&2&-1\end{bmatrix} & \begin{bmatrix}1&0&0\\0&3&0\\0&0&0\end{bmatrix} & \begin{bmatrix}1&-2&1\\0&1&-2\\0&0&1\end{bmatrix}\end{bmatrix}$ を返します

[因数分解]

[LQ]

LQ 因数分解。*m* × *n* 行列を 3 つの行列 L、Q、および P に因数分解します。ここで、*{[L[m × n の下台形行列]],[Q[n × n の直交行列]],[P[m × m の置換行列]]}*で、P*A=L*Q となります。

```
LQ(matrix)
```

例：

LQ $\left(\begin{bmatrix}1&2\\3&4\end{bmatrix}\right)$ は $\left\{\begin{bmatrix}2.2360\ldots & 0\\4.9193\ldots & 0.8944\ldots\end{bmatrix},\begin{bmatrix}0.4472\ldots & 0.8944\ldots\\0.8944\ldots & -0.4472\ldots\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right\}$ を返します

[LSQ]

最小 2 乗。系 matrix1*X=matrix2 に対応する最小ノルム最小 2 乗行列（またはベクトル）を表示します。

```
LSQ(matrix1, matrix2)
```

例：

LSQ $\left(\begin{bmatrix}1&2\\3&4\end{bmatrix},\begin{bmatrix}5\\11\end{bmatrix}\right)$ は $\begin{bmatrix}1\\2\end{bmatrix}$ を返します

[LU]

LU 分解。正方行列を 3 つの行列 L、U、および P に因数分解します。ここで、*{[L[下三角行列]],[U[上三角行列]],[P[置換行列]]}*で、P*A=L*U となります。

```
LU(matrix)
```

例：

LU $\left(\begin{bmatrix}1&2\\3&4\end{bmatrix}\right)$ は $\left\{\begin{bmatrix}1&0\\0.3333\ldots&1\end{bmatrix},\begin{bmatrix}3&4\\0&0.6666\ldots\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right\}$ を返します

[QR]

QR 因数分解。*m*×*n* 行列 A を数値的に Q*R に因数分解します。ここで、Q は直交行列、R は上三角行列で、R を返します。R は var2 に格納され、Q=A*inv(R)は var1 に格納されます。

```
QR(matrix A,var1,var2)
```

例：

$\mathrm{QR}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は $\left\{\begin{bmatrix}0.3612\ldots & 0.9486\ldots\\ 0.9486\ldots & -0.3162\ldots\end{bmatrix}, \begin{bmatrix}3.1622\ldots & 4.4271\ldots\\ 0 & 0.6324\ldots\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right\}$ を返します

[シュール]

シュール分解。正方行列を 2 つの行列に因数分解します。*matrix* が実数である場合、結果は*{[[直交行列]],[[上疑似三角行列]]}*になります。*matrix* が複素数の場合、結果は*{[[単位行列]],[[上三角行列]]}*になります。

```
SCHUR(matrix)
```

例：

$\mathrm{SCHUR}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は $\left\{\begin{bmatrix}0.4159\ldots & 0.9093\ldots\\ 0.9093\ldots & 0.4159\ldots\end{bmatrix}, \begin{bmatrix}5.3722\ldots & 1\\ 5.55\times10^{-17} & -0.3722\end{bmatrix}\right\}$ を返します

[SVD]

特異値分解。*m* × *n* 行列を 2 つの行列と 1 つのベクトルに因数分解します：*{[[m × m 正方直交行列]], [実数], [[n × n 正方直交行列]]}*。

```
SVD(matrix)
```

例：

$\mathrm{SVD}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は $\left\{\begin{bmatrix}0.4045\ldots & -0.9145\ldots\\ 0.9145\ldots & 0.4045\ldots\end{bmatrix}, [5.4649\ldots\ 0.3659\ldots], \begin{bmatrix}0.5760\ldots & 0.8174\ldots\\ 0.8174\ldots & -0.5760\end{bmatrix}\right\}$ を返します

[SVL]

特異値。行列の特異値を含むベクトルを返します。

```
SVL(matrix)
```

例：

$\mathrm{SVL}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$ は `[5.4649... 0.3659...]` を返します

[ベクトル]

[クロス積]

vector1 と *vector2* のクロス積。

```
CROSS(vector1, vector2)
```

例：

CROSS ([1 2],[3 4])は[0 0 -2]を返します

[ドット積]

2 つのベクトル vector1 と vector2 のドット積。

dot(vector1, vector2)

例：

dot([1 2],[3 4])は 11 を返します

[L²Norm]

ベクトルの l² ノルム（sqrt(x1^2+x2^2+...xn^2)）を返します。

l2norm(Vect)

例：

l2norm([3 4 -2])は√29 を返します

[L¹Norm]

ベクトルの l¹ ノルム（座標の絶対値の和）を返します。

l1norm(Vect)

例：

l1norm([3 4 -2])は 9 を返します

[最大ノルム]

ベクトルの l∞ノルム（座標の絶対値の最大）を返します。

maxnorm(Vect または Mtrx)

例：

maxnorm([1 2 3 -4])は 4 を返します

例

恒等行列

恒等行列は IDENMAT 関数で作成できます。たとえば、IDENMAT(2)は 2×2 恒等行列[[1,0],[0,1]]を作成します。

また、MAKEMAT（行列の作成）関数を使って恒等行列を作成することもできます。たとえば、MAKEMAT(I ≠J,4,4)の場合、対角要素がゼロ、他のすべての要素が数値 1 の 4 × 4 行列が作成されます。論理演算子（≠）は、I（行番号）と J（列番号）が等しい場合は 0 を返し、等しくない場合は 1 を返します（≠を挿入するには、関係パレット Shift 6 から選択します）。

行列の転置

TRN 関数は、行列の行 – 列の要素と列 – 行の要素を入れ替えます。たとえば、要素 1,2（行 1、列 2）と要素 2,1 が入れ替わり、要素 2,3 と要素 3,2 が入れ替わります。

たとえば、TRN([[1,2],[3,4]])は、行列[[1,3],[2,4]]を作成します。

行既約階段形

連立方程式

x − 2y + 3z = 14

2x + y − z = -3

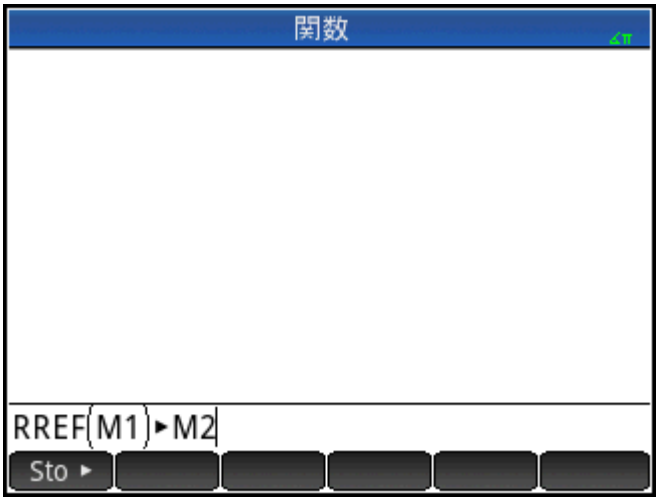
4x − 2y + 2z = 14

は、次の拡大行列として記述できます

$$\left[\begin{array}{ccc|c} 1 & -2 & 3 & 14 \\ 2 & 1 & -1 & -3 \\ 4 & -2 & 2 & 14 \end{array}\right]$$

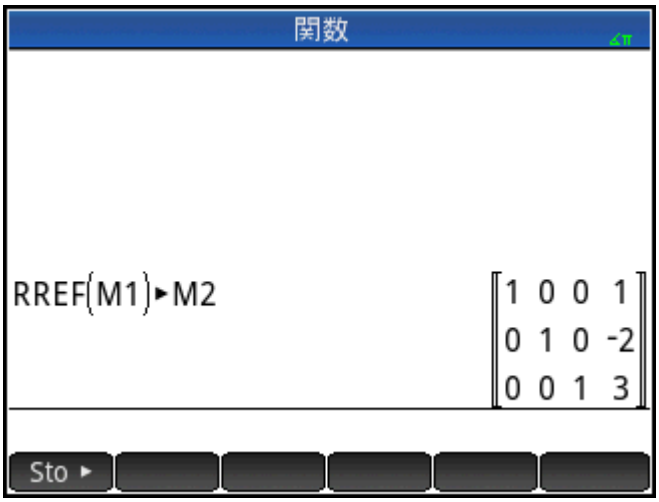
この行列は、3 x 4 実行列として任意の行列変数に格納できます。この例では、M1 を使用します。

次に、RREF 関数を使用してこれを行既約階段形に変更し、結果を任意の行列変数に格納できます。この例では、M2 を使用します。



行既約階段行列の 4 番目の列が、1 次方程式の解になります。

RREF 関数を使用する利点は、解がないか解が無限にあるような連立方程式に対応する不整合な行列でも動作することです。



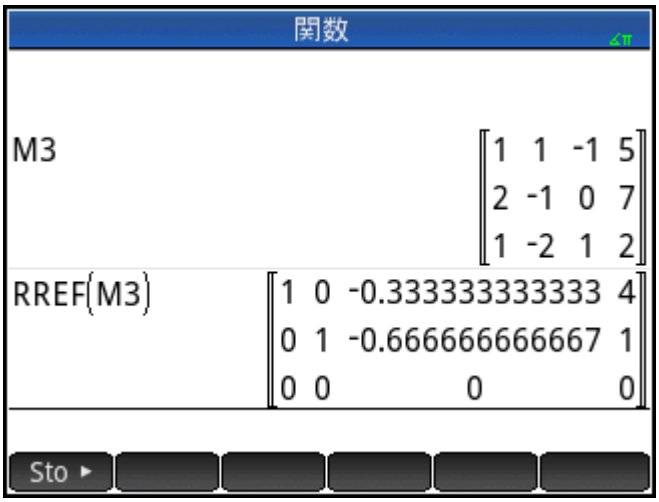
たとえば、次の連立方程式は無限個の解を持ちます。

x + y − z = 5

2x − y = 7

x − 2y + z = 2

拡大行列の行既約階段形で、最後の行のゼロは無限の解を持つ不整合な系であることを示しています。



27 注記および情報

HP Prime には、注記を入力するためのテキスト エディターが 2 つあります。

- Note（注記）エディター：アプリケーションから独立した注記の集合である Note Catalog（注記カタログ）内から開きます。
- Info（情報）エディター：アプリケーションの[情報表示]から開きます。[情報表示]で作成された注記はアプリケーションに関連付けられており、アプリケーションを別の電卓に送信した場合も保持されます。

Note Catalog（注記カタログ）

メモリが許す限り、Note Catalog（注記カタログ）には必要な数の注記を保存できます。これらの注記は、どのアプリケーションからも独立しています。Note Catalog（注記カタログ）では、注記が名前順に一覧表示されます。この一覧には、アプリケーションの[情報表示]で作成された注記は含まれませんが、それらの注記をクリップボード経由で Note Catalog（注記カタログ）にコピーして貼り付けることはできます。Note Catalog（注記カタログ）から、それぞれの注記を Note（注記）エディターで作成または編集します。

Note Catalog（注記カタログ）：ボタンおよびキー

Shift 0 Notes（注記）を押すと、Note Catalog（注記カタログ）が表示されます。Note Catalog（注記カタログ）内では、次のボタンおよびキーを利用できます。Note Catalog（注記カタログ）に注記が 1 つもない場合、一部のボタンは利用できません。

ボタンまたはキー	用途
編集	編集するために選択された注記を開きます
新しい	新しい注記を開始します。名前を入力するよう求められます
その他	タップすると、以下の追加機能が表示されます。 **[保存]：**選択した注記のコピーを作成します。新しい名前で保存するよう求められます **[名前の変更]：**選択した注記の名前を変更します **[並べ替え]：**注記のリストを並べ替えます（並べ替えのオプションはアルファベット順および作成順です） **[削除]：**すべての注記を削除します **[クリア]：**選択した注記のコピーを作成します。新しい名前で保存するよう求められます **[送信]：**選択した注記を別の HP Prime 電卓に送信します。

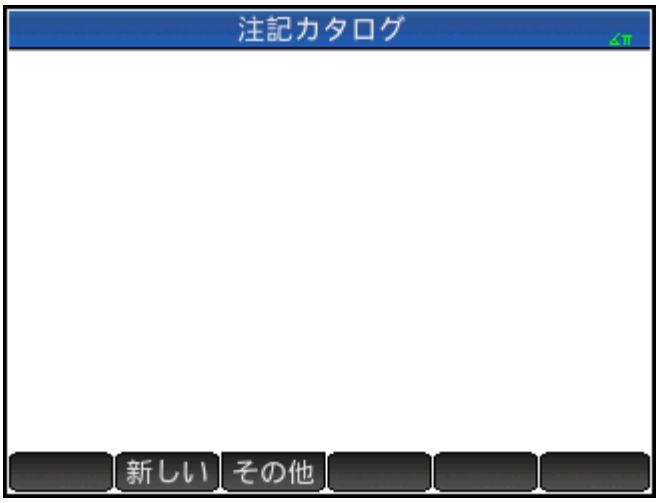
ボタンまたはキー	用途
	選択した注記を削除します
	カタログ内のすべての注記を削除します

Note（注記）エディター

Note（注記）エディターは注記を作成および編集する場所です。Note（注記）エディターを Note（注記）カタログから起動したり、アプリケーション内から起動したりすることもできます。アプリケーション内で作成された注記は、アプリケーションを別の電卓に送信した場合でも、そのアプリケーションと一緒に移動します。このような注記は、Note（注記）カタログには表示されません。関連付けられたアプリケーションが開いているときにのみ読むことができます。Note（注記）カタログから作成された注記は、アプリケーション固有ではないため、Note（注記）カタログを開くことでいつでも見ることができます。これらの注記を別の電卓に送信することもできます。

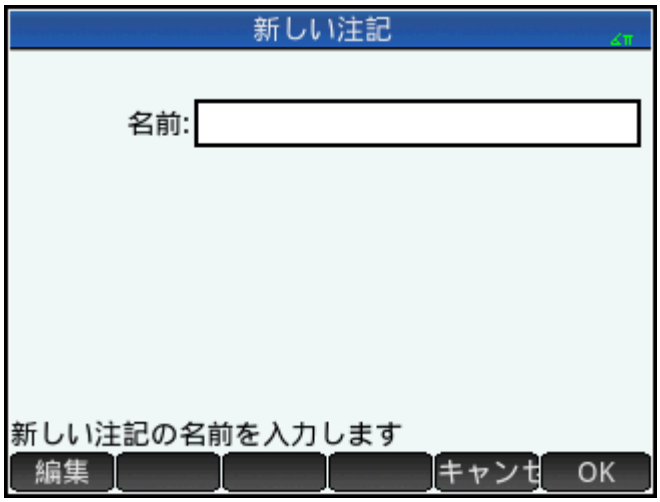
Note（注記）カタログから注記を作成するには、以下の操作を行います。

1. Note（注記）カタログを開きます。



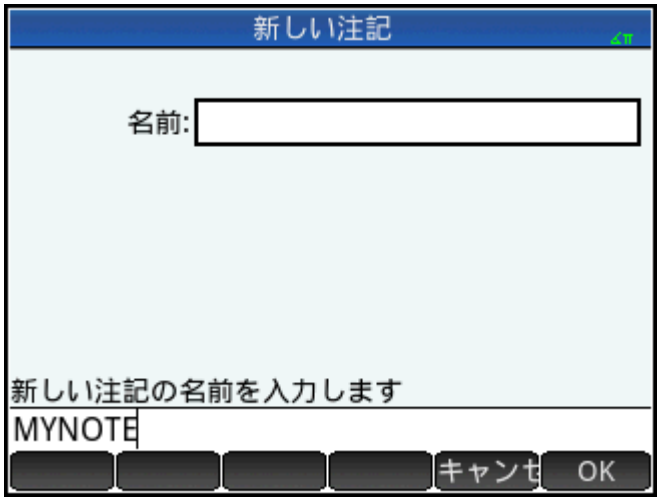
2. 新しい注記を作成します。

新しい



3. 注記の名前を入力します。この例では、注記の名前を MYNOTE とします。 ALPHA alpha ALPHA alpha

MYNOTE OK OK



4. 以下のセクションで説明する編集キーおよび形式オプションを使用して、注記を書き込みます。書き込みが終了したら、 Settings を押すか、 Apps Info を押してアプリケーションを開くことで、Note（注記）エディターを終了します。注記は自動的に保存されます。新しい注記にアクセスするには、Note（注記）カタログに戻ります。

アプリケーションの注記の作成

アプリケーション固有の注記を作成したり、アプリケーションを別の電卓に送信してもアプリケーションと一緒に移動する注記を作成したりすることもできます。この方法で作成された注記には、Note（注記）エディターのすべての形式機能が適用されます（以下を参照）。

Note（注記）エディター：ボタンおよびキー

以下のボタンおよびキーは、注記を追加または編集するとき利用できます。

ボタンまたはキー	用途
形式	テキスト形式メニューを開きます。538 ページの形式オプションを参照してください
スタイル	太字、斜体、下線、大文字、上付き文字、および下付き文字の各種オプションを表示します。538 ページの形式オプションを参照してください
•	3 種類の箇条書きを提供する切り替えボタンです。538 ページの形式オプションを参照してください
挿入	数式をテキストブック形式で入力するために 2D エディターを起動します。539 ページの数式の挿入を参照してください
⎵	テキスト入力中にスペースを入力します
ページ	複数ページの注記内で別のページに移動します
Shift View Copy	注記内のテキストをコピーする際のオプションを表示します。以下を参照してください
開始	コピー オプション。テキスト選択の開始位置をマークします

ボタンまたはキー	用途
終了	コピー オプション。テキスト選択の終了位置をマークします
すべて	コピー オプション。注記全体を選択します
切取	コピー オプション。選択したテキストを切り取ります
コピー	コピー オプション。選択したテキストをコピーします
Del	カーソルの左にある文字を削除します
Enter ≈	新しい行を開始します
Shift Esc Clear（クリア）：	注記全体を消去します
Vars Chars A	変数名および変数の内容を入力するためのメニュー
Mem B	数学コマンドを入力するためのメニュー
Shift Vars Chars A（クリア）：	特殊文字のパレットを表示します。入力するには、選択して OK をタップするか Enter ≈ を押します。[文字]メニューを**閉じず**に文字をコピーするには、選択して エコー をタップします

大文字および小文字の入力

以下の表では、大文字および小文字をすばやく入力する方法について説明します。

キー	用途
ALPHA alpha	次の文字を大文字にします
ALPHA alpha ALPHA alpha	大文字にロックされているときに、次の文字を小文字にします
Shift	3 種類の箇条書きを提供する切り替えボタンです。538 ページの形式オプションを参照してください。
Shift ALPHA alpha	大文字にロックされているときに、モードがリセットされるまで、すべての文字を小文字にします
ALPHA alpha	大文字ロック モードをリセットします

キー	用途
ALPHA alpha Shift	次の文字を小文字にします
ALPHA alpha Shift ALPHA alpha	ロック モード：モードがリセットされるまで、すべての文字を小文字にします
Shift	小文字にロックされているときに、次の文字を大文字にします
Shift ALPHA alpha	小文字にロックされているときに、モードがリセットされるまで、すべての文字を大文字にします
ALPHA alpha	小文字ロック モードをリセットします

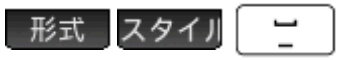
タイトル バーの通知領域の左側に、次に入力する文字に大文字小文字のどちらが適用されるかが示されます。

テキスト形式

Note（注記）エディターに異なる形式でテキストを入力できます。形式オプションはテキスト入力を開始する前に選択します。形式オプションについては、538 ページの形式オプションで説明します。

形式オプション

形式オプションは、Note（注記）エディターの 3 つのタッチ ボタンおよびアプリケーションの[情報表示]で利用できます。



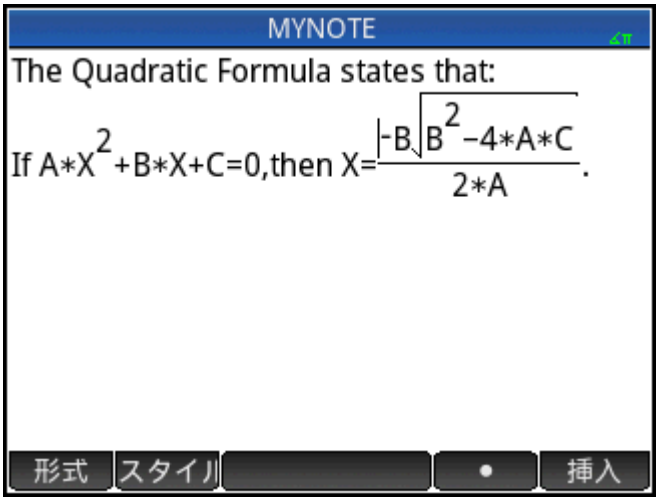
形式オプションの一覧を以下の表に示します。

カテゴリ	オプション
形式 フォント サイズ	10 ~ 22 pt.
形式 前景色	20 色から選択します。
形式 背景色	20 色から選択します。
形式 整列（テキスト整列）	左 中央

カテゴリ	オプション
	右
スタイル フォント スタイル	太字 斜体 下線 取り消し線 上付き文字 下付き文字
• 箇条書き	• —第 1 レベルの箇条書き ◦ —第 2 レベルの箇条書き ▷ —第 3 レベルの箇条書き ✗ —箇条書きをキャンセル

数式の挿入

次の図で示すように、数式をテキスト形式で注記に入力できます。[注記]エディターでは[ホーム]および[CAS]表示と同じ 2D エディターが使用され、 挿入 メニュー ボタンで開きます。



1. 目的のテキストを入力します。数式を開始する位置に来たら、 挿入 をタップします。
2. [ホーム表示]または[CAS 表示]と同様に数式を入力します。数学テンプレートと、[ツールボックス]メニュー内の関数を使用できます。
3. 数式の入力が終了したら、 ▶ を 2 回または 3 回（数式の複雑さに応じて）押してエディターを終了します。テキストの入力を継続することもできます。

注記をインポートするには、以下の操作を行います。

注記は、Note（注記）カタログから[情報表示]に、またはその逆方向にインポートできます。

ここでは、名前が Assignments の注記を Note（注記）カタログから[関数情報表示]にコピーします。

1. Note（注記）カタログを開きます。

   Shift

2. 注記 **Assignments** を選択して、 編集 をタップします。
3. クリップボードにコピーするためのコピーオプションを開きます。

   Shift View Copy （コピー）

   メニュー ボタンがコピー オプションに変わります。

   開始 ：コピーまたは切り取りを開始する位置をマークします。

   終了 ：コピーまたは切り取りを終了する位置をマークします。

   すべて ：プログラム全体を選択します。

   切取 ：選択範囲を切り取ります。

   コピー ：選択範囲をコピーします。
4. コピーまたは切り取るものを選択します（すぐ上のオプションを使用）。
5. コピー または 切取 をタップします。
6. Function（関数）アプリケーションの[情報ビュー]を開きます。

   Apps Info を押し、Function（関数）アプリケーションのアイコンをタップしてから、 Shift Apps Info を押します。
7. コピーしたテキストを貼り付ける位置までカーソルを移動して、クリップボードを開きます。

   Shift Menu Paste
8. クリップボードからテキストを選択して、 OK を押します。

注記を別の HP Prime に送信できます。

28 HP PPL でのプログラミング

この章では、HP Prime プログラミング言語（HP PPL）について説明します。この章では以下のことについて学習します。

- コマンドをプログラミングする
- プログラムで関数を記述する
- プログラムで変数を使用する
- プログラムを実行する
- プログラムをデバッグする
- カスタム アプリケーションを構築するためにプログラムを作成する
- プログラムを別の HP Prime に送信する

HP Prime プログラム

HP Prime プログラムには、タスクを実行するために自動的に実行する一連のコマンドが含まれます。

コマンドの構造

コマンドはセミコロン（;）で区切られます。複数の引数を取るコマンドでは、引数は括弧で囲まれ、コンマ（,）で区切られます。例、

`PIXON` (*xposition, yposition*);

コマンドの引き数はオプションの場合があります。引数が省略されている場合、その場所では初期設定値が使用されます。PIXON コマンドの場合、3 番目の引数を使用してピクセルの色を指定できます。

`PIXON` (*xposition, yposition [,color]*);

このマニュアルでは上記のように、コマンドのオプション引数は角かっこに囲まれて表記されます。PIXON の例では、グラフィックス変数（G）を 1 番目の引数として指定できます。初期設定は G0 で、現在表示されている画面を常に含みます。IXON コマンドの完全な構文は以下のとおりです。

`PIXON`(*[G,] xposition, yposition [,color]*);

関数引数をかっこで囲まない構文を使用する組み込みコマンドがあります。`RETURN` や `RANDOM` などです。

プログラムの構造

プログラムには任意の数のサブルーチン（それぞれは関数またはプロシージャー）を含めることができます。サブルーチンは名前で構成される見出しで始まり、その後にコンマで区切られたパラメーターまたは引数のリストを含むかっこが続きます。サブルーチンの本体は、BEGIN – END; のペアで囲まれた一連のステートメントです。たとえば、単純なプログラムの本体（MYPROGRAM）は次のようになります。

```
EXPORT MYPROGAM()

BEGIN

PIXON(1,1);

END;
```

コメント

プログラム行が 2 つのスラッシュ（//）で始まるときは、その行の残りは無視されます。これを利用してプログラムにコメントを挿入できます。

```
EXPORT MYPROGAM()

BEGIN

PIXON(1,1);

//This line is just a comment.

END;
```

Program Catalog（プログラム カタログ）

Program Catalog（プログラム カタログ）は、プログラムを実行およびデバッグしたり、別の HP Prime にプログラムを送信したりする場所です。プログラムの名前を変更したり、プログラムを削除したり、プログラム エディターを起動したりすることもできます。プログラム エディターはプログラムを作成したり編集したりする場所です。プログラムは[ホーム表示]または他のプログラムから実行することもできます。

Program Catalog（プログラム カタログ）を開く

Shift 1 Program Y （プログラム）を押すと Program Catalog（プログラム カタログ）が開きます。

Program Catalog（プログラム カタログ）にはプログラム名のリストが表示されます。Program Catalog（プログラム カタログ）の最初の項目は、アクティブ アプリケーションと同じ名前の組み込みエントリです。このエントリは、アクティブ アプリケーションのアプリケーション プログラム（そのようなプログラムが存在する場合）です。

Program Catalog（プログラム カタログ）: ボタンおよびキー

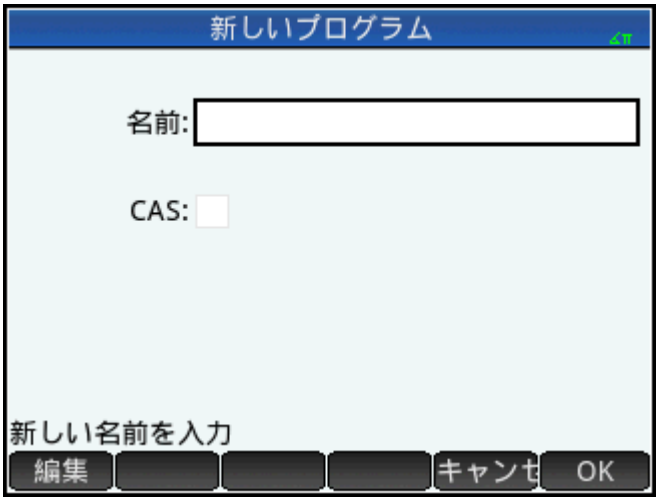
ボタンまたはキー	用途
編集	編集するために選択されたプログラムを開きます。

ボタンまたはキー	用途
新しい	新しいプログラム名の入力を要求してから、プログラム エディターを開きます。
その他	選択したプログラムの次のメニュー オプションを開きます。 **[保存]：**要求に従って指定した新しい名前で、選択されたプログラムのコピーを作成します **[名前の変更]：**選択されたプログラムの名前を変更します **[並べ替え]：**プログラムのリストを並べ替えます。（並べ替えオプションはアルファベット順および作成順です）。 **[削除]：**選択されたプログラムを削除します **[クリア]：**すべてのプログラムを削除します 初期メニューを再表示するには、 On Off または Esc Clear を押します。
送信	選択されたプログラムを別の HP Prime に送信します。
デバッグ	選択されたプログラムをデバッグします。
実行	選択されたプログラムを実行します。
Shift ▲ または Shift ▼	Program Catalog（プログラム カタログ）の先頭または末尾に移動します。
Del	選択されたプログラムを削除します。
Shift Esc Clear	カタログ内のすべての注記を削除します。

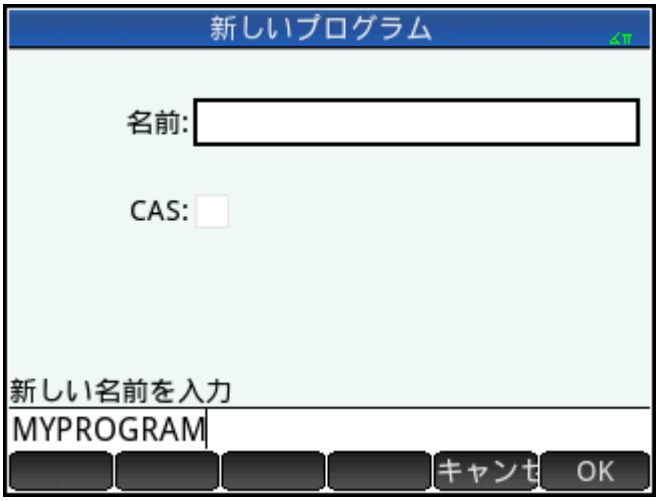
新しいプログラムの作成

次のいくつかのセクションでは、プログラム エディターおよびそのメニューを使用してみるために、3 まで数える簡単なプログラムを作成します。

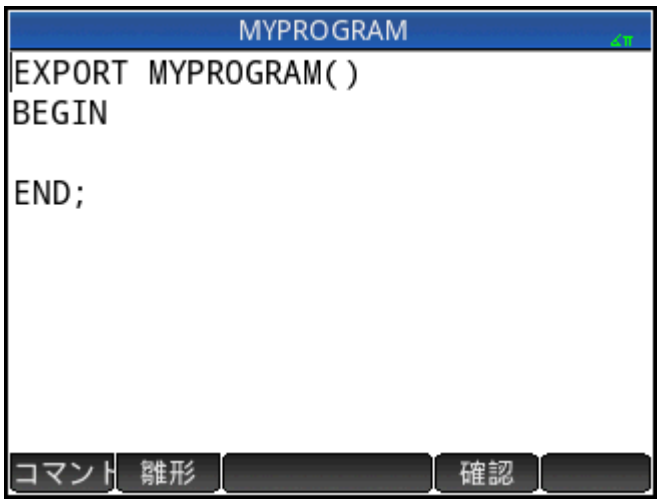
1. Program Catalog（プログラム カタログ）を開いて、新しいプログラムを起動します。 Shift 1 Program Y （プログラム） 新しい



2. プログラムの名前を入力します。 ALPHA alpha ALPHA alpha （英字モードをロックする）を押し、MYPROGRAM を入力してから、 OK を押します。



3. もう一度 OK を押します。プログラムのテンプレートが自動的に作成されます。テンプレートは、プログラムと同じ名前の関数見出し、EXPORT MYPROGRAM()、および関数のステートメントを囲む BEGIN-END のペアで構成されます。



ヒント：プログラム名には、英数字（文字と番号）およびアンダースコア文字のみを含めることができます。1 文字目は文字である必要があります。たとえば、GOOD_NAME および Spin2 は有効なプログラム名です。一方、スペースを含むおよび 2Cool!（数値で始まり、!を含む）は無効です。

プログラム エディター

HP Prime コマンドに慣れるまで、最も簡単なコマンドの入力方法は、[カタログ]メニュー（Mem カタログ）またはプログラム エディターの[コマンド]メニュー（コマンド）からコマンドを選択することです。変数、記号、数学関数、単位、または文字を入力するには、キーボードのキーを使用します。

プログラム エディター：ボタンおよびキー

プログラム エディターのボタンおよびキーについて、次の表で説明します。

ボタンまたはキー	意味
確認	エラーがないかどうか、現在のプログラムを検査します。
ページ または ALPHA alpha ▲ および ALPHA alpha ▼	プログラムが複数の画面にまたがっている場合、このボタンの右または左側をタップすることで前後の画面にすばやくジャンプできます。ボタンの左側をタップすると、前のページが表示されます。右側をタップすると、次のページが表示されます。（プログラムの最初のページを表示している場合、左側のタップは無効になります。）

ボタンまたはキー	意味
コマンド	一般的なプログラミング コマンドから選択できるメニューを開きます。コマンドは、以下のオプションでグループ分けされています。 文字列 図 行列 アプリケーション関数 整数 I/O その他 Esc Clear を押すとメイン メニューに戻ります。 このメニューのコマンドについては、579 ページの [コマンド]メニューの下にあるコマンドで説明しています。
離形	一般的なプログラミング コマンドを選択できるメニューを開きます。コマンドは、以下のオプションでグループ分けされています。 ブロック 分岐 ループ 変数 関数 Esc Clear を押すとメイン メニューに戻ります。 このメニューのコマンドについては、573 ページの [テンプレート]メニューの下にあるコマンドで説明しています。
Vars Chars A	変数の名前と値を選択するメニューを表示します。
Shift Vars Chars A （文字）	文字パレットを表示します。プログラムが開いているときにこのパレットを表示して文字を選択すると、カーソル位置でプログラムに追加されます。特定の文字を追加するには、選択して OK をタップするか Enter ≈ を押します。文字パレットを*閉じず*に文字を追加するには、選択して エコー をタップします。
Shift ▶ および Shift ◀	現在の行の末尾（または先頭）にカーソルを移動します。画面をスワイプすることもできます。
Shift ▲ および Shift ▼	プログラムの先頭（または末尾）に移動します。画面をスワイプすることもできます。

ボタンまたはキー	意味
ALPHA alpha および ALPHA alpha	カーソルを 1 画面右（または左）に移動します。画面をスワイプすることもできます。
Enter ≈	新しい行を開始します。
Del	カーソルの左にある文字を削除します。
Shift Del	カーソルの右にある文字を削除します。
Shift	プログラム全体を削除します。
Esc Clear	

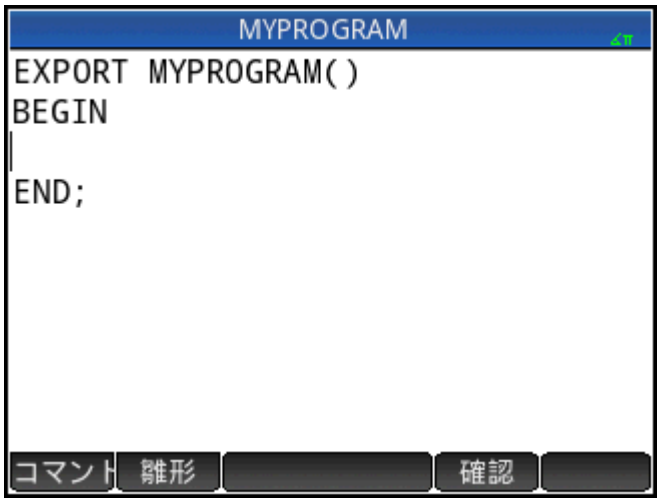
プログラム エディター内で Menu Paste を押すと、さらに 2 つのオプションが表示されます。

- **[ユーザーキーの作成]**：このオプションをタップしてからいずれかのキーを押すと、そのキーをユーザー キーとして再定義するためのテンプレートがプログラムに挿入されます。
- **[プラグマの挿入]**：このオプションをタップすると、#pragma mode 定義がプログラムに挿入されます。#pragma mode 定義は次の形式になっています。

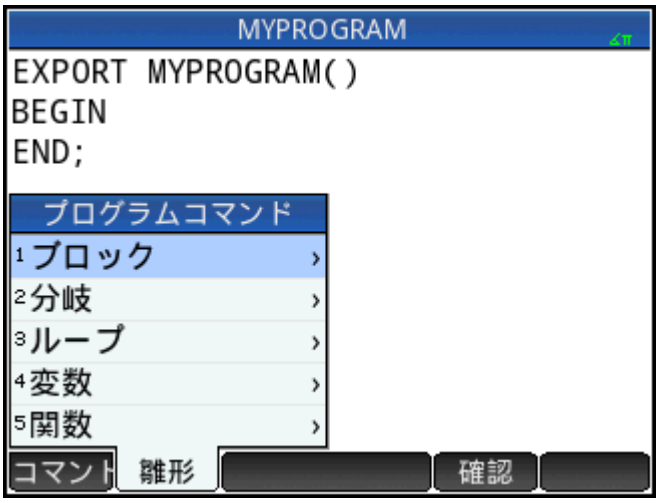
  ```
  #pragma mode( separator(), integer())
  ```

  #pragma mode 定義を使用して、桁のグループ分けに使用される一連の桁区切り記号と整数型を定義します。#pragma mode 定義によって、プログラムは強制的にこれらの設定を使用してコンパイルされます。この機能は、異なるグループ分け記号（.あるいは,）を使用する文化圏で作成されたプログラムを適応させる場合に役立ちます。

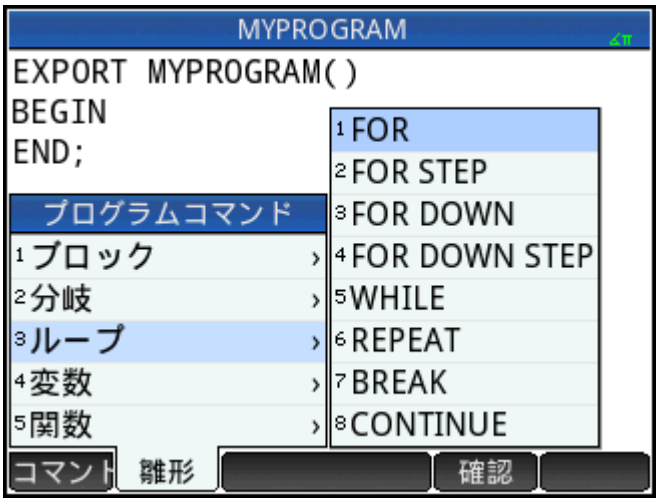
1. MYPROGRAM の例を続けます（541 ページの「HP PPL でのプログラミング」を参照）。カーソル キーを使用してコマンドの挿入位置にカーソルを置くか、目的の場所をタップします。この例では、カーソルを BEGIN と END の間に置く必要があります。

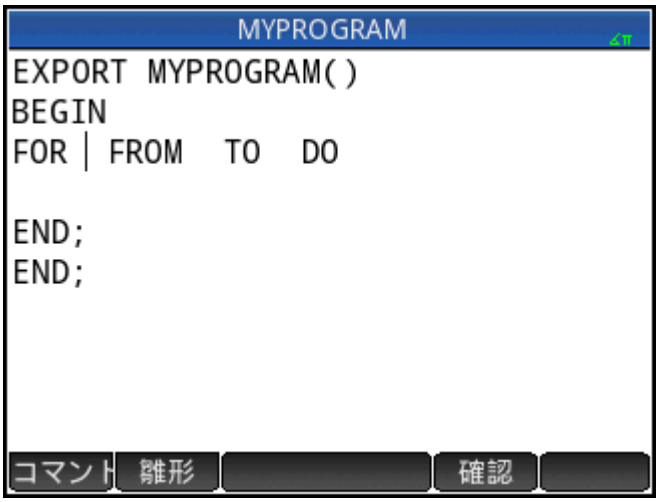


2. 雛形 をタップすると、ブロック、分岐、ループ、変数、および関数用の一般的なプログラミング コマンドのメニューが開きます。この例では、メニューから LOOP コマンドを選択します。

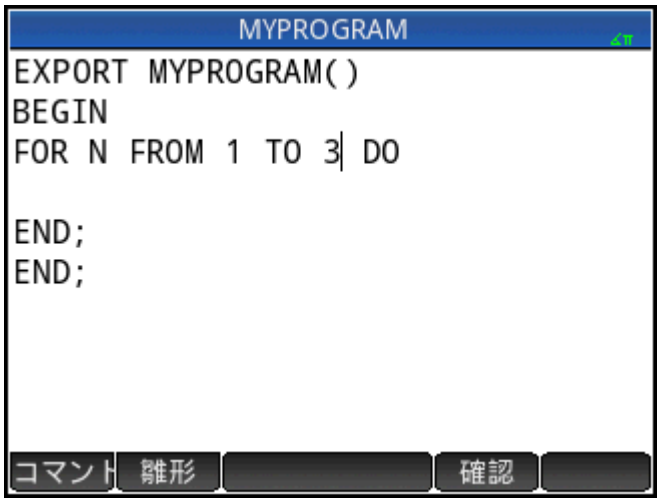


3. **[ループ]**を選択し、サブメニューから**[FOR]**を選択します。FOR_FROM_TO_DO_ テンプレートが挿入されます。不足している情報を入力する必要があります。





4. カーソル キーおよびキーボードを使用して、コマンドの不足部分を入力します。この場合、ステートメントを次のようにします。FOR N FROM 1 TO 3 DO

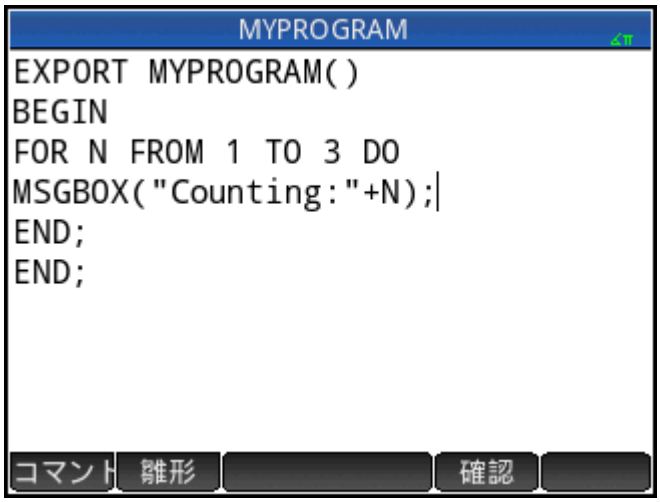


5. FOR ステートメントの下の空白行にカーソルを移動します。

6. コマンドをタップすると、一般的なプログラミング コマンドのメニューが開きます。

7. **[I/O]**を選択し、サブメニューから**[MSGBOX]**を選択します。

8. MSGBOX コマンドの引数を入力し、コマンドの末尾にセミコロンを入力します（ Shift + Ans ; ）。



9. 確認をタップして、プログラムの構文を検査します。

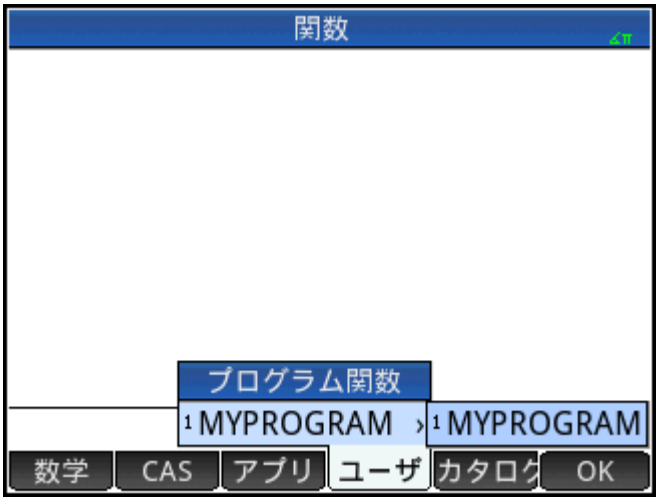
10. 終了したら、 Shift 1 Program Y を押して Program Catalog（プログラム カタログ）に戻るか、 Settings を押して[ホーム ビュー]に進みます。プログラムを実行する準備が整いました。

プログラムの実行

[ホーム ビュー]から、プログラムの名前を入力します。パラメーターを取るプログラムの場合は、プログラム名の後にかっこを置き、その中にコンマで区切られたパラメーターを含めます。プログラムを実行するには、 Enter を押します。

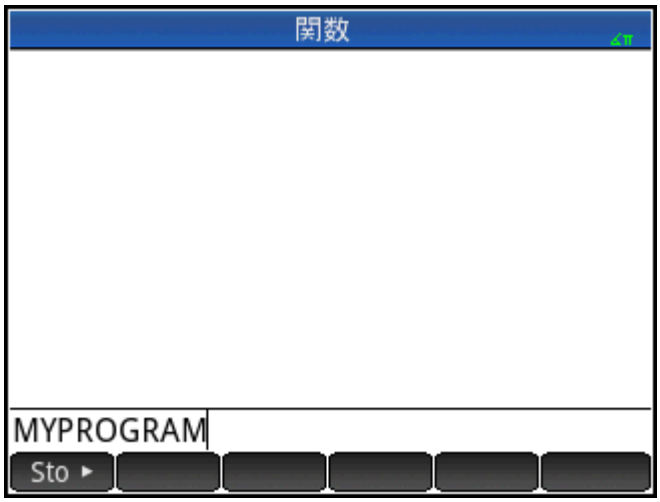
Program Catalog（プログラム カタログ）から、実行するプログラムを選択して 実行 をタップします。プログラムがカタログから実行されると、名前が `START()`（パラメーター無し）の関数が検索されます。

[ユーザー]メニュー（[ツールボックス]メニューの 1 つ）からプログラムを実行することもできます。



1. Mem を押して、 ユーザ をタップします。
2. **[MYPROGRAM >]**をタップしてメニューを展開し、**[MYPROGRAM]**を選択します。`MYPROGRAM` がエントリ行に表示されます。
3. Enter をタップするとプログラムが実行され、メッセージ ボックスが表示されます。

4. OK を 3 回タップすると、FOR ループがステップ実行されます。表示される数字が毎回 1 ずつ増分します。



プログラムの終了後、HP Prime で他の操作を再開できます。

プログラムが引数を取る場合は、 実行 を押すと、プログラム パラメーターの入力を求める画面が表示されます。

複数の関数を持つプログラム

▲ ツールボックス メニューの[ユーザー]サブメニューに複数のサブ項目があるエントリーを作成するには、単一のプログラムに複数の EXPORT コマンドを入力します。

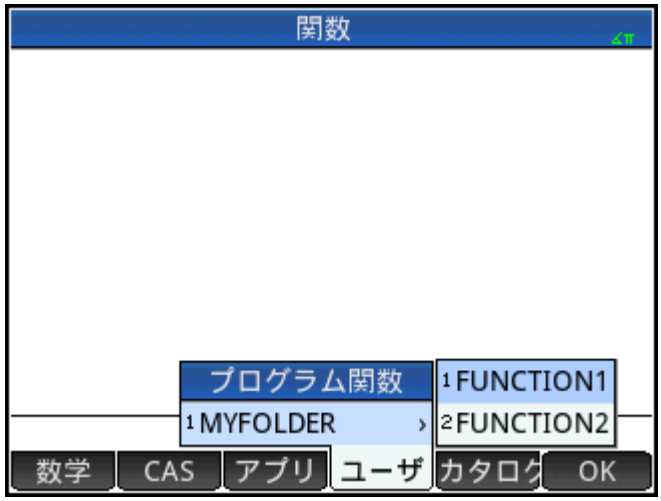
注記：通常は、プログラムとともに自動的に作成される EXPORT、BEGIN、および END の各コマンドを削除する必要があります。

以下は MYFOLDER という名前のプログラムの例です。このプログラムには以下の 2 つのユーザー定義関数が含まれています。

- FUNCTION1(X)は X+1 を返します。
- FUNCTION2(X)は X-1 を返します。

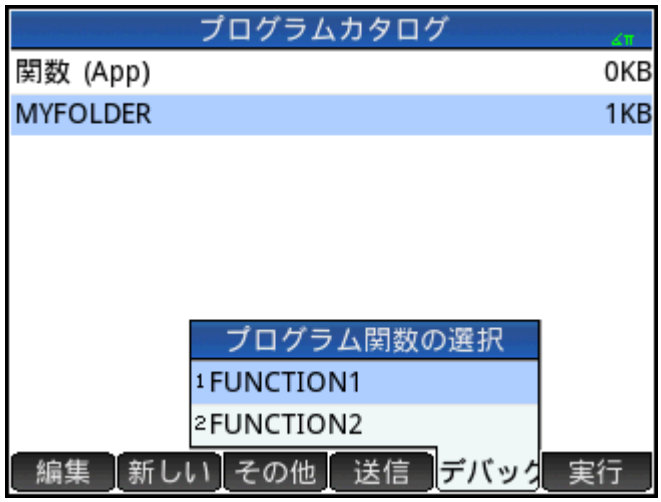
```
Program MYFOLDER

EXPORT FUNCTION1(X)

BEGIN

RETURN X+1;

END;

EXPORT FUNCTION2(X)

BEGIN

RETURN X-1;

END;
```

ここで、 Mem を押してから ユーザ をタップすると、MYFOLDER という名前のオプションが表示されます。**[MYFOLDER]**をタップすると、サブ項目 FUNCTION1 および FUNCTION2 が表示されます。



この手順を使用すると、必要な関数が含まれたカスタム フォルダーを作成して、使いやすくまとめることができます。

プログラム カタログからプログラムを選択して 実行 または デバッグ をタップすると、**[NAME1]** および**[NAME2]**のリストが表示されます。実行またはデバッグする関数を選択します。



プログラムのデバッグ

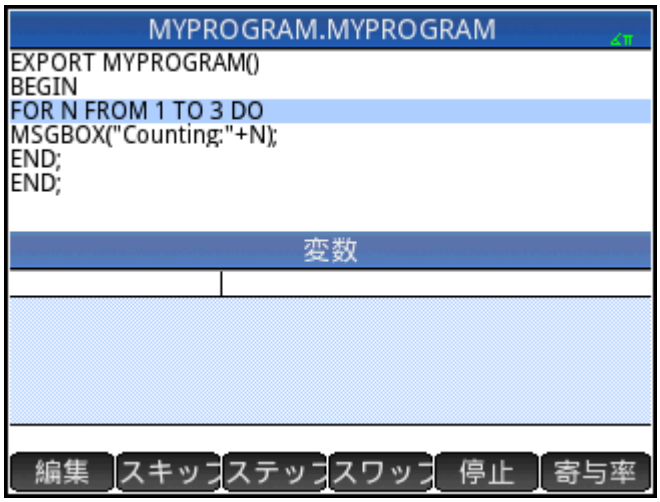
構文エラーがあるプログラムは実行できません。プログラムが期待する動作を実行しない場合、またはシステムが実行時エラーを検出した場合、プログラムを 1 ステップずつ実行して、ローカル変数の値を調べることができます。

前に作成した MYPROGRAM をデバッグしてみましょう。

1. Program Catalog（プログラム カタログ）で、MYPROGRAM を選択します。

2. デバッグをタップします。

ファイル内に複数の EXPORT 関数がある場合、デバッグする関数を選択するリストが表示されます。



プログラムをデバッグしている間、プログラムまたはプログラム内関数のタイトルがディスプレイの上部に表示されます。その下は、デバッグしているプログラムの現在行です。各変数の現在値が、画面のメイン部に表示されます。その下のメニュー ボタンをデバッガーで使用できます。

スキップ：プログラムの次の行または次のブロックにスキップします。

ステップ：現在行を実行します

変数：変数のメニューを開きます。いずれかを選択して変数リストに追加できます。プログラムをステップ実行すると変数がどのように変化するかを確認できます。

停止：デバッガーを閉じます

寄与率：デバッグなしでプログラム実行を継続します。

3. FOR ループコマンド ステップ を実行します。

   FOR ループが開始し、ディスプレイの 1 番上にプログラムの次の行（MSGBOX コマンド）が表示されます。

4. MSGBOX コマンド ステップ を実行します。

   メッセージ ボックスが表示されます。各メッセージ ボックスが表示された場合、 OK をタップするか Enter を押すことで閉じる必要があります。

   繰り返し ステップ をタップするか Enter を押すことで、プログラムがステップ単位に実行されます。

停止 をタップすると、プログラムの現在行でデバッガーが閉じます。 寄与率 をタップすると、デバッガーなしで残りのプログラムが実行されます。

プログラムの編集

プログラム エディターを使用してプログラムを編集します。プログラム エディターは Program Catalog（プログラム カタログ）からアクセスできます。

1. Program Catalog（プログラム カタログ）を開きます。

   Shift 1 Program Y

2. 編集するプログラムをタップします（または、矢印キーを使用してプログラムを選択し、Enter を押します）。

   プログラム エディターが開きます。プログラムの名前がディスプレイのタイトル バーに表示されます。プログラムの編集に使用できるボタンおよびキーの一覧は、545 ページのプログラム エディター：ボタンおよびキーに掲載されています。

プログラムまたはプログラムの一部のコピー

グローバル `Copy` および `Paste` コマンドを使用して、プログラムの一部または全体をコピーできます。手順は次のとおりです。

1. Program Catalog（プログラム カタログ）を開きます。

   Shift 1 Program Y

2. コピーするコードが含まれるプログラムをタップします。

3. Shift View Copy （コピー）を押します。

   メニュー ボタンがコピー オプションに変わります。

   開始 ：コピーまたは切り取りを開始する位置をマークします。

   終了 ：コピーまたは切り取りを終了する位置をマークします。

   すべて ：プログラム全体を選択します。

   切取 ：選択範囲を切り取ります。

   コピー ：選択範囲をコピーします。

4. コピーまたは切り取るものを選択します（すぐ上のオプションを使用）。

5. コピー または 切取 をタップします。

6. Program Catalog（プログラム カタログ）に戻り、挿入先のプログラムを開きます。

7. コピーまたは切り取ったコードを挿入する位置までカーソルを移動します。

8. Shift Menu Paste （貼り付け）を押します。クリップボードが開きます。最後にコピーまたは切り取ったものがすでにリストの先頭で選択されているため、 OK をタップするだけです。コードは、カーソル位置を開始位置としてプログラムに貼り付けられます。

プログラムの削除

プログラムを削除するには、以下の操作を行います。

1. Program Catalog（プログラム カタログ）を開きます。

   Shift 1 Program Y

2. 削除するプログラムを選択して、 Del を押します。

3. プログラムを削除する場合は OK をタップし、キャンセルする場合は キャンセ をタップします。

すべてのプログラムの削除

すべてのプログラムを一度に削除するには、以下の操作を行います。

1. Program Catalog（プログラム カタログ）を開きます。

2. （クリア）を押します。

3. すべてのプログラムを削除する場合は OK をタップし、キャンセルする場合は キャンセ をタップします。

プログラムの内容の削除

プログラムを削除せずに内容をクリアできます。プログラムには名前だけが残ります。

1. Program Catalog（プログラム カタログ）を開きます。

2. プログラムをタップして開きます。

3. （クリア）を押します。

プログラムの共有

アプリケーション、注記、行列、およびリストを送信できるように、電卓間でプログラムを送信できます。

HP Prime プログラミング言語

HP Prime プログラミング言語を使用してプログラム、関数、および変数をシステムに追加することで、HP Prime の機能を拡張できます。作成するプログラムは、スタンドアロンでもアプリケーションに関連付けてもかまいません。作成する関数および変数はローカルでもグローバルでもかまいません。グローバルに宣言されている場合、 または を押すと、[ユーザー]メニューに表示されます。以降のセクションで、変数および関数について説明してから、簡単なプログラムを作成してプログラム、関数、および変数を作成するためのさまざまなテクニックを紹介します。

変数と可視性

HP Prime プログラム内の変数は、数値、リスト、行列、グラフィック オブジェクト、および文字列を格納するために使用できます。変数の名前は、先頭が文字で始まる英数字（文字および数字）の並びでなければなりません。名前の大文字と小文字は区別されるため、変数名 `MaxTemp` と `maxTemp` は異なります。

HP Prime にはさまざまな種類の変数が組み込まれていて、可視性はグローバルです（電卓のどこにいても見ることができる）。たとえば、組み込み変数 `A` から `Z` は実数の格納に使用でき、`Z0` から `Z9` は複素数の格納に使用でき、`M0` から `M9` は行列やベクトルの格納に使用できます。これらの名前は予約されています。他のデータには使用できません。たとえば、プログラムの名前を `M1` にしたり、名前が `Z8` の変数に実数を格納したりできません。これらの予約変数以外に、各 HP アプリケーションには独自の変数が予約されています。`Root`、`Xmin`、`Numstart` がその例です。これらのアプリケーション変数のほとんどは、アプリケーションに対してローカルですが、グローバルに設計されている変数もいくつかあります。たとえば、`C1` は 2 変数統計アプリケーションで統計データを格納するために

使用されます。この変数はグローバルであるため、システムのどこからでもそのデータにアクセスできます。重ねて説明しますが、これらの名前をプログラムの名前にしたり、設計で許可されない種類のデータを格納したりすることはできません。(システムおよびアプリケーション変数の完全なリストは「変数」章で示しています。)

プログラム内の特定の関数内だけで使用するための変数を宣言できます。これは、LOCAL 宣言を使用して行います。ローカル変数を使用することで、他の電卓に影響を与えない変数を宣言して使用できます。ローカル変数は特定の種類に固定されていません。つまり、ローカル名の変数には浮動小数点数、整数、リスト、行列、および記号表現を格納できます。同じローカル変数に異なる種類を格納できますが、この方法は未熟なプログラミングであり、避けることをお勧めします。

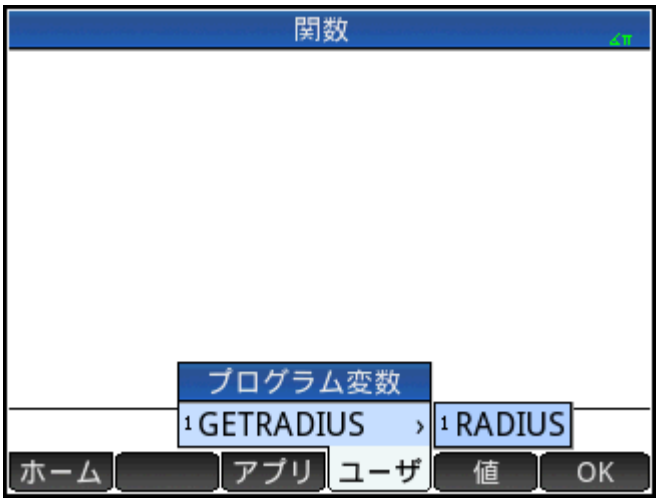
プログラムで宣言する変数には説明的な名前を付けるようにしてください。たとえば、円の半径を格納する変数の名前には、VGFTRFG よりも RADIUS の方が適しています。変数名が目的に合っていると、その変数が何のために使用されているかを思い出しやすくなります。

プログラムの実行後に変数が必要になった場合は、EXPORT コマンドを使用してプログラムからエクスポートできます。これを行うには、プログラムの最初のコマンド(すなわち、プログラム名の上の行で)は EXPORT RADIUS にしてください。すると、値が RADIUS に割り当てられた場合、名前が変数メニュー(Vars Chars A)に表示され、可視性がグローバルになります。この機能により、HP Prime のさまざまな環境の間に拡張性が高く強力な対話性を提供できます。別のプログラムが同じ名前で変数をエクスポートした場合、最後にエクスポートされたバージョンがアクティブになります。

以下のプログラムでは、RADIUS の値をユーザーに要求し、プログラムの外で使用するために変数をエクスポートします。

```
EXPORT RADIUS;

EXPORT GETRADIUS()

BEGIN

INPUT(RADIUS);

END;
```

変数 RADIUS 用の EXPORT コマンドが、RADIUS が割り当てられている関数の見出しの前に表示されます。このプログラムを実行した後に、名前が RADIUS の新しい変数が[変数]メニューの USER GETRADIUS セクションに表示されます。



変数名の修飾

HP Prime には、見た目が同じ名前のシステム変数がたくさんあります。たとえば、[関数]アプリケーションには名前が Xmin の変数がありますが、[極座標]アプリケーション、[パラメトリック]アプリケーション、[数列]アプリケーション、[解く]アプリケーションにもあります。プログラムおよび[ホーム表示]では、変数名を修飾することでこれらの変数の特定バージョンを参照できます。これは、変数が属しているアプリケーション（またはプログラム）の名前、その後にドット（.）、それから実際の変数名を入力することで行います。たとえば、修飾された変数 Function.Xmin は、関数アプリケーション内の Xmin の値を参照します。同様に、修飾された変数 Parametric.Xmin は、パラメトリックアプリケーション内の Xmin の値を参照します。同じ名前（Xmin）の変数ですが、異なる値を割り当てることができます。プログラム内のローカル変数も同様に使用できます。プログラムの名前、その後にドット、それから変数名を指定します。

関数、その引数、およびパラメーター

プログラム内に独自の関数を定義でき、パラメーターを使用してデータを関数に渡すことができます。関数は値を返しても（RETURN ステートメントを使用）、返さなくてもかまいません。[ホーム表示]からプログラムを実行すると、プログラムは最後に実行されたステートメントから返された値を返します。

さらに、プログラム内で関数を定義し、他のプログラムで使用するためにエクスポートできます。その方法は、変数を定義して他の場所で使用する場合と同じです。

このセクションでは、簡単なプログラムをいくつか作成し、HP Prime でのプログラミングの側面をいくつか説明します。各プログラムはカスタム アプリケーションの構築ブロックとして使用されます。

プログラム ROLLDIE

まず、ROLLDIE という名前のプログラムを作成します。1 つのさいころの回転をシミュレートします。1 から関数に渡される数字までのランダム整数を返します。

Program Catalog（プログラム カタログ）内に、名前が ROLLDIE の新しいプログラムを作成します。（ヘルプについては、543 ページの新しいプログラムの作成を参照してください。） 次に、プログラム エディターでコードを入力します。

```
EXPORT ROLLDIE(N)
BEGIN
RETURN 1+RANDINT(N-1);
END;
```

最初の行は関数の見出しです。RETURN ステートメントを実行すると、1 から N までのランダム整数が計算され、関数の結果として返されます。RETURN コマンドで関数の実行が終了します。つまり、RETURN ステートメントと END ステートメントの間のステートメントは無視されます。

[ホーム表示]（実際は、数字を使用できる電卓内の任意の場所）では、ROLLDIE(6) を入力でき、1 から 6（1 と 6 を含む）までのランダム整数が返されます。

プログラム ROLLMANY

ROLLDIE 内の EXPORT コマンドにより、別のプログラムは ROLLDIE 関数を使用して、任意の数の面を持つさいころの n 回転を生成できます。以下のプログラムでは、ROLLDIE 関数を使用して 2 つのさいころの *n* 回転を生成します。それぞれにローカル変数 sides で指定された数の面が割り当てられ

ます。結果はリスト L2 に格納されます。L2(1)は結合された合計が 1 になる回数を示し、L2(2)は結合された合計が 2 になる回数を示します。以下同様です。L2(1)は 0 になるはずです（2 つのさいころの数の合計は少なくとも 2 でなければならないため）。

ここでは「=」の代わりに保存記号（▶）を使用します。 を押すと、この記号が取得されます。構文は Var▶Value です。つまり、右側の値が左側の変数に格納されます。

```
EXPORT ROLLMANY(n,sides)
BEGIN
 LOCAL k,roll;
 // initialize list of frequencies
 MAKELIST(0,X,1,2*sides,1) ▶ L2;
 FOR k FROM 1 TO n DO
ROLLDIE(sides)+ROLLDIE(sides) ▶ roll;
 L2(roll)+1 ▶ L2(roll);
 END;
END;
```

関数を宣言するときに EXPORT コマンドを除外することで、その可視性を関数が定義されたプログラム内に制限できます。たとえば、以下のように ROLLDIE 関数を ROLLMANY プログラム内で定義できます。

```
ROLLDIE();
EXPORT ROLLMANY(n,sides)
BEGIN
 LOCAL k,roll;
 // initialize list of frequencies
 MAKELIST(0,X,1,2*sides,1) ▶ L2;
 FOR k FROM 1 TO n DO
ROLLDIE(sides)+ROLLDIE(sides) ▶ roll;
 L2(roll)+1 ▶ L2(roll);
 END;
END;
ROLLDIE(n)
BEGIN
RETURN 1+RANDINT(n-1);
END;
```

2 番目の ROLLMANY プログラムでは、別のプログラムからエクスポートされた ROLLDIE 関数がありません。その代わり、ROLLDIE は ROLLMANY だけを見ることができます。ROLLDIE 関数は、呼ばれる前に宣言される必要があります。上記プログラムの最初の行には ROLLDIE 関数の宣言が含まれています。ROLLDIE 関数の定義はプログラムの最後にあります。

最後に、結果のリストをグローバル リスト変数 L2 に直接格納するのではなく、ROLLMANY の呼び出しの結果として返すことができます。この方法により、ユーザーが結果を他の場所に格納したい場合、簡単に行うことができます。

```
ROLLDIE();
EXPORT ROLLMANY(n,sides)
BEGIN
 LOCAL k,roll,results;
 // initialize list of frequencies
 MAKELIST(0,X,1,2*sides,1) ▶ results;
 FOR k FROM 1 TO n DO
ROLLDIE(sides)+ROLLDIE(sides) ▶ roll;
 results(roll)+1 ▶ results(roll);
 END;
RETURN results;
END;
ROLLDIE(N)
BEGIN
RETURN 1+RANDINT(N-1);
END;
```

[ホーム表示]で、ROLLMANY(100,6) ▶ L5 を入力すると、2 つの 6 面さいころを 100 回転したシミュレーションの結果はリスト L5 に格納されます。

ユーザー キーボード：キー押下のカスタマイズ

シフト キーおよび英字キーで提供される機能を含めて、別の機能をキーボードのキーに割り当てることができます。この機能を利用して、特別なニーズに合わせてキー ボードをカスタマイズできます。たとえば、メニュー上でいくつかネストされているために辿り着くのが難しい関数（ALOG など）に

を割り当てることができます。

カスタマイズされたキーボードはユーザー キーボードと呼ばれ、ユーザー モードになったときに有効になります。

ユーザー モード

以下の 2 つのユーザー モードがあります。

- 一時的ユーザー モード：次にキーが押されたときに、そのときにだけ、そのキーに割り当てたオブジェクトに入ります。そのオブジェクトに入ると、キーボードは自動的に初期設定操作に戻ります。

  一時的ユーザー モードを有効にするには、 Shift Help User （ユーザー）を押します。**[1U]**がタイトル バーに表示されます。**[1]**により、ユーザー キーボードが 1 回のキー押下にのみ有効であることがわかります。

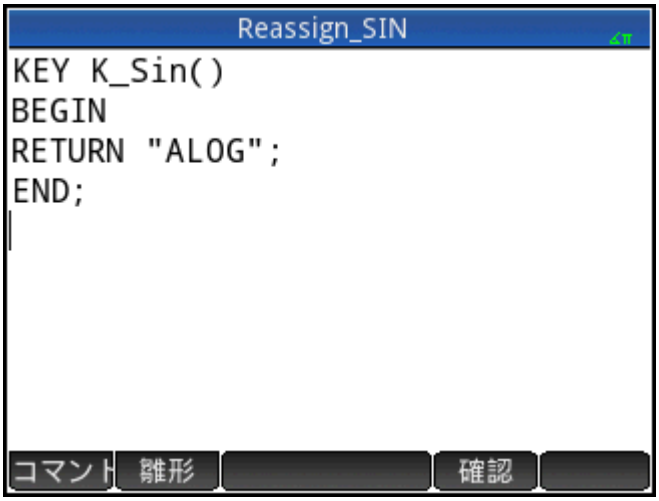
- 永続的ユーザー モード：現在から*ユーザー モードをオフにするまで*キーを押すたびに、キーに割り当てたオブジェクトに入ります。

  永続的ユーザー モードを有効にするには、 Shift Help User Shift Help User を押します。↑**U** がタイトル バーに表示されます。ユーザー キーボードが有効になり、再度 Shift Help User を押すまで有効なままになります。

ユーザー モード時に割り当てを変更していないキーを押した場合、キーの標準操作が実行されます。

キー割り当ての変更

一般的に使用される関数（ALOG など）をキーボードの独自キーに割り当てたいとします。必要なことは、以下の図の構文のような新しいプログラムを作成するだけです。



プログラムの最初の行では、再割り当てするキーを内部名を使用して指定します。（すべてのキーの名前は562 ページのキーの名前で指定されています。大文字と小文字が区別されます。）

行 3 では、再割り当てされているキーが押されたときに生成されるテキストを入力します。このテキストは引用符で囲まれている必要があります。

次にカーソル位置に `ALOG` を挿入したいときは、 Shift Help User SIN ASIN G を押すだけです。

プログラムの RETURN 行には任意の文字列を入力できます。たとえば、「Newton」を入力すると、再割り当てされたキーを押したときにこのテキストが返されます。システム関数に加えてユーザー定義関数、システム変数に加えてユーザー定義変数をプログラムから返すこともできます。

シフト キーとの組み合わせを再割り当てすることもできます。たとえば、 ALPHA alpha Shift ÷ を再割り当てして、小文字の「t」ではなく、`SLOPE(F1(X),3)` を生成するように変更できます。する

と、[ホーム表示]で ALPHA Shift ÷ が入力されてから Enter が押されると、関数アプリケーションで現在 F1(X)と定義されている関数が X = 3 での傾斜を返します。

ヒント：キーを再割り当てするプログラムを作成する簡単な方法は、プログラム エディターを開いているときに Menu Paste を押して**[ユーザー キーを作成]**を選択することです。すると、再割り当てするキー（またはキーの組み合わせ）を押すように求められます。キー（またはキーの組み合わせ）の内部名が自動的に付加された状態で、プログラム テンプレートが表示されます。

キーの名前

キーを再割り当てするプログラムの最初の行には、内部名を使用して再割り当てするキーを指定する必要があります。以下の表は各キーの内部名を示しています。キーの名前は大文字と小文字を区別することに注意してください。

キーの内部名とキーの状態				
キー	名前	Shift + キー	ALPHA + キー	ALPHA Shift + キー
Apps Info	K_Apps	KS_Apps	KA_Apps	KSA_Apps
Symb Setup	K_Symb	KS_Symb	KA_Symb	KSA_Symb
▲	K_Up	KS_Up	KA_Up	KSA_Up
Help User	K_Help	—	KA_Help	KSA_Help
Esc Clear	K_Esc	KS_Esc	KA_Esc	KSA_Esc
Settings	K_Home	KS_Home	KA_Home	KSA_Home
Plot Setup	K_Plot	KS_Plot	KA_Plot	KSA_Plot
◀	K_Left	KS_Left	KA_Left	KSA_Left
▶	K_Right	KS_Right	KA_Right	KSA_Right
View Copy	K_View	KS_View	KA_View	KSA_View
CAS Settings	K_Cas	KS_Cas	KA_Cas	KSA_Cas
Num Setup	K_Num	KS_Num	KA_Num	KSA_Num

キーの内部名とキーの状態				
キー	名前	Shift + キー	ALPHA + キー	ALPHA Shift + キー
	K_Down	KS_Down	KA_Down	KSA_Down
	K_Menu	KS_Menu	KA_Menu	KSA_Menu
	K_Vars_	KS_Vars_	KA_Vars_	KSA_Vars_
	K_Math	KS_Math	KA_Math	KSA_Math
	K_Templ	KS_Templ	KA_Templ	KSA_Templ
	K_Xttn	KS_Xttn	KA_Xttn	KSA_Xttn
	K_Abc	KS_Abc	KA_Abc	KSA_Abc
	K_Bksp	KS_Bksp	KA_Bksp	KSA_Bksp
	K_Power	KS_Power	KA_Power	KSA_Power
	K_Sin	KS_Sin	KA_Sin	KSA_Sin
	K_Cos	KS_Cos	KA_Cos	KSA_Cos
	K_Tan	KS_Tan	KA_Tan	KSA_Tan
	K_Ln	KS_Ln	KA_Ln	KSA_Ln
	K_Log	KS_Log	KA_Log	KSA_Log
	K_Sq	KS_Sq	KA_Sq	KSA_Sq
	K_Neg	KS_Neg	KA_Neg	KSA_Neg
	K_Paren	KS_Paren	KA_Paren	KSA_Paren
	K_Comma	KS_Comma	KA_Comma	KSA_Comma

キーの内部名とキーの状態				
キー	名前	Shift + キー	ALPHA alpha + キー	ALPHA alpha Shift + キー
Enter ≈	K_Ente	KS_Enter	KA_Enter	KSA_Enter
EEX Sto▸ P	K_Eex	KS_Eex	KA_Eex	KSA_Eex
7 List Q	K_7	KS_7	KA_7	KSA_7
8 { } R	K_8	KS_8	KA_8	KSA_8
9 !,∞,→ S	K_9	KS_9	KA_9	KSA_9
÷ x^{-1} T	K_Div	KS_Div	KA_Div	KSA_Div
ALPHA alpha	K_Alpha	KS_Alpha	KA_Alpha	KSA_Alpha
4 Matrix U	K_4	KS_4	KA_4	KSA_4
5 [ ] V	K_5	KS_5	KA_5	KSA_5
6 ≤,≥,≠ W	K_6	KS_6	KA_6	KSA_6
× ∡ X	K_Mul	KS_Mul	KA_Mul	KSA_Mul
Shift	—	—	—	—
1 Program Y	K_1	KS_1	KA_1	KSA_1
2 i Z	K_2	KS_2	KA_2	KSA_2
3 π #	K_3	KS_3	KA_3	KSA_3
− Base :	K_Minus	KS_Minus	KA_Minus	KSA_Minus
On Off	K_On	—	KA_On	KSA_On

キーの内部名とキーの状態				
キー	名前	Shift + キー	ALPHA + キー	ALPHA Shift + キー
0 Notes	K_0	KS_0	KA_0	KSA_0
. =	K_Dot	KS_Dot	KA_Dot	KSA_Dot
_	K_Space	KS_Space	KA_Space	KSA_Space
+ Ans ;	K_Plus	KS_Plus	KA_Plus	KSA_Plus

アプリケーション プログラム

アプリケーションとは、表示、プログラム、注記、および関連データを 1 つにまとめたもののことです。アプリケーション プログラムを作成することで、アプリケーションの表示、およびユーザーがこれらの表示とどのようにやり取りするかを再定義できます。(a) 特別な名前の専用プログラム関数を使用する方法と、(b) **[表示]**メニューで表示を再定義する方法があります。

専用プログラム関数の使用

以下の表で示すように、9 個の専用プログラム関数名があります。これらの関数は、表で示されている対応キーが押されたときに呼ばれます。これらの関数は、アプリケーションを制御するプログラムに記述され、そのアプリケーションのコンテキストで使用されるように設計されています。

プログラム	名前	対応するキー操作
Symb	記号表示	Symb Setup
SymbSetup	記号のセットアップ	Shift Symb Setup
Plot	プロット表示	Plot Setup
PlotSetup	プロットのセットアップ	Shift Plot Setup
Num	数値表示	Num Setup
NumSetup	Num Setup（数値のセットアップ）	Shift Num Setup
Info	Info View（情報表示）	Shift Apps Info

プログラム	名前	対応するキー操作
START	アプリケーションを起動	開始
RESET	アプリケーションをリセットまたは初期化	リセット

[表示]メニューの再定義

アプリケーションは**[表示]**メニューを使用して、上記の表で示されている 7 個の標準表示以外の表示を定義できます。初期設定では、すべての HP アプリケーションにこのメニューに含まれる追加表示セットが用意されています。`VIEW` コマンドを使用することで、アプリケーション用に作成したプログラムを実行するようにこれらの表示を再定義できます。`VIEW` コマンドの構文は以下のとおりです。

```
VIEW "text", function()
```

関数の宣言の前に `VIEW "text", function()` を追加することで、アプリケーションの表示リストが上書きされます。たとえば、アプリケーション プログラムに 3 つの表示「SetSides」、「RollDice」、および「PlotResults」が定義されている場合は、 View Copy を押すと、アプリケーションの初期設定表示リストの代わりに SetSides、RollDice、および PlotResults が表示されます。

アプリケーションのカスタマイズ

アプリケーションがアクティブのときは、それに関連付けられたプログラムが Program Catalog（プログラム カタログ）の最初の項目として表示されます。カスタム アプリケーションを作成するための関数は、このプログラム内に置きます。アプリケーションのカスタマイズに役立つ手順を以下に示します。

1. カスタマイズする HP アプリケーションを決めます。カスタマイズしたアプリケーションは、HP アプリケーションのすべてのプロパティを継承します。
2. アプリケーション ライブラリ（ Apps Info ）に移動し、HP アプリケーションを選択し、 保存 をタップして一意名でアプリケーションを保存します。
3. 必要に応じて新しいアプリケーションをカスタマイズします（たとえば、軸または角度の単位設定を構成することで）。
4. Program Catalog（プログラム カタログ）を開き、新しいアプリケーション プログラムを選択して、 編集 をタップします。
5. カスタマイズされたアプリケーションを操作する関数を作成します。関数を作成するときは、すでに説明したアプリケーション命名規則を使用してください。
6. `VIEW` コマンドをプログラムに記述して、アプリケーションの[表示]メニューを修正します。
7. アプリケーションで新しいグローバル変数を作成するかどうかを決めます。作成する場合は、アプリケーション プログラム内の `Start()` 関数から呼び出される別個のユーザー プログラムから、それらを `EXPORT`（エクスポート）することをお勧めします。こうすることで、それらの値が失われなくなります。
8. アプリケーションを検査して、関連付けられたプログラムをデバッグします。

プログラムから複数のアプリケーションをリンクできます。たとえば、関数アプリケーションに関連付けられたプログラムから、1 変数統計アプリケーションを起動するコマンドを実行し、1 変数統計アプリケーションに関連付けられたプログラムから関数アプリケーションに戻る（または、他のアプリケーションを起動する）ことができます。

例

以下の例で、カスタム アプリケーションを作成する手順を示します。このアプリケーションは、組み込みの 1 変数統計アプリケーションに基づいています。1 組のさいころの回転をシミュレートします。それぞれのさいころの面数は、ユーザーが指定します。結果は一覧になります。表またはグラフで見ることができます。

1. アプリケーション ライブラリで、1 変数統計アプリケーションを選択します。ただし、開かないでください。

   Apps Info **[1 変数統計]**を選択します。

2. 保存 をタップします。
3. 新しいアプリケーションの名前（`DiceSimulation` など）を入力します。
4. OK を 2 回タップします。新しいアプリケーションがアプリケーション ライブラリに表示されます。
5. Program Catalog（プログラム カタログ）を開きます。

   Shift 1 Program Y

6. プログラムをタップして開きます。

   カスタマイズされたアプリケーションごとに、1 つのプログラムが関連付けられます。最初はこのプログラムは空です。関数をこのプログラムに入力することで、アプリケーションをカスタマイズしてください。

ここで、ユーザーがアプリケーションとどのように対話するかを決めます。この例では、以下のことができるようにします。

- アプリケーションを起動して初期化し、短い注記を表示します
- 各さいころの面の数を指定します
- さいころの回転数を指定します
- シミュレーションの結果をグラフで表示します
- シミュレーションの結果を数値で表示します

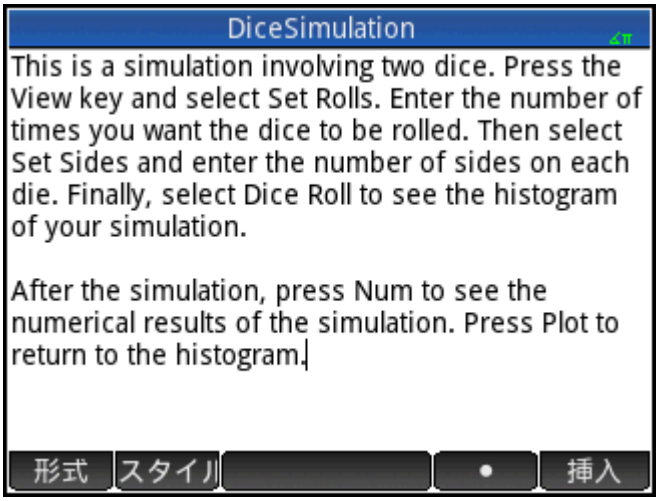
この点を前提として、以下の表示を作成します：

START、ROLL DICE、SET SIDES、および SET ROLLS。

START オプションでは、アプリケーションを初期化して、ユーザーに指示を伝える注記を表示します。ユーザーは[数値表示]および[プロット表示]でもアプリケーションと対話します。

これらの表示は Num Setup および Plot Setup を押すことで有効化されますが、実際にはアプリケーションプログラム内の関数 `Plot()` がいくつかの構成を行った後に後者の表示を起動します。

以下のプログラムを入力する前に、Shift Apps Info を押して Info（情報）エディターを開き、図に示すテキストを入力します。この注記はアプリケーションに添付され、ユーザーが[表示]メニューから[開始]オプションを選択したとき（または Shift Apps Info を押したとき）に表示されます。



この章ですでに説明したさいころの面数を取得するプログラムはここで拡張され、2 つのさいころが出す可能性のある合計値はデータセット D1 に格納されます。DiceSimulation アプリケーションのプログラムに以下のサブルーチンを入力します。

DiceSimulation プログラム

```
DICESIMVARS();
ROLLDIE();
 EXPORT SIDES,ROLLS;
EXPORT DiceSimulation()
BEGIN
END;
VIEW "Start",START()
BEGIN
 D1:={};
 D2:={};
 SetSample(H1,D1);
 SetFreq(H1,D2);
 H1Type:=1;
 STARTVIEW(6,1);
END;
VIEW "Roll Dice",ROLLMANY()
BEGIN
 LOCAL k,roll;
 D1:= MAKELIST(X+1,X,1,2*SIDES-1,1);
 D2:= MAKELIST(0,X,1,2*SIDES-1,1);
```

```
 FOR k FROM 1 TO ROLLS DO
 roll:=ROLLDIE(SIDES)+ROLLDIE
 (SIDES);
 D2(roll-1):= D2(roll-1)+1;
 END;
 Xmin:= -0.1;
 Xmax:= MAX(D1)+1;
 Ymin:= -0.1;
 Ymax:= MAX(D2)+1;
 STARTVIEW(1,1);
END;
VIEW "Set Sides",SETSIDES()
BEGIN
 REPEAT
 INPUT(SIDES,"Die Sides","N=","Enter# of sides",2);
 SIDES:= FLOOR(SIDES);
 IF SIDES<2 THEN MSGBOX("# of sides must be >= 4");
 END;
 UNTIL SIDES >=4;
 STARTVIEW(7,1);
END;

VIEW "Set Rolls",SETROLLS()
BEGIN
 REPEAT
 INPUT(ROLLS,"Num of rolls","N=","Enter# of rolls",25);
 ROLLS:= FLOOR(ROLLS);
 IF ROLLS<1 THEN MSGBOX("You must enter a num >=1");
 END;
 UNTIL ROLLS>=1;
 STARTVIEW(7,1);
END;
```

```
PLOT()
BEGIN
 Xmin:=-0.1;
 Xmax:= MAX(D1)+1;
 Ymin:= -0.1;
 Ymax:= MAX(D2)+1;
 STARTVIEW(1,1);
END;
Symb()
BEGIN
 SetSample(H1,D1);
 SetFreq(H1,D2);
 H1Type:=1;
 STARTVIEW(0,1);
END;
```

ROLLMANY()ルーチンは、この章ですでに提示したプログラムを改造したものです。カスタム[表示]メニューからの選択で呼び出されたプログラムにパラメーターを渡すことができないため、エクスポートされた変数の SIDES および ROLLS が前のバージョンで使用されていたパラメーターの代わりに使用されています。

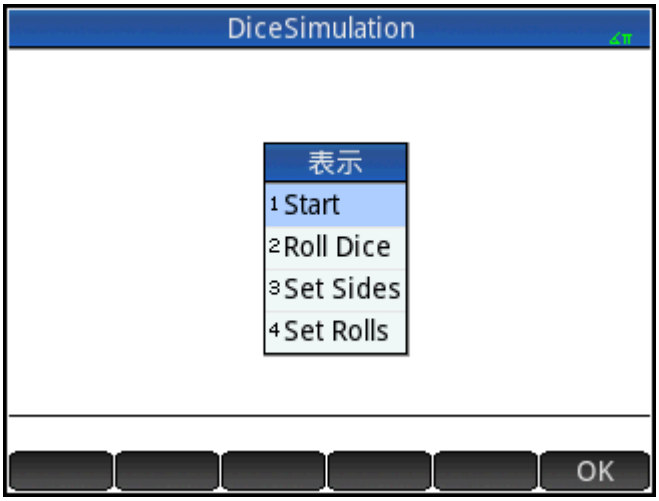
上記のプログラムは、他の 2 つのユーザー プログラム ROLLDIE()および DICESIMVARS()を呼び出します。ROLLDIE()はこの章ですでに示しました。次は DICESIMVARS です。この名前でプログラムを作成し、以下のコードを入力します。

プログラム DICESIMVARS

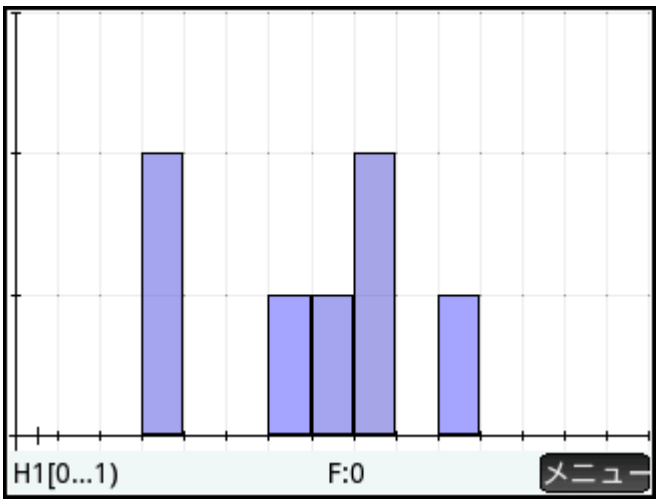
```
EXPORT ROLLS,SIDES;
EXPORT DICESIMVARS()
BEGIN
10 ▶ ROLLS;
6 ▶ SIDES;
END;
```

1. Apps Info を押して、DiceSimulation を開きます。アプリケーションの機能を説明する注記が表示されます。

2. View Copy を押して、カスタム アプリケーション メニューを表示します。ここで、アプリケーション（Start（開始））をリセットしたり、さいころの面数と回転数を設定したり、シミュレーションを行ったりできます。



3. **[回転数を設定]**を選択して、100 を入力します。
4. **[回転数を設定]**を選択して、6 を入力します。
5. **[さいころを回転]**を選択します。図で示すようなヒストグラムが表示されます。



6. Num Setup を押すとデータが表示され、Plot Setup を押すとヒストグラムに戻ります。
7. 別のシミュレーションを実行するには、View Copy を押して**[さいころを回転]**を選択します。

プログラム コマンド

このセクションでは、各プログラム コマンドについて説明します。まず、離形 メニューの下にあるコマンドについて説明します。コマンド メニューの下にあるコマンドは、579 ページの[コマンド]メニューの下にあるコマンドで説明します。

[テンプレート]メニューの下にあるコマンド

ブロック

ブロック コマンドは、サブルーチンまたは関数の始まりと終わりを決めます。サブルーチンまたは関数から結果を回収する Return コマンドもあります。

BEGIN END

構文：`BEGIN command1; command2;…; commandN; END;`

コマンド、または一緒に実行するコマンド セットを定義します。以下は、簡単なプログラムです。

```
EXPORT SQM1(X)
BEGIN
RETURN X^2-1;
END;
```

このブロックは簡単な RETURN コマンドです。

ホーム表示で `SQM1(8)` を入力すると、返される結果は 63 になります。

RETURN

構文：`RETURN` *expression*;

expression の現在の値を返します。

KILL

構文：`KILL`;

現在のプログラムのステップ単位実行を停止します（デバッグで使用）。

分岐

以下の記述では、複数形の単語 *commands* は、1 つのコマンドまたはコマンド セットの両方を表します。

IF THEN

構文：`IF` *test* `THEN` *commands* `END`;

test を評価します。*test* が真である場合（0 でない）は、*commands* を実行します。そうでない場合は、何も実行しません。

IF THEN ELSE

構文：`IF` *test* `THEN` *commands 1* `ELSE` *commands 2* `END`;

test を評価します。*test* が真である場合（0 でない）、*commands 1* を実行し、そうでない場合は *commands 2* を実行します。

test がリストを返す場合、*commands 1* および *commands 2* は単一のオブジェクトを返すか、両方とも *test* が返すリストと同じサイズのリストを返す必要があります。

commands 1 および *commands 2* の両方がリストを返す場合、各リストは同じサイズであり、テスト リストの要素の *test* の結果に応じて、各要素が *commands 1* または *commands 2* のどちらかから選択されます。

CASE

構文：

```
CASE
 IF test THEN commands END;
 IF test THEN commands END;
...
[ DEFAULT commands]
END;
```

test1 を評価します。真である場合、*commands1* を実行して、CASE を終了します。そうでない場合は、*test1* を評価します。真である場合、*commands2* を実行して、CASE を終了します。真が見つかるまで、テストの評価を続行します。真になるテストが見つからない場合、初期設定のコマンドが指定されていれば、これを実行します。CASE コマンドの分岐数は 127 に制限されています。

例：

```
CASE
IF A<0 THEN RETURN "negative"; END;
IF 0≤A≤1 THEN RETURN "small"; END;
DEFAULT RETURN "large";
END;
```

IFERR

```
IFERR commands1 THEN commands2 END;
```

commands1 のシーケンスを実行します。*commands1* の実行中にエラーが発生した場合は、*commands2* のシーケンスを実行します。

注記：エラー番号は変数 *Ans* に格納されています。この変数は、IFERR コマンドの THEN 句の *commands2* 構文で使用できます。

IFERR ELSE

```
IFERR commands1 THEN commands2 ELSE commands3 END;
```

commands1 のシーケンスを実行します。*commands1* の実行中にエラーが発生した場合は、*commands2* のシーケンスを実行します。そうでない場合は、*commands3* のシーケンスを実行します。

ループ

FOR

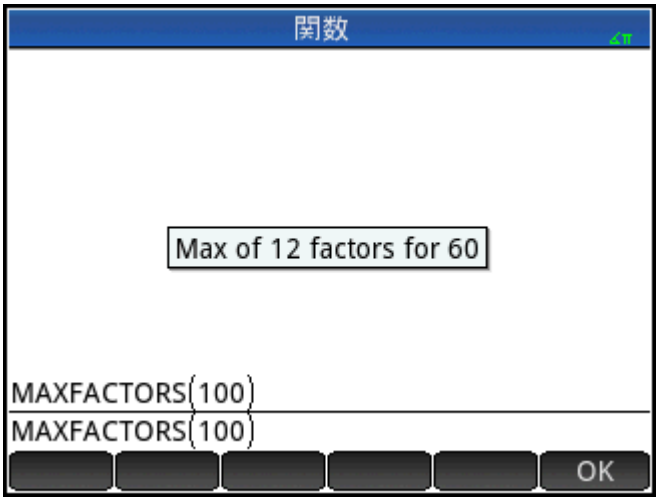
構文：`FOR` *var* `FROM` *start* `TO` *finish* `DO` commands `END;`

変数 *var* を *start* に設定し、この変数が *finish* 以下である間は、*commands* を実行してから、1（*increment*）を *var* に加えます。

例 1：このプログラムでは、2 から N の整数のうち、因数の数が最も多いものを求めます。

```
EXPORT MAXFACTORS(N)
BEGIN
LOCAL cur,max,k,result;
1 ▶ max;1 ▶ result;
FOR k FROM 2 TO N DO
 SIZE(CAS.idivis(k)) ▶ cur;
 IF cur(1) > max THEN
 cur(1) ▶ max;
 k ▶ result;
 END;
END;
MSGBOX("Max of "+ max +" factors for "+result);
END;
```

ホームで、`MAXFACTORS(100)`を入力します。

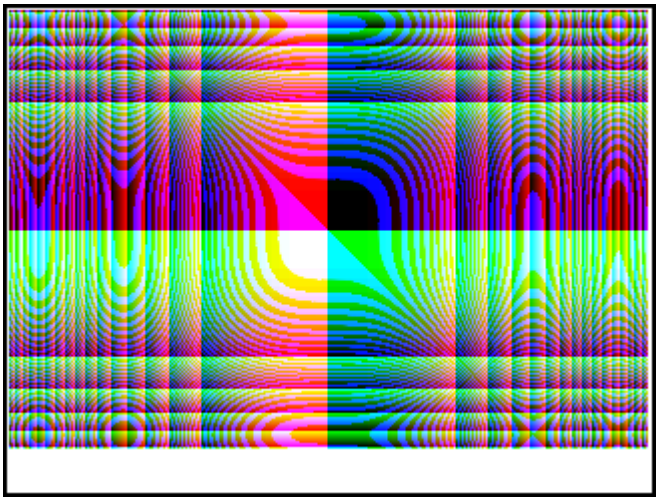


FOR STEP

構文：`FOR` *var* `FROM` *start* `TO` *finish* [*STEP increment*] `DO` commands END;

変数 *var* を *start* に設定し、この変数が *finish* 以下である間は、コマンドのシーケンスを実行してから、*increment* を *var* に加えます。

例 2：このプログラムでは魅力的なパターンを画面に描きます。



```
EXPORT
DRAWPATTERN()
BEGIN
LOCAL
xincr,yincr,color;
STARTAPP("Function");
RECT();
xincr := (Xmax - Xmin)/318;
yincr := (Ymax - Ymin)/218;
FOR X FROM Xmin TO Xmax STEP xincr DO
FOR Y FROM Ymin TO Ymax STEP yincr DO
color := RGB(X^3 MOD 255,Y^3 MOD 255, TAN(0.1*(X^3+Y^3)) MOD 255);
PIXON(X,Y,color);
END;
END;
WAIT;
END;
```

FOR DOWN

構文：FOR *var* FROM *start* DOWNTO *finish* DO *commands END*;

変数 *var* を *start* に設定し、この変数が *finish* 以上である間は、コマンドのシーケンスを実行してから、1（decrement）を *var* から引きます。

FOR STEP DOWN

構文：FOR *var* FROM *start* DOWNTO *finish* [*STEP increment*] DO *commands END*;

変数 *var* を *start* に設定し、この変数が *finish* 以上である間は、コマンドのシーケンスを実行してから、*increment* を *var* から引きます。

WHILE

構文：`WHILE` *test* `DO` *commands* `END;`

`test` を評価します。結果が真である場合（0 でない）、*commands* を実行し、繰り返します。

例：完全数とは、その数自身を除くすべての約数の合計に等しい数のことです。たとえば、6 は 6 = 1+2+3 であるため完全数です。以下の例では、引数が完全数のときに真を返します。

```
EXPORT ISPERFECT(n)
BEGIN
 LOCAL d, sum;
 2 ▶ d;
 1 ▶ sum;
WHILE sum <= n AND d < n DO
 IF irem(n,d)==0 THEN sum+d ▶ sum;
 END;
 d+1 ▶ d;
 END;
 RETURN sum==n;
END;
```

次のプログラムでは、1000 までのすべての完全数を表示します。

```
EXPORT PERFECTNUMS()
BEGIN
LOCAL k;
FOR k FROM 2 TO 1000 DO
 IF ISPERFECT(k) THEN
 MSGBOX(k+" is perfect, press OK");
 END;
END;
END;
```

REPEAT

構文：`REPEAT` commands `UNTIL` test;

テストが真（0 でない）になるまでコマンドのシーケンスを繰り返します。

次の例は、SIDES の正値を要求するものです。この章ですでに提示したプログラムを修正しています。

```
EXPORT SIDES;
EXPORT GETSIDES()
BEGIN
 REPEAT
 INPUT(SIDES,"Die Sides","N = ","Enter num sides",2);
 UNTIL SIDES>0;
END;
```

BREAK

構文：BREAK(n)

n ループ レベルで抜け出すことでループから出ます。実行は、ループ後の最初のステートメントで再開します。引数が無い場合、単一のループから出ます。

CONTINUE

構文：CONTINUE

ループの次の繰り返しの先頭に実行を移します。

変数

これらのコマンドでユーザー定義変数の可視性を制御できます。

LOCAL

構文：LOCAL var1,var2,…varn;

var1 や var2 などの変数をそれらが存在するプログラムにローカルにします。

EXPORT

構文：EXPORT var1, [var2, …, varn];

または

EXPORT var1:=val1, [var2:=val2, … varn:=valn];

変数 *var1, var2* などをエクスポートします。それらはグローバルに利用でき、 Vars を押して ユーザ を選択したときに**[ユーザー]**メニューに表示されます。

例：

```
EXPORT ratio:=0.15;
```

関数

これらのコマンドでユーザー定義関数の可視性を制御できます。

EXPORT

構文：EXPORT FunctionName(Parameters)

または

```
EXPORT FunctionName(Parameters)
BEGIN
FunctionDefinition
END;
```

プログラムで、グローバルにエクスポートする関数または変数を宣言します。エクスポートされる関数は、ツールボックスの[ユーザー]メニューに表示されます。エクスポートされる変数は、[変数]の[CAS]、[アプリ]、または[ユーザー]メニューに表示されます。

例：

```
EXPORT X2M1(X);
Export X2M1(X)
BEGIN
RETURN X^2-1;
END;
```

VIEW

構文：`VIEW “text”, functionname();`

現在のアプリケーションの**[表示]**メニューを置き換えて、「テキスト」付きのエントリを追加します。「テキスト」が選択されて、ユーザーが OK または Enter ≈ を押すと、`functionname()`が呼び出されます。

KEY

ユーザー キーボード作成時のキーボード名へのプレフィックスです。560 ページのユーザー キーボード：キー押下のカスタマイズを参照してください。

[コマンド]メニューの下にあるコマンド

文字列

文字列は、二重引用符（""）で囲まれた連続する複数の文字です。二重引用符を文字列内に置くには、2 つの連続する二重引用符を使用します。\文字はエスケープ シーケンスを開始し、直後に続く文字は特別に解釈されます。\n は新しい行を挿入し、2 つのバックスラッシュは 1 つのバックスラッシュを挿入します。文字列内に新しい行を置くには、 Enter ≈ を押します。その場所でテキストが折り返されます。

ASC

構文：`ASC(string)`

文字列の ASCII コードを含むリストを返します。

例：`ASC("AB")`は[65,66]を返します。

LOWER

文字列内の大文字を小文字に変換します。

例：

LOWER("ABC")は"abc"を返します

LOWER("ABΓ")は"αβγ"を返します

UPPER

文字列内の小文字を大文字に変換します。

例：

UPPER("abc")は"ABC"を返します

UPPER("αβγ")は"ABΓ"を返します

CHAR

構文：CHAR(vector)またはCHAR(integer)

vector内の文字コード、またはintegerの単一コードに対応する文字列を返します。

例：CHAR(65)は「A」を返します。

CHAR([82,77,72])は「RMH」を返します。

DIM

構文：DIM(string)

文字列内の文字数を返します。

例：DIM("12345")は5を返し、DIM("""")およびDIM("\n")は1を返します。(2つの二重引用符とエスケープ シーケンスが使用されている点に注目してください。)

STRING

構文：STRING(Expression, [Mode], [Precision], [Separator]または{Separator, ["[DecimalPoint[Exponent[NegativeSign]]]"], [DotZero]}], [SizeLimit]または{SizeLimit, [FontSize], [Bold], [Italic], [Monospaced]}]

Expressionを評価し、その結果を文字列として返します。

追加パラメーターは数字の表示方法を指定します。

Modeを指定する場合、以下のように指定する必要があります：

0：現在の設定を使用します

1：標準

2：固定

3：関数

4：工学

5：浮動

6：丸め

真分数モードを指定するには、この値に7を追加し、帯分数モードを指定するには、14を追加します。

Precision は、-1（現在の設定）または 0 から 12 のいずれかです。

Separator は、数字と区切り文字のセットを含む文字列です。最後の数字が小数点の直前のものと見なされます。Separator は数字でもかまいません。-1 は初期設定を使用することを意味し、0 から 10 はホーム設定で使用できる 11 個の組み込みの区切り文字のうちのいずれかを使用することを指定します。

“[DecimalPoint[Exponent[NegativeSign]]]”は 0 から 3 文字の文字列です。最初の 1 文字は小数点に使用され、2 番目の文字は指数に、最後の文字は負の記号に使用されます。

DotZero がゼロでない場合、数字は 0.1 ではなく.1 の形式で表示されます。

SizeLimit が指定されると、コマンドは与えられたピクセル数に適合する数値表現を生成しようとします。フォント サイズ（10 から 22）およびプロパティ（太字、斜体、および等幅。ブール値で指定し、0 が false）を指定することもできます。結果が適合している保証はありませんが、コマンドは適合させようとします。

例：

文字列	結果
string(F1)、when F1(X) = COS(X)の場合	"COS(X)"
STRING(2/3)	0.666666666667
string(L1)、L1 = {1,2,3}の場合	"{1,2,3}"
string(M1)、M1 =の場合 $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	"[[1,2,3],[4,5,6]]"

INSTRING

構文：INSTRING (str1,str2)

str1 内での *str2* の初回出現時のインデックスを返します。*str2* が *str1* 内に存在しない場合、0 を返します。文字列内の最初の文字は位置 1 です。

例：

INSTRING ("vanilla","van") は 1 を返します。

INSTRING ("banana","na")は 3 を返します。

INSTRING ("ab","abc")は 0 を返します。

LEFT

構文：LEFT (str,n)

文字列 *str* の最初の *n* 文字を返します。*n* ≥ DIM(str)または *n* < 0 の場合、str を返します。*n* == 0 の場合、その文字列を返します。

例：LEFT("MOMOGUMBO",3)は「MOM」を返します。

RIGHT

構文：RIGHT(str,n)

文字列 *str* の最後の *n* 文字を返します。*n* <= 0 の場合、空の文字列を返します。*n* > DIM(str)の場合、str を返します。

例：RIGHT("MOMOGUMBO",5)は、「GUMBO」を返します。

MID

構文：MID(str,pos, [n])

文字列 *str* からインデックス位置を始点として *n* 文字を抽出します。*n* はオプションです。指定されない場合は、残りのすべての文字列を抽出します。

例：MID("MOMOGUMBO",3,5)は「MOGUM」を返します。MID("PUDGE",4)は「GE」を返します。

ROTATE

構文：ROTATE(str,n)

文字列 *str* の文字の順列を返します。0 <=*n* < DIM(str)の場合、*n* だけ左方向にずらします。 – DIM(str) < *n* <= – 1 の場合、*n* だけ右方向にずらします。*n* > DIM(str)または *n* < – DIM(str)の場合、*str* を返します。

例：

ROTATE("12345",2)は「34512」を返します。

ROTATE("12345",-1)は「51234」を返します。

ROTATE("12345",6)は「12345」を返します。

STRINGFROMID

構文：STRINGFROMID(integer)

内部文字列表内で、指定された *integer* に関連付けられた組み込み文字列を、現在の言語で返します。

例：

STRINGFROMID(56)は「Complex」を返します。

STRINGFROMID(202)は「Real」を返します。

REPLACE

構文：REPLACE(object1, start, object2)

start で始まる object1 の一部を object2 で置き換えます。オブジェクトは行列、ベクトル、または文字列が可能です。

例：

REPLACE("12345",”3”,”99”)は「12995」を返します。

描画

HP Prime には G0 から G9 までの 10 個の組み込みグラフィックス変数があります。G0 は常に現在の画面グラフィックスです。

G1 から G9 は、グラフィックスを使用するアプリケーションをプログラミングするときに、一時グラフィック オブジェクト（短縮して GROB と呼ばれます）を保存するために使用できます。これらは一時的であるため、電卓の電源をオフにするとクリアされます。

グラフィックス変数を変更するために、26 個の関数を使用できます。そのうちの 13 個は、現在のアプリケーション内で変数 Xmin、Xmax、Ymin、および Ymax で定義されたデカルト平面を使用するデカルト座標を操作します。

残りの 13 個は、ピクセル 0,0 が GROB の左上ピクセル、320, 240 が右下であるピクセル座標を操作します。2 番目の関数セットの関数名には、_P サフィックスが付きます。

C→PX

デカルト座標から画面座標に変換します。

構文：`C→PX(x,y)` または `C→PX({x,y})`

DRAWMENU

構文：`DRAWMENU({string1, string2, …, string6})`

ディスプレイの下部に 6 ボタン メニュー（ラベルは string1、string2、…、string6）を描画します。

例：

`DRAWMENU("ABC", "", "DEF")` は、1 番目と 3 番目のボタンにそれぞれ ABC および DEF ラベルが付いたメニューを作成します。他の 4 つのメニュー キーは空白です。

FREEZE

構文：`FREEZE`

キーが押されるまで、プログラム実行を一時停止します。これにより、プログラム実行終了後に画面が再描画されることがなくなります。画面上の変更された表示はそのままユーザーが見ることができます。

PX→C

画面座標からデカルト座標に変換します。

RGB

構文：`RGB(R, G, B, [A])`

描画関数の色パラメーターとして使用できる整数を赤、緑、および青コンポーネント値に基づいて（それぞれ 0 から 255）返します。

アルファが 128 より大きい場合、透明としてフラグされた色を返します。Prime にアルファ チャネル ブレンドはありません。

例：

`RGB(255,0,128)` は 16711808 を返します。

`RECT(RGB(0,0,255))` は青い画面を作ります j。

`LINE(0,0,8,8,RGB(0,255,0))` は緑色の行を描画します。

ピクセルおよびデカルト

ARC_P、ARC

構文：`ARC(G, x, y, r [ , a1, a2, c])`

構文：`ARC_P(G, x, y, r [ , a1, a2, c])`

中心点が x,y、半径が r、色が c、開始角度が a1 で終了角度が a2 の弧または円を G に描画します。

G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。

r はピクセルで指定されます。

c はオプションで、指定されない場合は黒が使用されます。#RRGGBB（HTML での色の指定と同様)のように指定してください。

a1 および a2 は現在の角度モードに従い、オプションです。初期設定は完全な円です。

例：

`ARC(0,0,60,0,π,RGB(255,0,0))` は、中心が(0,0)（現在の[プロットのセットアップ]ウィンドウを使用）で半径が 60 ピクセルの赤色の半円を描画します。半円は、0 から π まで反時計回りに描画されます。

BLIT_P、BLIT

構文：`BLIT([trgtGRB, dx1, dy1, dx2, dy2], [srcGRB, sx1, sy1, sx2, sy2, c, alpha])`

構文：`BLIT_P ([trgtGRB, dx1, dy1, dx2, dy2], [srcGRB, sx1, sy1, sx2, sy2, c, alpha])`

点（sx1, sy1）（この点を含む）から点（sx2, sy2）（この点を含まない）の srcGRB の領域を、点（dx1, dy1）と（dx2, dy2）の間の trgtGRB の領域にコピーします。実行するとき、現在の領域を取得するために、sx1 と sx2 のそれぞれに 1 が加算されます。色 c のピクセルは srcGRB からコピーしません。

数値 alpha は、0（透明）から 255（不透明）の範囲にする必要があります。alpha はソース ビットマップの透明度を表し、アルファ チャネルと呼ばれます。

trgtGRB はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。

srcGRB はいずれのグラフィックス変数でもかまいません。

dx2, dy2 はオプションで、指定されない場合は、コピー先領域がコピー元領域のサイズと同じになるように計算されます。

sx2, sy2 はオプションで、指定されない場合は、srcGRB の右下になります。

sx1, sy1 はオプションで、指定されない場合は、srcGRB の左上になります。

dx1, dy1 はオプションで、指定されない場合は、trgtGRB の左上になります。

c は#RRGGBB で指定されるどの色でもかまいません。指定されない場合は、srcGRB のすべてのピクセルがコピーされます。

alpha はオプションで、指定されない場合は、初期設定で 255（不透明）になります。

注記：trgtGRB と srcGRB に同じ変数を使用すると、コピー元とコピー先が重なったときに結果が予測できない場合があります。

c および alpha の両方を使用する場合は、システムが各パラメーターの目的を区別できるように、ソースの x 座標および y 座標も指定することをおすすめします。

DIMGROB_P、DIMGROB

構文：`DIMGROB_P(G, w, h, [color])` または `DIMGROB_P(G, list)`

構文：`DIMGROB(G, w, h, [color])` または `DIMGROB(G, list)`

GROB G の大きさを w × h に設定します。リスト変数で指定された色またはグラフィック データを使用してグラフィック G を初期化します。グラフィック データを使用してグラフィックが初期化される場合、リストは整数のリストです。各整数は基数 16 で表示され、16 ビットごとに 1 つの色を表します。

色は A1R5G5B5 形式です（アルファ チャネルに 1 ビット、R、G、B に 5 ビット）。

FILLPOLY_P、FILLPOLY

構文：`FILLPOLY_P([G],{(x1, y1), (x2, y2),…(xn, yn)}, Color, [Alpha])`

構文：`FILLPOLY([G],{(x1, y1), (x2, y2),…(xn, yn)}, Color, [Alpha])`

点のリストで定義された多角形の場合、RGB 番号 Color で定義された色で多角形を塗りつぶします。Alpha が 0 から 255（両端を含む）の間の整数で指定された場合、対応する透明レベルの多角形が描画されます。リストの代わりに点のベクトルを使用できます。この場合、点は複素数で表現できます。

例：

`FILLPOLY_P({(20,20), (100, 20), (100, 100), (20, 100)}, #FF, 128)` は、側面が 80 ピクセル、紫色で透明レベルが 128 の四角形を、画面の左上付近に描画します。

GETPIX_P、GETPIX

構文：`GETPIX([G], x, y)`

構文：`GETPIX_P([G], x, y)`

座標 x,y のピクセル G の色を返します。

G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0（現在のグラフィック）です。

GROBH_P、GROBH

構文：`GROBH(G)`

構文：`GROBH_P(G)`

G の高さを返します。

G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。

GROBW_P、GROB

構文：`GROBW(G)`

構文：`GROBW_P(G)`

G の幅を返します。

G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。

INVERT_P、INVERT

構文：`INVERT([G, x1, y1, x2, y2])`

構文：`INVERT_P([G, x1, y1, x2, y2])`

選択された領域の反転ビデオを実行します。G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。

x2, y2 はオプションで、指定されない場合は、グラフィックの右下になります。

x1, y1 はオプションで、指定されない場合は、グラフィックの左上になります。1 つの x,y ペアのみが指定される場合、左上を参照します。

LINE_P、LINE

構文：`LINE_P([G], x1, y1, x2, y2, [color])`

構文：`LINE_P([G],points_definition, lines_definitions, otation_matrix or {rotation_matrix or -1, ["N"], [{eye_x, eye_y, eye_z} or -1], [{3Dxmin, 3Dxmax, 3Dymin, 3Dymax, 3Dzmin, 3Dzmax}]}, [zstring])`

構文：`LINE_P([G],pre_rotated_points, line_definitions, [zstring])`

構文：`LINE([G], x1, y1, x2, y2, [color])`

構文：`LINE([G],points_definition, lines_definitions, otation_matrix or {rotation_matrix or -1, ["N"], [{eye_x, eye_y, eye_z} or -1], [{3Dxmin, 3Dxmax, 3Dymin, 3Dymax, 3Dzmin, 3Dzmax}]}, [zstring])`

構文：`LINE([G],pre_rotated_points, line_definitions, [zstring])`

基本形式の LINE_P では、指定された色を使用してグラフィック内の指定されたピクセル座標間に 1 本の線を引きます。

高度な形式の LINE_P では、三角形の頂点のポテンシャル 3D 変換によって同時に複数の線をレンダリングできます。

これは主として、頂点および線のセットを持っていて、それらすべてを一度に（すばやく）表示したい場合に使用されます。

`points_definition` は、点定義のリストまたは行列です。各点は 2 から 4 個の数字で定義されます：x、y、z および color。有効な点定義には、複数の形式を含めることができます。いくつかの例を示します：[x, y, z, c], {x, y, z, c}, {x, y, #c}, {(x, y), c}, (x,y)。リストの代わりに点のベクトルを使用できます。この場合、点は複素数で表現できます。

`line_ definitions` は、行定義のリストまたは行列です。各行は 2 から 4 個の数字で定義されます：p1、p2、color、および alpha。p1 および p2 は行を定義する 2 つの点の `points_definition` のインデックスです。color は点ごとの色定義を上書きする場合に使用されます。色ではなく、Alpha を指定する必要がある場合、色には-1 を使用します。

`{Color, [Alpha], line_1, ..., line_n}`は、各線に同じ色を再指定することを回避する場合にも有効な形式です。

`rotation_matrix` は、通常の 3D または 4D 幾何を使用して点の回転および変換を指定する、2*2 から 3*4 のサイズの行列です。

`{eye_x, eye_y, eye_z}`は視点の位置を定義します（投影）。

`{3Dxmin, 3Dxmax, 3Dymin, 3Dymax, 3Dzmin, 3Dzmax}`は、事前に変換したオブジェクト上で 3D クリッピングを実行する場合に使用されます。

各点は、`rotation_matrix` 倍することで回転および変換されます。さらに、次の式で計算された視点の位置に基づいて表示平面に投影されます：x=eye_z/z*x-eye_x および y=eye_z/ z*y-eye_y。

3D クリッピング データが指定されると、各線は 3D でクリッピングされます。

「N」が指定された場合、Z 座標は回転の後に 0 から 255 の間で正規化されるので、z クリッピングが簡単にできます。

zstring が指定された場合、z 値文字列を使用してピクセル単位の z クリッピングが発生します（以下を参照してください）。

LINE_P は、すべての変換された点を含む文字列を返します。1 つの行内で同じ点と変換を使用して TRIANGLE または LINE を複数回呼び出す場合、points_definition をこの文字列で置き換えて、TRIANGLE および LINE への後続の呼び出しで変換定義を削除することによって、実行できます。

ZString の詳細：

TRIANGLE_P([G]) は z クリッピングに適合する文字列を返します。

z クリッピングを使用するには、TRIANGLE_P を呼び出して z クリッピング文字列を作成します（各ピクセルに対して 255 で初期化）。それから、三角形の各頂点に対応する z（0 から 255）値で LINE_P を呼び出すことができます。LINE_P はすでに描画されているピクセルよりさらにピクセルを描画することはありません。ZString は必要に応じて自動的に更新されます。

PIXOFF_P、PIXOFF

構文：PIXOFF([G], x, y)

構文：PIXOFF_P([G], x, y)

座標 x,y のピクセル G の色を白に設定します。G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0（現在のグラフィック）です。

PIXON_P、PIXON

構文：PIXON([G], x, y [,color])

構文：PIXON_P([G], x, y [,color])

座標(x,y)のグラフィックス変数 G のピクセルの色を入力した色に設定します。G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0（現在のグラフィックス）です。

オプションの color は、aaRRGGBB 形式の任意の 16 進整数で入力できます。これは、上位バイトがアルファ チャネルになっている RGB カラーです。アルファ チャネルの数値は、0（不透明）から 255（透明）の範囲の任意の整数にできます。color が指定されない場合は、初期設定の黒色が使用されます。

RECT_P、RECT

構文：RECT([G, x1, y1, x2, y2, edgecolor, fillcolor])

構文：RECT_P([G, x1, y1, x2, y2, edgecolor, fillcolor])

外周には edgecolor、内側には fillcolor を使用して、点 x1,y1 と x2,y2 の間で G 上に長方形を描画します。

G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0（現在のグラフィック）です。

x1, y1 はオプションです。初期設定値はグラフィックの左上を表します。

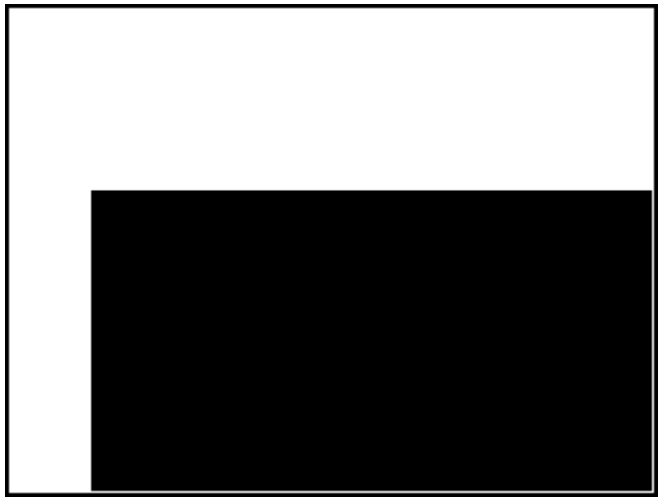
x2, y2 はオプションです。初期設定値はグラフィックの右下を表します。

edgecolor および fillcolor は、#RRGGBB で指定されるどの色でもかまいません。両方ともオプションで、指定されない場合、fillcolor は初期設定で edgecolor になります。

GROB を消去するには、RECT(G) を実行します。画面をクリアするには、RECT() を実行します。

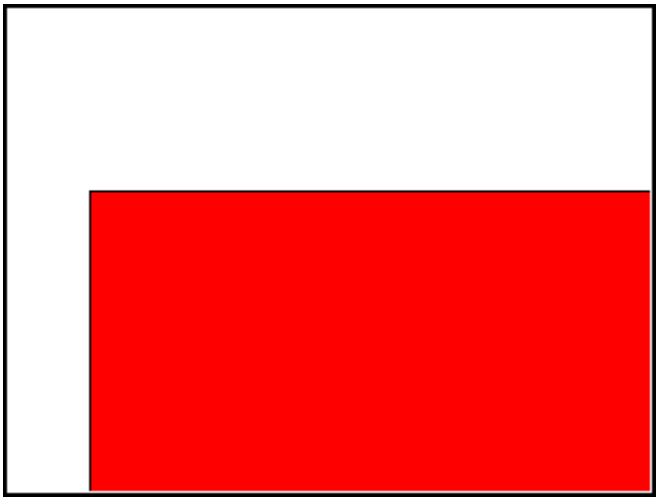
複数のオプション パラメーターを持つコマンド（RECT など）でオプション引数が指定された場合, 指定された引数はまず左端のパラメーターに対応します。たとえば、以下のプログラムでは、RECT_P コマンドの引数 40 および 90 は、x1 および y1 に対応します。引数#000000 は edgecolor に対応します。追加引数が 1 つだけのためです。追加引数が 2 つだった場合、それらは hedgecolor および fillcolor ではなく、x2 および y2 を参照します。このプログラムは、辺が黒で、黒で塗りつぶされた長方形を生成します。

```
EXPORT BOX()

BEGIN

RECT();

RECT_P(40,90,#0 00000);

WAIT;

END;
```



以下のプログラムも RECT_P コマンドを使用します。この場合、引数 320 および 240 のペアは x2 および y2 に対応します。このプログラムは、辺が黒で、赤で塗りつぶされた長方形を生成します。

```
EXPORT BOX()

BEGIN

RECT();

RECT_P(40,90,32 0,240,#000000,# FF0000);

WAIT;

END;
```



SUBGROB_P、SUBGROB

構文：`SUBGROB(srcGRB [ ,x1, y1, x2, y2], trgtGRB)`

構文：`SUBGROB_P(srcGRB [ ,x1, y1, x2, y2], trgtGRB)`

点 x1,y1 および点 x2,y2 の間の srcGRB 領域を trgtGRB にコピーするように設定します。

srcGRB はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。

trgtGRB は、G0 を除く、グラフィックス変数のいずれでもかまいません。

x2, y2 はオプションで、指定されない場合は、srcGRB の右下になります。

x1, y1 はオプションで、指定されない場合は、srcGRB の左上になります。

例：`SUBGROB(G1, G4)` は G1 を G4 にコピーします。

TEXTOUT_P、TEXTOUT

構文：`TEXTOUT(text [ ,G], x, y [ ,font, c1, width, c2])`

構文：`TEXTOUT_P(text [ ,G], x, y [ ,font, c1, width, c2])`

テキストを、色 c1 を使用し、グラフィックス G の位置 x, y に、フォントを使用して描画します。ピクセル幅 width を超えるテキストは描画しません。また、色 c2 を使用してテキストを描画する前に背景は消去しません。

G はオプションで、いずれのグラフィックス変数でもかまいません。初期設定は G0 です。このコマンドは、テキスト出力の最後のピクセルの x 座標を返します。

font には次の値を指定できます：

0：[ホーム設定]画面で選択されている現在のフォント。1：小フォント。2：大フォント。font はオプションで、指定されない場合は、[ホーム設定]画面で選択されている現在のフォントです。

c1 は#RRGGBB で指定されるどの色でもかまいません。初期設定は黒（#000000）です。

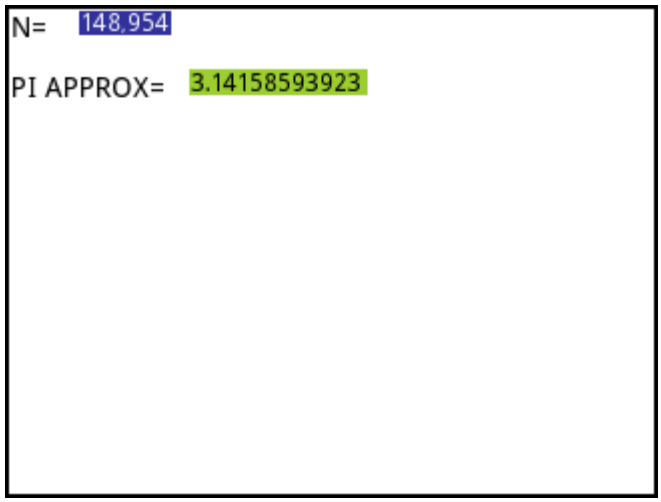
width はオプションで、指定されない場合はクリッピングは行われません。

c2 は#RRGGBB で指定されるどの色でもかまいません。c2 はオプションです。指定されない場合は、背景は消去されません。

例：

以下のプログラムは、逆正接(1)の級数を使用して π の連続近似値を表示します。テキストの色および背景色が指定されています（テキストの幅は 100 ピクセルに制限されています）。

```
EXPORT PISERIES()
BEGIN
LOCAL sign;
K:=2;
A:=4;
sign:=-1;
RECT();
TEXTOUT_P("N=",0,0);
TEXTOUT_P("PI APPROX=",0,30);
REPEAT
A+sign*4/(2*K-1)▶A;
TEXTOUT_P(K ,35,0,2,#FFFFFF,100,#333399);
TEXTOUT_P(A ,90,30,2,#000000,100,#99CC33);
sign*-1▶sign;
K+1▶K;
UNTIL 0;
END;
```



プログラムは、ユーザーが On Off を押して終了するまで、実行されます。

TRIANGLE_P、TRIANGLE

構文：`TRIANGLE_P([G], x1, y1, x2, y2, x3, y3, c1, [c2, c3], [Alpha], ["ZString", z1, z2, z3])`

構文：TRIANGLE_P([G], {x1, y1, [c1], [z1]}, {x2, y2, [c2], [z2]},{x3, y3, [c3], [z3]}, ["ZString"])

構文：TRIANGLE_P([G],points_definition, triangle_definitions, rotation_matrix or {rotation_matrix or -1, ["N"], [{eye_x, eye_y, eye_z} or -1], [{3Dxmin, 3Dxmax, 3Dymin, 3Dymax, 3Dzmin, 3Dzmax}]}, [zstring])

構文：TRIANGLE_P([G],pre_rotated_points, triangle_definitions, [zstring])

構文：TRIANGLE_P([G])

基本形式の TRIANGLE では、指定された色と透明度（0 ≤ Alpha ≤ 255）を使用してグラフィック内の指定されたピクセル座標間に 1 つの三角形を描画します。3 つの色が指定された場合、頂点間で色を混ぜ合わせます。

高度な形式の TRIANGLE_P では、三角形の頂点のポテンシャル 3D 変換によって同時に複数の三角形をレンダリングできます。

これは主として、頂点および三角形のセットを持っていて、それらすべてを一度に（すばやく）表示したい場合に使用されます。

points_definition は、点定義のリストまたは行列です。各点は 2 から 4 個の数字で定義されます：x、y、z および color。有効な点定義には、複数の形式を含めることができます。いくつかの例を示します：[x, y, z, c]、{x, y, z, c}、{x, y, #c}、{(x, y), c}、(x,y)...リストの代わりに複数の点のベクトルを使用できます。この場合、点は複素数で表現できます。

triangle_ definitions は、三角形定義のリストまたは行列です。各三角形は 3 から 5 個の次の数字で定義されます：p1、p2、p3、color、および alpha。p1、p2、および p3 は三角形を定義する 3 つの点の points_definition のインデックスです。color は点ごとの色定義を上書きする場合に使用されます。色ではなく、Alpha を指定する必要がある場合、色には-1 を使用します。

{Color, [Alpha], triangle_1, ..., triangle_n}は、各三角形に同じ色を再指定することを回避する場合にも有効な形式です。

rotation_matrix は、通常の 3D および 4D 幾何を使用して点の回転および変換を指定する、2*2 から 3*4 のサイズの行列です。

{eye_x, eye_y, eye_z}は視点の位置を定義します（投影）。

{3Dxmin, 3Dxmax, 3Dymin, 3Dymax, 3Dzmin, 3Dzmax}は、事前に変換したオブジェクト上で 3D クリッピングを実行する場合に使用されます。

各点は、rotation_matrix 倍することで回転および変換されます。さらに、以下の式で計算された視点の位置に基づいて表示平面に投影されます：x=eye_z/z*x-eye_x および y=eye_z/ z*y-eye_y。

各三角形は、3D クリッピング データが指定されると、3D でクリッピングされます。

「N」が指定された場合、Z 座標は回転の後に 0 から 255 の間で正規化されるので、z クリッピングが簡単にできます。

zstring が指定された場合、z 値文字列を使用してピクセル単位の z クリッピングが発生します（以下を参照してください）。

TRIANGLE_P は、すべての変換された点を含む文字列を返します。1 つの行内で同じ点と変換を使用して TRIANGLE または LINE を複数回呼び出す場合、points_definition をこの文字列で置き換えて、TRIANGLE および LINE への後続の呼び出しで変換定義を削除することによって、実行できます。

ZString の詳細：

TRIANGLE_P([G]) は z クリッピングに適合する文字列を返します。

z クリッピングを使用するには、TRIANGLE_P([G]) を呼び出して z クリッピング文字列を作成します（各ピクセルに対して 255 で初期化）。それから、三角形の各頂点に対応する z（0 から 255）の値で TRIANGLE_P を呼び出すことができます。TRIANGLE_P([G]) はすでに描画されているピクセルよりもさらにピクセルを描画することはありません。ZString は必要に応じて自動的に更新されます。

行列

行列コマンドによっては、コマンドが適用される行列変数名をその引数として使用するものもあります。有効な名前は、グローバル変数 M0 から M9、または行列を含むローカル変数です。行列をコマンドへの引数として直接入力することもできます。

ADDCOL

構文：ADDCOL(matrixname, vector, column_number)

指定された行列の column_number の前に挿入された新しい列に、vector の値を挿入します。ベクトル内の値の数は行列内の行数と同じである必要があります。

ADDROW

構文：ADDROW(matrixname, vector, row_number)

指定された行列の row_number の前に挿入された新しい行に、vector の値を挿入します。ベクトル内の値の数は行列内の列数と同じである必要があります。

DELCOL

構文：DELCOL(name ,column_number)

行列名から *column column_number* を削除します。

DELROW

構文：DELROW(name ,row_number)

行列名から *row row_number* を削除します。

EDITMAT

構文：EDITMAT(matrix variable, [title], [read only]) または EDITMAT(matrix, [title], [read only])

指定した行列を表示または編集できます。

行列変数（M0 ~ M9）を使用する場合、 OK をタップすると変数が更新されます。

オプションのタイトルは、"タイトル"または{"タイトル", ["行名"], ["列名"]}のどちらかの形式で指定します。入力した場合、"タイトル"は行列エディターの上部に表示されます。"行名"および"列名"を入力すると、エディターでは行見出しおよび列見出しとして使用されます。

read only に 0 以外の値を設定した場合は、行列を表示することのみできます。つまり、ユーザーは行列を編集できません。

EDITMAT はコマンドが完了するとすぐに行列を返します。プログラムで使用する場合は、 OK をタップするとプログラムに戻ります。

REDIM

構文：REDIM(name, size)

指定された行列（name）またはベクトルの次元を size に変更します。行列の場合、size は 2 つの整数（n1、n2）のリストです。ベクトルの場合、size は 1 つの整数（n）を含むリストです。行列内の既存の値は保持されます。塗りつぶしの値は 0 になります。

REPLACE

構文：REPLACE(name, start, object)

name に格納された行列またはベクトルの一部を、位置 start で始まるオブジェクトで置き換えます。行列の場合の start は、2 つの数字を含むリストです。ベクトルの場合、1 つの数字です。REPLACE は、リスト、グラフィックス、および文字列でも機能します。たとえば、REPLACE("123456", 2, "GRM") は「1GRM56」に置き換えられます。

SCALE

構文：SCALE(name, value, rownumber)

指定した行列の指定した row_number に value をかけます。

SCALEADD

構文：SCALEADD(name, value, row1, row2)

行列（name）の指定した row1 に value をかけてから、行列（name）の 2 番目に指定した row2 にこの結果を加えて、row1 を結果で置き換えます。

SUB

構文：SUB(name, start, end)

サブオブジェクト（リスト、行列、またはグラフィックの一部）を抽出して、name に格納します。start および end はそれぞれ、行列には 2 つの数、ベクトルまたはリストには 1 つの数、またはグラフィックスには順序付きペア（X,Y）を持つリストを使用して指定されます：SUB(M1{1,2},{2,2})

SWAPCOL

構文：SWAPCOL(name, column1, column2)

指定された行列（name）の column1 と column2 を入れ換えます。

SWAPROW

構文：SWAPROW(name, row1, row2)

指定された行列（name）内の row1 と row2 を入れ換えます。

アプリケーション関数

これらのコマンドを使用することで、HP アプリケーションを起動したり、現在のアプリケーションの表示を開いたり、[表示]メニュー内のオプションを変更したりできます。

STARTAPP

構文：STARTAPP("name")

名前が name のアプリケーションを起動します。これによって、アプリケーション プログラムが存在する場合、その START 関数が実行されます。アプリケーションの初期設定表示が起動されます。ユーザーがアプリケーション ライブラリン内で 開始 をタップしたときは、START 関数が常に実行されます。このコマンドはユーザー定義アプリケーションでも機能します。

例：STARTAPP("Function") は関数アプリケーションを起動します。

STARTVIEW

構文：STARTVIEW(n [,draw?])

現在のアプリケーションの n 番目の表示を起動します。*draw?* が真（0 でない）の場合、その表示の画面がただちに再描画されます。

表示番号（n）は、次のとおりです。

記号：0

プロット：1

数字：2

記号のセットアップ：3

プロットのセットアップ：4

数値のセットアップ：5

アプリケーションの情報：6

表示のメニュー：7

1 番目の特殊表示（分割画面のプロット詳細）：8

2 番目の特殊表示（分割画面のプロット表）：9

3 番目の特殊表示（自動スケール）：10

4 番目の特殊表示（小数）：11

5 番目の特殊表示（整数）：12

6 番目の特殊表示（三角法）：13

かっこの中の特殊表示は関数アプリケーションを参照し、他のアプリケーションでは異なる場合があります。特殊表示の番号は、該当するアプリケーションの[表示]メニューでの位置に対応します。1 番目の特殊表示は STARTVIEW(8) で起動され、2 番目は STARTVIEW(9) で、となります。

アプリケーションに固有でない表示も、0 より小さい n の値を指定して起動することができます。

ホーム画面：-1

ホーム設定：-2

メモリ管理者：-3

アプリケーション ライブラリ：-4

Matrix（行列）カタログ：-5

List（リスト）カタログ：-6

Program Catalog（プログラム カタログ）：-7

Note（注記）カタログ：-8

VIEW

構文：VIEW ("string"[,program_name])

BEGIN

Commands;

END;

カスタム オプションを**[表示]**メニューに追加します。**[string]**を選択すると、program_name が実行されます。567 ページの例セクションの「*DiceSimulation プログラム*」を参照してください。

整数

BITAND

構文：BITAND(int1, int2, … intn)

指定された整数のビット単位の論理積を返します。

例：BITAND(20,13) は 4 を返します。

BITNOT

構文：BITNOT(int)

指定された整数のビット単位の論理否定を返します。

例：BITNOT(47) は 549755813840 を返します。

BITOR

構文：BITOR(int1, int2, … intn)

指定された整数のビット単位の論理和を返します。

例：BITOR(9,26) は 27 を返します。

BITSL

構文：BITSL(int1 [,int2])

左にビット シフトします。1 つまたは 2 つの整数を入力とし、最初の整数のビットを 2 番目の整数で示す桁数だけ左にシフトした結果を返します。2 番目の整数がない場合、ビットは 1 桁だけ左にシフトされます。

例：

BITSL(28,2) は 112 を返します。

BITSL(5) は 10 を返します。

BITSR

構文：BITRL(int1 [,int2])

右にビット シフトします。1 つまたは 2 つの整数を入力とし、最初の整数のビットを 2 番目の整数で示す桁数だけ右にシフトした結果を返します。2 番目の整数がない場合、ビットは 1 桁だけ右にシフトされます。

例：

BITSR(112,2)は 28 を返します

BITSR(10)は 5 を返します。

BITXOR

構文：BITXOR(int1, int2, … intn)

指定された整数のビット単位の排他論理和を返します。

例：BITXOR(9,26)は 19 を返します。

B→R

構文：B→R(#integerm)

基数が m の整数を 10 進数整数（基数 10）に変換します。基数マーカー m は、b（2 進数の場合）、o（8 進数の場合）、または h（16 進数の場合）に指定できます。

例：B→R(#1101b)は 13 を返します

GETBASE

構文：GETBASE(#integer[m])

指定された整数の基数を、現在の初期設定の基数で返します。0 = 初期設定、1 = 2 進数、2 = 8 進数、3 = 16 進数です。

例：GETBASE(#1101b)は#1h（初期設定の基数が 16 進数の場合）を返し、GETBASE (#1101)は#0h を返します。

GETBITS

構文：GETBITS(#integer)

整数の符号化に使用されるビット数を返します。

整数が指定されない場合は、[ホーム設定]のページ 1 にある[整数]ボックスの現在値が使用されます。

例：

GETBITS(#22122)は 32 を返します

GETBITS(#1:45h)は 45 を返します

R→B

構文：R→B(integer)

初期設定の基数の表現で、10 進整数（基数 10）を整数に変換します。

例：R→B(13)は#1101b（初期設定の基数が 2 進の場合）または#Dh（初期設定の基数が 16 進の場合）を返します。

SETBITS

構文：SETBITS(#integer[m] [,bits])

integer を表現するビット数を設定します。有効な値は – 63 から 64 の範囲です。m または bits が省略された場合は、初期設定値が使用されます。

例：SETBITS(#1111b, 15)は#1111:15b を返します

SETBASE

構文：SETBASE(#integer[m][c])

基数 m で表現される integer を c で示される基数で表示します。c には 1（2 進数）、2（8 進数）、または 3（16 進数）を指定できます。パラメーター m には b（2 進数）、d（10 進数）、o（8 進数）、または（16 進数）を指定できます。m が省略された場合の入力は、初期設定の基数であると見なされます。同様に、c が省略された場合の出力は、初期設定の基数で表示されます。

例：SETBASE (#34o,1)は#11100b を返し、SETBASE (#1101)は#0h（初期設定の基数が 16 進数の場合）を返します。

I/O

I/O コマンドは、プログラムにデータを入力する場合、およびプログラムからデータを出力する場合に使用されます。ユーザーがプログラムと対話できます。

CHOOSE

構文：CHOOSE(var, "title", "item1", "item2",…,"itemn")

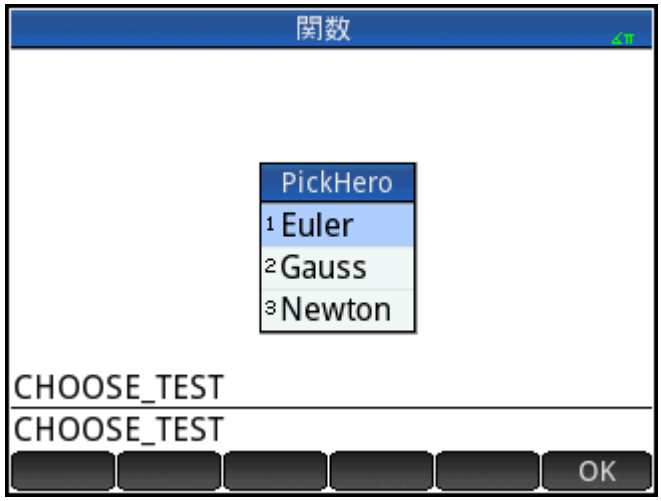
選択項目が含まれる、タイトル付きの選択ボックスを表示します。ユーザーがオブジェクトを選択した場合、指定された名前の変数が、選択されたオブジェクトの番号（整数 1、2、3、…）または 0（

をタップした場合）を含むように更新されます。

ユーザーがオブジェクトを選択した場合は真（ゼロ以外）を返し、そうでない場合は偽（0）を返します。

例：

```
CHOOSE

(N,"PickHero","Euler","Gauss","Newton");

IF N==1 THEN PRINT("You picked Euler"); ELSE IF N==2 THEN PRINT("You
picked Gauss");ELSE PRINT("You picked Newton");

END;

END;
```



CHOOSE の実行後に、N の値は 0、1、2、または 3 を含むように更新されます。IF THEN ELSE コマンドにより、選択された人の名前が端末に出力されます。

EDITLIST

構文：EDITLIST(listvar)

listvar を読み込むリスト エディターを起動し、指定されたリストを表示します。プログラミングで使用する場合、ユーザーが OK をタップするとプログラムに戻ります。

例：EDITLIST(L1) はリスト L1 を編集します。

EDITMAT

構文：EDITMAT(matrixvar)

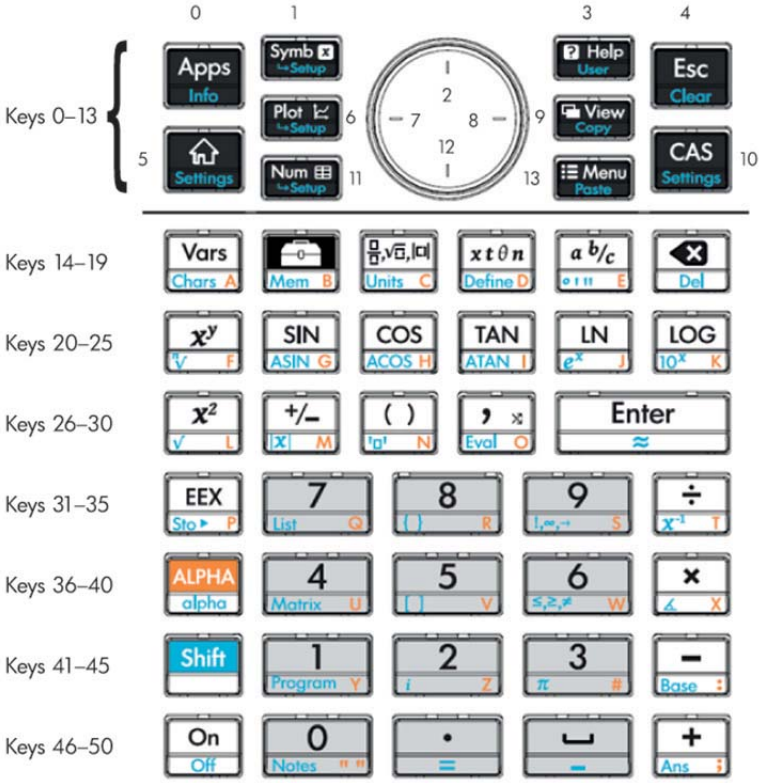
行列エディターを起動し、指定された行列を表示します。プログラミングで使用する場合、ユーザーが OK をタップするとプログラムに戻ります。

例：EDITMAT(M1) は行列 M1 を返します。

GETKEY

構文：GETKEY

キーボード バッファーの最初のキーの ID を返します。GETKEY を最後に呼び出してから押されたキーがない場合、-1 を返します。キー ID は 0 から 50 までの整数で、図 27-1 に示すように左上（キー 0）から右下（キー 50）の順に番号付けされます。



INPUT

構文： `INPUT(var,[“title”], [“label”], [“help”], [reset_value], [initial_value])`

構文： `INPUT({vars},[“title”], [{“labels”}], [{“help”}], [{reset_values}], [{initial_values}])`

このコマンドの簡単な形式では、指定されたタイトルと label という名前の 1 つのフィールドを含むダイアログ ボックスを開き、ヘルプを下部に表示します。ダイアログ ボックスには、[キャンセル]および[OK]メニュー キーが含まれます。ユーザーは label フィールドに値を入力できます。ユーザーが[OK]メニュー キーを押した場合は、変数 var は入力した値で更新され、1 が返されます。ユーザーが[キャンセル]メニュー キーを押した場合は、変数は更新されず、0 が返されます。

コマンドの複雑な形式では、複数フィールドのダイアログ ボックスを作成するためにリストが使用されます。var がリストである場合、各要素は次の構文を使用する変数名またはリストです。

- {var_name, real, [{pos}]}：チェック ボックス コントロールを作成します。real が>1 の場合、このチェック ボックスはラジオ グループ内の次の n -1 個のチェック ボックスでプールされます（n 個のチェック ボックスのうち、1 つのみをチェックできます）。
- {var_name, [allowed_types_matrix] , [{pos}]}：編集フィールドを作成します。[allowed_types_matrix]には、許可されるすべての種類をリストします（[-1]はすべての種類が許可されることを示します）。許可される種類が文字列のみの場合、そのエディションでは二重引用符は非表示になります。
- {var_name, {Choose items}, [{pos}]}：選択フィールドを作成します。

pos を指定する場合は、{field start in screen %, field width in screen%, line(starts at 0)}という形式のリストで指定します。これにより、フィールドの位置とサイズを正確に制御できます。pos は、ダイアロ

グ ボックス内のすべてのフィールドに指定するか、またはすべてのフィールドに指定しないかのどちらかにする必要があります。

1 ページにつきコントロールの最大行数は 7 行です。7 行を超えるコントロールは後続のページに置かれます。複数のページが作成された場合は、["title"]がタイトルのリストになります。

ISKEYDOWN

構文：`ISKEYDOWN(key_id);`

key_id で指定されたキーが現在押されている場合、true（0 以外）を返します。そうではない場合、false（0）を返します。

MOUSE

構文：`MOUSE[(index)]`

各潜在的ポインターの現在の位置を示す 2 つのリスト（ポインターが使用されない場合は空のリスト）を返します。出力は{x , y, original x, original y, type}です。type は 0（新規）、1（完了）、2（ドラッグ）、3（ストレッチ）、4（回転）、および 5（ロング クリック）です。

オプションのパラメーター index は、パラメーターが省略された場合に返される要素（x、y、original x など）のうち、n 番目の要素です。ポインター操作が発生しない場合は、-1 です。

MSGBOX

構文：`MSGBOX(expression or string [ ,ok_cancel?]);`

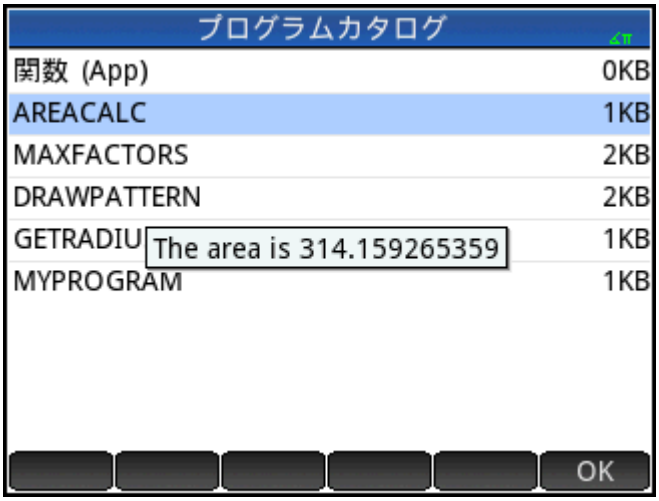
指定された式または文字列の値でメッセージ ボックスを表示します。

`ok_cancel?`が真の場合、 OK および キャンセ ボタンを表示し、そうでない場合は OK ボタンだけを表示します。`ok_cancel` の初期設定値は偽です。

ユーザーが OK をタップした場合は true（ゼロ以外）を返し、 キャンセ を押した場合は false（0）を返します。

```
EXPORT AREACALC()
BEGIN
LOCAL radius;
INPUT(radius, "Radius of Circle","r = ","Enter radius",1);
MSGBOX("The area is " +π*radius^2);
END;
```

ユーザーが半径に 10 を入力した場合、以下のようなメッセージ ボックスが表示されます。



PRINT

構文：`PRINT(expression or string);`

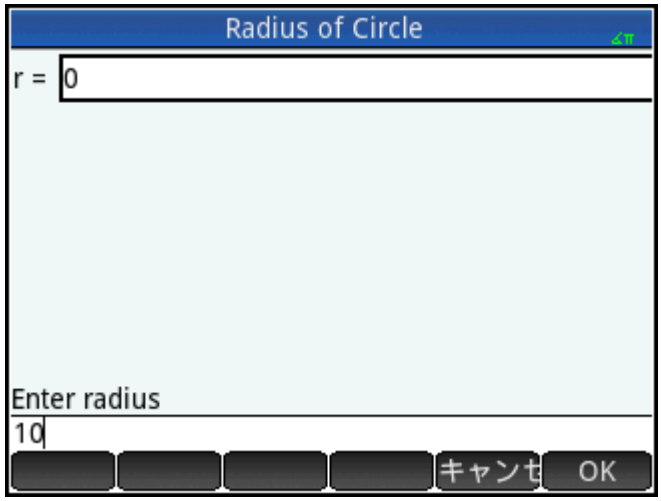
expression または string の結果を端末に出力します。

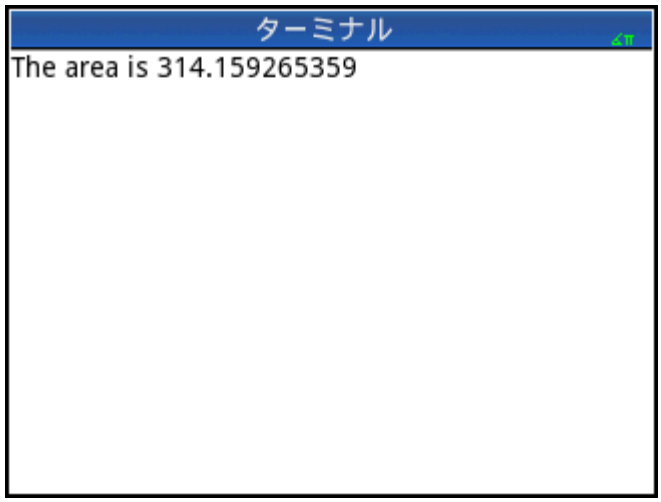
端末は、`PRINT` コマンドが実行されたときにのみ表示される、プログラム テキスト出力表示メカニズムです。表示されているときに、 または を押してテキストを表示したり、 Del を押してテキストを消去したり、他のキーを押して端末を非表示にしたりできます。 On Off を押すと、端末とのやり取りが停止します。引数を指定しないで PRINT を実行すると、端末がクリアされます。

[グラフィック]セクションにデータを出力するためのコマンドもあります。たとえば、コマンド `TEXTOUT` および `TEXTOUT_P` はテキスト出力に使用できます。

この例では、円の半径の値を入力することをユーザーに求め、端末に円の面積を出力します。

```
EXPORT AREACALC()
BEGIN
LOCAL radius;
INPUT(radius, "Radius of Circle","r = ","Enter radius",1);
PRINT("The area is " +π*radius^2);
END;
```





半径に LOCAL 変数を使用していること、ローカル変数に小文字を使用する命名規則に注目してください。このような規則に従うことで、プログラムの読みやすさが向上します。

WAIT

構文：WAIT(n);

プログラム実行を n 秒間一時停止します。引数がない場合、または n = 0 の場合、プログラム実行は 1 分間一時停止します。

その他

%CHANGE

構文：%CHANGE(x,y)

x から y への変化量（パーセント）です。

例：%CHANGE(20,50) は 150 を返します。

%TOTAL

構文：%TOTAL(x,y)

x に対する y のパーセントです。

例：%TOTAL(20,50) は 250 を返します。

CAS

構文：CAS.function() または CAS.variable

CAS を使用して、関数を実行するか変数を返します。

EVALLIST

構文：EVALLIST({list})

リストの各要素の内容を評価し、評価したリストを返します。

EXECON

構文：EXECON (&expr, List1, [List2,…])

1 つ以上のリスト内の要素に基づく新しいリストを、アンパサンド文字（&）を含む式に従い各要素を反復的に変更することで作成します。

例：

EXECON("&1+1",{1,2,3}) は{2,3,4}を返します

&の後に直接数字が続く場合、それはリスト内の位置を示します。例：

EXECON("&2-&1",{1, 4, 3, 5}"は{3, –1, 2}を返します。

上記の例では、&2 は各要素ペアの 2 番目の要素を示し、&1 は 1 番目の要素を示します。これらの間にあるマイナス演算子は、ペアがなくなるまで各ペアの 2 番目から 1 番目を引きます。この場合（リストが 1 つだけの場合）、&には 1 から 9（1 と 9 を含む）の数字のみを付加できます。

EXECON は複数のリストに対しても演算できます。例：

EXECON("&1+&2",{1,2,3},{4,5,6}) は{5,7,9}を返します。

上記の例では、&1 は最初のリストの要素を示し、&2 は 2 番目のリストの対応する要素を示します。これらの間にあるプラス演算子は、ペアがなくなるまで 2 つの要素を加算します。リストが 2 つの場合、&に付加する数は 2 桁にできます。この場合、1 番目の桁はリスト番号（左から右への順）を参照します。2 番目の桁は従来通り 1 から 9（1 と 9 を含む）まで指定できます。

EXECON は、指定されたリストの指定された要素から演算を開始することもできます。例：

EXECON("&23+&1",{1,5,16},{4,5,6,7}) は{7,12}を返します。

上記の例の&23 は、演算が 2 番目のリストの 3 番目の要素から開始されることを示します。その要素に対して最初のリストの 1 番目の要素が加算されます。この処理は、ペアがなくなるまで続きます。

→HMS

構文：→HMS(value)

10 進数値を 60 進法形式に変換します。つまり、結果を 60 で割った値が得られます。これにより、度、分、秒や時、分、秒単位の値を求めることができます。

例：→HMS(54.8763) は 54°52′ 34.68″ を返します。

HMS→

構文：HMS→(value)

60 進法形式で表現される値を 10 進法形式に変換します。

例：HMS→(54°52′ 34.68″) は 54.8763 を返します。

ITERATE

構文：ITERATE(expr, var, ivalue, #times)

expr を var について繰り返し評価します（var = ivalue から始めて、#times 回）。

例：ITERATE(X^2, X, 2, 3) は 256 を返します

TICKS

構文：TICKS

内部時計の値をミリ秒単位で返します。

TEVAL

構文：TEVAL(parameter)

パラメーターの評価にかかる秒単位の時間を返します。

TYPE

構文：TYPE(object)

オブジェクトのタイプを返します：

0：実数

1：整数

2：文字列

3：複素数

4：行列

5：エラー

6：リスト

8：関数

9：単位

14.?：cas オブジェクト。小数部は cas タイプです。

変数とプログラム

HP Prime には、ホーム変数、アプリケーション変数、CAS 変数、およびユーザー変数の 4 つの変数タイプがあります。[変数]メニュー（ 変数 ）からこれらの変数を取得できます。

ホーム変数の名前は予約されています。つまり、これらはシステムから削除できず、設計されているタイプと異なるタイプのオブジェクトを格納するために使用することはできません。たとえば、A – Z および θ は実数を格納するために予約されており、Z0 – Z9 は複素数を格納するために予約され、L0 – L9 はリストを格納するために予約されています。つまり、L8 に行列を格納したり、Z にリストを格納したりできません。

ホーム変数は、ホームおよびアプリケーション内で同じ値を保持します。つまり、システムが共有するグローバル変数です。プログラムで使用するには、このことを理解する必要があります。

アプリケーション変数名も予約されていますが、いくつかのアプリケーションで同じアプリケーション変数名を共有できます。ただし、その変数が現在のアプリケーションのものでない場合は、変数の名前を修飾する必要があります。たとえば、現在のアプリケーションが関数アプリケーションの場合、`Xmin` は関数アプリケーションの[プロット表示]に最小の x-value を返します。[極座標]アプリケーションの[プロット表示]に最小値が必要な場合は、`Polar.Xmin` を入力する必要があります。アプリケーション変数は、アプリケーションと対話的に連携するときに作成する定義と設定を表します。アプリケーションで作業するときは、アプリケーション関数はアプリケーション変数にも結果を格納する場合があります。プログラムでのアプリケーション変数は、アプリケーションのデータをカスタマイズするために編集したり、アプリケーションの操作から結果を取得したりするために使用されます。

CAS 変数はホーム実数変数 A – Z に似ていますが、小文字であることと、ホーム表示ではなく CAS 表示で使用するように設計されている点が異なります。さらに、ホーム変数とアプリケーション変数には常に値が含まれていますが、CAS 変数は単なる記号で特定の値が含まれていないことがある、という違いがあります。CAS 変数は、ホーム変数やアプリケーション変数のように型付けされません。たとえば、CAS 変数 t には実数、リスト、ベクトルなどを格納できます。CAS 変数に格納されている値がある場合は、ホーム表示から CAS 変数を呼び出すと、その内容が返されます。

ユーザー変数は、ユーザーによって直接作成された変数、またはユーザーがユーザー プログラムからエクスポートすることで作成された変数です。プログラムが他の電卓および他のプログラムとやり取りするためのいくつかのメカニズムの、1 つを提供します。プログラムで作成されたユーザー変数は、そのプログラムにローカル、またはグローバルにできます。プログラムからエクスポートされた変数は、**[変数]**メニューのユーザー変数内、エクスポート元プログラムの横に表示されます。ユーザー変数は複数文字でもかまいませんが、一定の規則に従う必要があります。詳しくは、556 ページの変数と可視性を参照してください。

CAS 変数のようなユーザー変数は型付けされていないため、さまざまなタイプのオブジェクトを格納できます。

以下のセクションでは、プログラムでアプリケーション変数を使用する方法について説明します。各アプリケーション変数とその内容について説明します。すべてのホーム変数およびアプリケーション変数のリストについては、「変数」章を参照してください。プログラムでのユーザー変数については、556 ページの HP Prime プログラミング言語 を参照してください。

アプリケーション変数

すべてのアプリケーション変数がすべてのアプリケーションで使用されるわけではありません。たとえば、S1Fit は、極座標アプリケーションでのみ使用されます。ただし、多くの変数は、関数、高度なグラフ作成、パラメトリック、極座標、数列、解く、1 変数統計、および 2 変数統計アプリケーションで共有されています。ある変数がこれらのアプリケーションの一部で利用できない場合、またはこれらのアプリケーション（または他の一部のアプリケーション）の一部でのみ利用できる場合には、その変数を使用できるアプリケーションのリストが変数名の下に表示されます。

以下のセクションでは、使用される表示ごとにアプリケーション変数のリストを示します。[変数]メニューに表示されるカテゴリごとの変数一覧を見るには、「変数」章の「アプリケーション変数」セクションを参照してください。

現在のアプリケーション変数

これらの変数により、ユーザーは現在アクティブなアプリケーションに関連付けられているデータおよびファイルにアクセスできます。

AFiles

HP Prime アプリケーションには、それぞれ任意の数のファイルを関連付けることができます。これらのファイルはアプリケーションとともに送信されます。たとえば、icon.png という名前のファイルがアプリケーションに関連付けられている場合、このファイルはアプリケーション ライブラリ内でアプリケーション アイコン用に使用されます。

`AFiles` はこれらのファイルすべてのリストを返します。

`AFiles("name")` は与えられた名前を持つファイルの内容を返します。

`AFiles("name"):= object` は与えられた名前のファイルにオブジェクトを保存します。

AFilesB

HP Prime アプリケーションには、それぞれ任意の数のファイルを関連付けることができます。これらのファイルはアプリケーションとともに送信されます。AFilesB は、AFiles 変数のバイナリー版です。

`AFilesB` はアプリケーションに関係付けられたすべてのファイルのリストを返します。

`AFilesB("name")` は与えられた名前を持つファイルのサイズを返します。

`AFilesB("name", position, [nb])` は与えられた名前のファイルの位置 position から nb バイトを読み取り返します（position は 0 で開始）。

`AFilesB("name", position):= value` または`{values...}`は、位置 position から n バイトを与えられた名前のファイルに保存します。

ANote

ANote は、HP アプリケーションに関連付けられた注記を返します。これは、ユーザーが Shift

Apps Info を押したときに表示される注記です。

`ANote:="string"`は、アプリケーションに関係付けられた注記が文字列を含むように設定します。

AProgram

AProgram は、HP Prime アプリケーションに関連付けられたプログラムを返します。

`AProgram:="string"`は、アプリケーションに関連付けられたプログラムが文字列を含むように設定します。

AVars

AVars は HP Prime アプリケーションに関係付けられたすべての変数の名前のリストを返します。

`AVars(n)` はアプリケーションに関係付けられた n 番目の変数の内容を返します。

`AVars("name")` はアプリケーションに関係付けられた指定の変数の内容を返します。

`AVars(n または"name"):= value` は、指定されたアプリケーション変数が、与えられた値を含むように設定します。「name」が既存の変数でない場合、新しい変数が作成されます。

AVars("name"):= value を使ってアプリケーション変数を作成した後は、変数名を入力することによってその変数を使用できます。

DelAVars

DelAVars(n,または"name")は指定されたアプリケーション変数を削除します。

DelAFiles

DelAFiles("name")は HP アプリケーションに関連付けられた指定のファイルを削除します。

[プロット表示]変数

Axes

軸をオンまたはオフにします。

[プロットの設定]表示では、AXES をチェック（またはチェック解除）します。

プログラムでは、以下のように入力します。

0 ▶ Axes：軸をオンにします。

1 ▶ Axes：軸をオンにします。

Cursor

カーソルのタイプを設定します。（背景が無地の場合、反転または点滅を使用できます。）

[プロット設定]表示では、**[Cursor]**を選択します。

プログラムでは、以下のように入力します。

0 ▶ Cursor：実線の十字（初期設定）。

1 ▶ Cursor：十字を反転します。

2 ▶ Cursor：十字を点滅します。

GridDots

[プロット表示]の背景ドット グリッドのオンとオフを切り替えます。[プロット設定]表示では、GRID DOTS をチェック（またはチェック解除）します。プログラムでは、以下のように入力します。

0 ▶ GridDots：グリッドドットをオンにします（初期設定）。

1 ▶ GridDots：グリッドドットをオフにします。

GridLines

[プロット表示]の背景線グリッドのオンとオフを切り替えます。

[プロット設定]表示では、GRID LINES をチェック（チェック解除）します。

プログラムでは、以下のように入力します。

0 ▶ GridDots：グリッド線をオンにします（初期設定）。

1 ▶ GridLines：グリッド線をオフにします

Hmin/Hmax

1 変数統計

ヒストグラムの棒の最小値と最大値を定義します。

1 変数統計の[プロット設定]表示では、HRNG の値を設定します。

プログラムでは、以下のように入力します。

n_1 ▶ Hmin

n_2 ▶ Hmin

ここで、$n_1 < n_2$

Hwidth

1 変数統計

ヒストグラムの棒の幅を設定します。

1 変数統計の[プロット設定]表示では、Hwidth の値を設定します。

プログラムでは、以下のように入力します。

n ▶ Hwidth、ここで n > 0

Labels

[プロット表示]に X および Y 範囲を示すラベルを描画します。

[プロット設定]表示では、Labels をチェック（またはチェック解除）します。

プログラムでは、以下のように入力します。

1 ▶ Labels：ラベルをオンにします（初期設定）。

2 ▶ Labels：ラベルをオフにします。

Method

関数、解く、パラメトリック、極座標、2 変数統計

グラフ方式を設定します：[adaptive]、[Fixed-Step セグメント]、または[Fixed-Step ドット]。

プログラムでは、以下のように入力します。

0 ▶ Method—[adaptive]を選択します。

1 ▶ Method：[Fixed-Step セグメント]を選択します。

2 ▶ Method：[Fixed-Step セグメント]を選択します。

Nmin/Nmax

数列

独立変数の最小値と最大値を定義します。

[プロットのセットアップ]表示に**[N 範囲]**フィールドとして表示します。[プロットのセットアップ]表示では、N 範囲の値を入力します。

プログラムでは、以下のように入力します。

n_1 ▶ Hmin

n_2 ▶ Hmin

ここで、$n_1 < n_2$

PixSize

幾何

幾何アプリケーションで各矩形ピクセルの大きさを設定します。[プロット表示]では、`Pixel Size` に正の値を入力します。

または、`PixSize:=n` を入力します。n>0 です。

Recenter

カーソル位置を中心にズームします。

[プロット ズーム設定倍率]から、**[Recenter]**をチェック（またはチェック解除）します。

プログラムでは、以下のように入力します。

`0` ▸ `Recenter`：「中央に再表示」をオンにします（初期設定）。

`1` ▸ `Recenter`：「中央に再表示」をオフにします。

S1mark-S5mark

2 変数統計

散布図に使用する目盛りを設定します。

2 変数統計の[プロット設定]表示では、`S1 Mark-S Mark` の 1 つを選択します。

ScrollText

幾何

[プロット表示]の現在のコマンドが自動または手動でスクロールするかどうかを指定します。[プロット表示]では、[テキストのスクロール]を選択またはクリアします。

手動でスクロールするときは `ScrollText:=0` を入力し、自動でスクロールするときは `ScrollText:=1` を入力することもできます。

SeqPlot

数列

階段状またはクモの巣プロットから選択できます。

[プロットのセットアップ]表示では、[シーケンスプロット]を選択してから、[階段状]または[クモの巣]を選択します。

プログラムでは、以下のように入力します。

`0` ▸ `SeqPlot`：階段状。

`1` ▸ `SeqPlot`：クモの巣。

θmin/θmax

極座標

最小および最大の独立値を設定します。

[プロット設定]表示では、θ Rng の値を入力します。

プログラムでは、以下のように入力します。

n_1 ▶ Hmin

n_2 ▶ Hmin

ここで、$n_1 < n_2$

θstep

極座標

独立変数のステップ サイズを設定します。

[プロット設定]表示では、θ Step の値を入力します。

プログラムでは、以下のように入力します。

n ▶ θstep

ここで、n > 0

Tmin/Tmax

パラメトリック

最小および最大の独立変数値を設定します。

[プロット設定]表示では、T 範囲の値を入力します。

プログラムでは、以下のように入力します。

n_1 ▶ Hmin

n_2 ▶ Hmin

ここで、$n_1 < n_2$

Tstep

パラメトリック

独立変数のステップ サイズを設定します。

[プロット設定]表示では、T ステップの値を入力します。

プログラムでは、以下のように入力します。

n ▶ Tstep

ここで、n > 0

X 目盛

水平軸の目盛間の距離を設定します。

[プロット設定]表示では、X 目盛の値を入力します。

プログラムでは、以下のように入力します。

n ▶ Xtick

ここで、n > 0

Y 目盛

垂直軸の目盛間の距離を設定します。

[プロット設定]表示では、Y 目盛の値を入力します。

プログラムでは、以下のように入力します。

`n ▶ Ytick`

ここで、n > 0

Xmin/Xmax

プロット画面の横方向の最小値および最大値を設定します。

[プロット設定]表示では、X 範囲の値を入力します。

プログラムでは、以下のように入力します。

n_1 ▶ Hmin

n_2 ▶ Hmin

ここで、$n_1 < n_2$

Ymin/Ymax

プロット画面の縦方向の最小値および最大値を設定します。

[プロット設定]表示では、Y 範囲の値を入力します。

プログラムでは、以下のように入力します。

n_1 ▶ Hmin

n_2 ▶ Hmin

ここで、$n_1 < n_2$

Xzoom

水平方向のズーム倍率を設定します。

[プロット表示]では、Menu Paste を押してから、ズーム を押します。**[倍率の設定]**までスクロールして選択してから、OK をタップします。X Zoom の値を入力して、OK をタップします。

プログラムでは、以下のように入力します。

`n ▶ Xzoom`

ここで、n > 0

初期設定値は 4 です。

Yzoom

[プロット表示]では、 Menu Paste を押してから、 ズーム を押します。**[倍率の設定]**までスクロールして選択してから、 OK をタップします。`Y` ズームの値を入力して、 OK をタップします。

プログラムでは、以下のように入力します。

`n ▶ Yzoom`

ここで、n > 0

初期設定値は 4 です。

[記号表示]変数

AltHyp

推論

仮説検証に使用する対立仮説を指定します。

[記号表示]では、`Alt Hypoth` のオプションを選択します。

プログラムでは、以下のように入力します。

`0 ▶ AltHyp` : μ < μ0

`1 ▶ AltHyp` : μ < μ0

`2 ▶ AltHyp` : μ < μ0

E0...E9

解く

方程式または式を含みます。[記号表示]では、`E0` から `E9` の 1 つを選択し、式または方程式を入力します。独立変数を選択するには、[数値表示]で選択します。

プログラムでは、以下（例）のように入力します。

`X+Y*X-2=Y ▶ E1`

F0...F9

関数

X に式を格納します。[記号表示]では、F0 から F9 の 1 つを選択して、式を入力します。

プログラムでは、以下（例）のように入力します。

`SIN(X) ▶ F1`

H1...H5

1 変数統計

[1 変数統計]記号変数は H1 ~ H5 です。これらの変数には 1 変数統計分析のためのデータ値が含まれています。たとえば、H1(n)は H1 分析用のデータ セット内の n 番目の値を返します。引数がない場合、H1 は、H1 を定義するオブジェクトのリストを返します。これらのオブジェクトは指定される順序で次のようになります。

- データ リストを定義する、一重引用符で囲まれた式（または空の二重引用符）
- オプションでデータ リスト内の各値の頻度を定義する、一重引用符で囲まれた式（または空の二重引用符）
- プロット タイプ番号
- オプション番号
- プロットの色

プロット タイプ番号は 1 から 9 の整数で、変数 H1 から H5 のそれぞれで使用する統計プロット タイプを示します。各番号に対応するタイプは次のとおりです。

- **[1]**：ヒストグラム（初期設定）
- **[2]**：箱ひげ
- **[3]**：正規確率
- **[4]**：線
- **[5]**：棒
- **[6]**：パレート
- **[7]**：コントロール
- **[8]**：ドット
- **[9]**：茎葉

オプション番号は 0 から 2 の整数で、プロット タイプで指定できるオプションを示します。各番号に対応するタイプは次のとおりです。

- **[0]**：オプションなし
- **[1]**：箱ひげ図で異常値を表示しない
- **[2]**：箱ひげ図で異常値を表示する

例：

`H3:={"D1", "", 2, 1, #FF:24h}`は、データ リストとして D1 を使用し、頻度は指定せず、異常値抜きの箱ひげ図を青色で描画するように H3 を定義します。

Method

推論

仮説検証結果または信頼区間を計算する推論アプリケーションが設定されているかどうかを指定します。[記号表示]では、Method の選択を行います。

プログラムでは、以下のように入力します。

`0 ▶ Method`：仮説検証の場合

`2 ▶ Method`：信頼区間の場合

`3 ▶ Method`：カイ 2 乗の場合

`4 ▶ Method`：回帰の場合

R0...R9

極座標

θ に式を格納します。[記号表示]では、R0 から R9 の 1 つを選択して、式を入力します。

プログラムでは、以下のように入力します。

```
SIN(θ) ▶ R1
```

S1...S5

2 変数統計

[2 変数統計]アプリケーションの変数は S1 ~ S5 です。これらの変数には 2 変数統計分析を定義するためのデータが含まれています。S1 は、S1 を定義するオブジェクトのリストを返します。各リストには、順番に次の項目が含まれています。

- 独立変数データ リストを定義する、一重引用符で囲まれた式（または空の二重引用符）
- 従属変数データ リストを定義する、一重引用符で囲まれた式（または空の二重引用符）
- オプションで従属変数のデータ リストの頻度を定義する、文字列または式
- フィット タイプ番号
- フィット式
- 散布図の色
- 散布図の標本マーク タイプ番号
- フィット プロットの色

フィット タイプ番号は 1 から 13 の整数で、変数 S1 から S5 のそれぞれで使用する統計プロット タイプを示します。各番号に対応するタイプは次のとおりです。

- **[1]**：1 次
- **[2]**：対数
- **[3]**：指数
- **[4]**：べき乗
- **[5]**：指数
- **[6]**：逆
- **[7]**：ロジスティック
- **[8]**：2 次
- **[9]**：3 次
- **[10]**：4 次
- **[11]**：三角
- **[12]**：中央値をつないだ線
- **[13]**:ユーザー定義

散布図の標本マーク タイプ番号は 1 から 9 の整数で、散布図の各点を表す図形を示します。各番号に対応するタイプは次のとおりです。

- **[1]**:小さい白抜きの丸
- **[2]**:小さい白抜きの四角形
- **[3]**：細い x
- **[4]**：白抜きの十字
- **[5]**：小さい白抜きのひし形
- **[6]**：太い x
- **[7]**：小さい塗りつぶしの丸
- **[8]**：細いひし形
- **[9]**：大きい白抜きの丸

例：

`S1:={"C1", "C2", "", 1, "", #FF:24h, 1, #FF:24h}`は、独立データとして C1 を使用し、従属データとして C2 を使用し、従属データの頻度は指定せず、線形フィットを描画し、線形フィットの方程式は指定せず、散布図の色は青でマークのタイプは 1、フィット プロットの色は青で描画するように設定します。

InfType

推論

仮説検証または信頼区間のタイプを指定します。変数 `Method` の値に依存します。[記号表示]では、`Type` の選択を行います。

またはプログラムでは、以下のリストからの整数定数を変数 Type に格納します。`Method=0` の場合、定数値とその意味は次のとおりです。

0 Z-Test：1 μ

1 Z-Test：$\mu_1 - \mu_2$

2 Z-Test：1 π

3 Z-Test：$\pi_1 - \pi_2$

4 T-Test：1 μ

5 T-Test：$\mu_1 - \mu_2$

`Method=1` の場合、定数値とその意味は次のとおりです。

0 Z-Int：1 μ

1 Z-Int：$\mu_1 - \mu_2$

2 Z-Int：1 π

3 Z-Int：$\pi_1 - \pi_2$

4 T-Int：1 μ

5 T-Int：$\mu_1 - \mu_2$

`Method=2` の場合、定数値とその意味は次のとおりです。

0 カイ 2 乗適合度検定

1 カイ 2 乗 2 要因検定

Method=3 の場合、定数値とその意味は次のとおりです。

0 線形 t 検定

1 区間：傾き

2 区間：切片

3 区間：平均応答

4 予測区間

X0, Y0...X9,Y9

パラメトリック

2 つの式を T: X(T) および Y(T) に格納します。[記号表示]では、X0-Y0 から X9-Y9 のどれかを選択し、式を T に入力します。

プログラムでは、Xn および Yn 内の T に式を格納します。ここで、n は 0 から 9 の整数です。

例：

```
SIN(4*T)▶ Y1;2*SIN(6*T)▶ X1
```

U0...U9

数列

N に式を格納します。[記号表示]では、U0 から U9 のどれかを選択し、N、Un(N-1)、または Un(N-2) に式を入力します。

プログラムでは、RECURSE コマンドを使用して U*n* に式を格納します。ここで、*n* は 0 から 9 の整数です。

例：

```
RECURSE (U,U(N-1)*N,1,2) ▶ U1
```

数値表示変数

C0...C9

2 変数統計

数値データのリストを格納します。[数値表示]では、C0 から C9 に数値データを入力します。

プログラムでは、以下のように入力します。

```
 LIST ▶ Cn
```

ここで、n = 0, 1, 2, 3 ... 9、LIST はリストまたはリストの名前です。

D0...D9

1 変数統計

数値データのリストを格納します。[数値表示]では、D0 から D9 に数値データを入力します。

プログラムでは、以下のように入力します。

```
LIST ▶ Dn
```

ここで、n = 0, 1, 2, 3 ... 9、LIST はリストまたはリストの名前です。

NumIndep

関数、パラメトリック、極座標、数列、高度なグラフ作成

独立値のリスト（または、独立値の 2 値セット）を[独自の表をビルド]で使用するように指定します。[数値表示]で 1 つずつ値を入力します。

プログラムでは、以下のように入力します。

```
LIST ▶ NumIndep
```

List には、リスト自身またはリストの名前を指定できます。高度なグラフ作成アプリケーションの場合、リストは数字のリストではなく、ペア（2 要素ベクトル）のリストです。

NumStart

関数、パラメトリック、極座標、数列

[数値表示]で表の開始値を設定します。

[数値のセットアップ]表示から、NUMSTART の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ NumStart
```

NumXStart

高度なグラフ作成

[数値表示]内の表で x 値の開始値を設定します。

[数値のセットアップ]表示から、NUMXSTART の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ NumXStart
```

NumYStart

高度なグラフ作成

[数値表示]内の表で Y 値の開始値を設定します。

[数値のセットアップ]表示から、NUMYSTART の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ NumYStart
```

NumStep

関数、パラメトリック、極座標、数列

[数値表示]で独立変数のステップ サイズ（増分値）を設定します。

[数値のセットアップ]表示から、NUMSTEP の値を入力します。

プログラムでは、以下のように入力します。

n ▶ NumStep

ここで、n > 0

NumXStep

高度なグラフ作成

[数値表示]で独立 X 変数のステップ サイズ（増分値）を設定します。

[数値のセットアップ]表示から、NUMXSTEP の値を入力します。

プログラムでは、以下のように入力します。

n ▶ NumXStep

ここで、n > 0

NumYStep

高度なグラフ作成

[数値表示]で独立 Y 変数のステップ サイズ（増分値）を設定します。

[数値のセットアップ]表示から、NUMYSTEP の値を入力します。

プログラムでは、以下のように入力します。

n ▶ NumYStep

ここで、n > 0

NumType

関数、Parametric（パラメトリック）、極座標、数列、高度なグラフ作成

表の形式を設定します。

[数値のセットアップ]表示では、NumType の選択を行います。

プログラムでは、以下のように入力します。

0 ▶ NumType：自動（初期設定）。

1 ▶ NumType：BuildYourOwn。

NumZoom

関数、パラメトリック、極座標、数列

[数値表示]でズーム倍率を設定します。

[数値のセットアップ]表示から、NUMZOOM の値を入力します。

プログラムでは、以下のように入力します。

n ▶ NumZoom

ここで、n > 0

NumXZoom

高度なグラフ作成

[数値のセットアップ]表示から、NUMXZOOM の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ NumXZoom
```

ここで、n > 0

NumYZoom

高度なグラフ作成

[数値表示]で Y 列の値のズーム倍率を設定します。

[数値のセットアップ]表示から、NUMYZOOM の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ NumYZoom
```

ここで、n > 0

推論アプリケーション変数

以下の変数は推論アプリケーションで使用されます。推論アプリケーションの[数値表示]内のフィールドに対応します。この表示に表示される変数セットは、[記号表示]で選択された仮説検証または信頼区間に依存します。

Alpha

仮説検証の α 水準を設定します。[数値表示]から、Alpha の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ Alpha
```

ここで、n > 0

Conf

信頼区間の信頼水準を設定します。[数値表示]から、C の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ Conf
```

ここで、n > 0

ExpList

カイ 2 乗適合度検定で期待されるカウントをカテゴリ別に格納します。[記号表示]のフィールド Expected で、Count を選択します。次に、[数値表示]で、ExpList にデータを入力します。

$Mean_1$

1 平均仮説検証または信頼区間の標本の平均値を設定します。2 平均検証または信頼区間について、1 番目の標本の平均値を設定します。[数値表示]から、xまたは $\bar{x}_1$ の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ Mean1
```

Mean2

2 平均検証または信頼区間について、2 番目の標本の平均値を設定します。[数値表示]から、$\bar{x}_2$ の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ Mean2
```

μ0

仮説検証の母集団平均の想定値を設定します。[数値表示]から、μ_0 の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ μ0
```

ここで、$0 < \mu_0 < 1$

n1

仮説検証または信頼区間の標本のサイズを設定します。2 つの平均または 2 つの母集団の違いを含む検証または区間について、1 番目の標本のサイズを設定します。[数値表示]から、n_1 の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ n1
```

n2

2 つの平均または 2 つの母集団の違いを含む検証または区間について、2 番目の標本のサイズを設定します。[数値表示]から、n_2 の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ n2
```

ObsList

カイ 2 乗適合度検定で求められたカウント データを格納します。[数値表示]で、`ObsList` にデータを入力します。

ObsMat

カイ 2 乗 2 要因検定で求められたカウントをカテゴリ別に格納します。[数値表示]で、`ObsMat` にデータを入力します。

π0

1 比率の Z 検定で成功が予測される比率を設定します。[数値表示]から、π_0 の値を設定します。

プログラムでは、以下のように入力します。

```
n ▶ π0
```

ここで、$0 < \pi_0 < 1$

Pooled

スチューデントの 2 つの平均を含む T 分布を使用した検定または区間で、標本がプールされているかどうかを指定します。[数値表示]から、`Pooled` の値を設定します。

プログラムでは、以下のように入力します。

`0` ▶ `Pooled`：プールされない（初期設定）。

`1` ▶ `Pooled`：プールされる。

ProbList

カイ 2 乗適合度検定で期待される確率をカテゴリ別に格納します。[記号表示]の[期待]ボックスで、`Probability` を選択します。次に、[数値表示]で、`ProbList` にデータを入力します。

s_1

仮説検証または信頼区間の標本標準偏差を設定します。2 つの平均または 2 つの母集団の違いを含む検証または区間について、1 番目の標本の標本標準偏差を設定します。[数値表示]から、s_1 の値を設定します。

プログラムでは、以下のように入力します。

`n` ▶ s_1

s_2

2 つの平均または 2 つの母集団の違いを含む検証または区間について、2 番目の標本の標本標準偏差を設定します。[数値表示]から、s_2 の値を設定します。

プログラムでは、以下のように入力します。

`n` ▶ s_2

σ_1

仮説検証または信頼区間の母標準偏差を設定します。2 つの平均または 2 つの母集団の違いを含む検証または区間について、1 番目の標本の母標準偏差を設定します。[数値表示]から、σ_1 の値を設定します。

プログラムでは、以下のように入力します。

`n` ▶ σ_1

σ_2

2 つの平均または 2 つの母集団の違いを含む検証または区間について、2 番目の標本の母標準偏差を設定します。[数値表示]から、σ_2 の値を設定します。

プログラムでは、以下のように入力します。

`n` ▶ σ_2

x_1

1 母集団仮説検証または信頼区間の成功数を設定します。2 つの母集団の違いを含む検証または区間について、1 番目の標本の成功数を設定します。[数値表示]から、x_1 の値を設定します。

プログラムでは、以下のように入力します。

`n` ▶ x_1

x_2

2 つの母集団の違いを含む検証または区間について、2 番目の標本の成功数を設定します。[数値表示]から、x_2 の値を設定します。

プログラムでは、以下のように入力します。

`n` ▶ x_2

Xlist

回帰検定および区間の説明データ（X）のリストを格納します。[数値表示]で、`Xlist` にデータを入力します。

Xval

平均応答の信頼区間および将来の応答の予測区間について、調査対象の説明変数（X）の値を格納します。ウィザードで求められたときに値を入力します。

Ylist

回帰検定および区間の応答データ（Y）のリストを格納します。[数値表示]で、`Ylist` にデータを入力します。

金融アプリケーション変数

以下の変数は金融アプリケーションで使用されます。金融アプリケーションの[数値表示]のフィールドに対応します。

CPYR

1 年あたりの複利計算期間です。キャッシュ フロー計算の 1 年あたりの複利計算期間の数を設定します。金融アプリケーションの[数値表示]から、`C/YR` の値を入力します。

プログラムでは、以下のように入力します。

`n` ▶ `CPYR`

ここで、n > 0

BEG

複利期間の期首または期末のどちらで利息が複利計算されるかを指定します。金融アプリケーションの[数値表示]から、`End` を選択またはクリアします。

プログラムでは、以下のように入力します。

`1` ▶ `BEG`：期末に複利計算する（初期設定）。

`0` ▶ `BEG`：期首に複利計算する。

FV

将来価値です。投資の将来価値を設定します。金融アプリケーションの[数値表示]から、FV の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ FV
```

正の値は投資またはローンのリターンを表します。

IPYR

1 年あたりの利益です。キャッシュ フローの年間利率を設定します。金融アプリケーションの[数値表示]から、I%YR の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ IPYR
```

ここで、n > 0

NbPmt

支払いの回数です。キャッシュ フローの支払回数を設定します。金融アプリケーションの[数値表示]から、N の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ NbPmt
```

ここで、n > 0

PMT

支払い額です。キャッシュ フローの各支払額を設定します。金融アプリケーションの[数値表示]から、PMT の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ PMT
```

支払い額は、支払を行っている場合は負、受け取っている場合は正です。

PPYR

年間の支払回数です。キャッシュ フロー計算の年間の支払回数を設定します。金融アプリケーションの[数値表示]から、P/YR の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ PPYR
```

ここで、n > 0

PV

現在価値です。投資の現在価値を設定します。財務アプリケーションの[数値表示]から、PV の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ PV
```

注記：負の値は投資またはローンを表します。

GSize

グループ サイズです。償却表の各グループのサイズを設定します。金融アプリケーションの[数値表示]から、`Group Size` の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ GSize
```

線形ソルバーアプリケーション変数

以下の変数は線形ソルバーアプリケーションで使用されます。アプリケーションの[数値表示]のフィールドに対応します。

LSystem

2x2 または 3x3 の連立一次方程式を表す 2x3 または 3x4 行列を格納します。線形ソルバーアプリケーションの[数値表示]から、連立一次方程式の係数および定数を入力します。

プログラムでは、以下のように入力します。

```
matrix▶LSystem
```

matrix は行列、または行列変数 M0 から M9 のうちの 1 つの名前です。

三角ソルバーアプリケーション変数

以下の変数は三角ソルバーアプリケーションで使用されます。アプリケーションの[数値表示]のフィールドに対応します。

SideA

辺 a の長さです。角度 A の反対側の辺の長さを設定します。三角ソルバーの[数値表示]から、a の正の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ SideA
```

ここで、n > 0

SideB

辺 b の長さです。角度 B の反対側の辺の長さを設定します。三角ソルバーの[数値表示]から、b の正の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ SideB
```

ここで、n > 0

SideC

辺 c の長さです。角度 C の反対側の辺の長さを設定します。三角ソルバーの[数値表示]から、c の正の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ SideC
```

ここで、n > 0

AngleA

角度 A の単位です。角度 A の単位を設定します。この変数の値は角度モード設定に応じて解釈されます（度またはラジアン）。三角ソルバーの[数値表示]から、角度 A の正の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ AngleA
```

ここで、n > 0

AngleB

角度 B の単位です。角度 B の単位を設定します。この変数の値は角度モード設定に応じて解釈されます（度またはラジアン）。三角ソルバーの[数値表示]から、角度 B の正の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ AngleB
```

ここで、n > 0

AngleC

角度 C の単位です。角度 C の単位を設定します。この変数の値は角度モード設定に応じて解釈されます（度またはラジアン）。三角ソルバーの[数値表示]から、角度 C の正の値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ AngleC
```

ここで、n > 0

TriType

三角ソルバーアプリケーションの[数値表示]内の の状態に対応します。一般的な三角ソルバーまたは直角三角ソルバーのどちらが使用されるかを決定します。[三角ソルバー]表示から、 をタップします。

プログラムでは、以下のように入力します。

```
0 ▶ TriType：一般的な三角ソルバー
1 ▶ TriType：直角三角ソルバー
```

[ホーム設定]変数

以下の変数（Ans を除く）はホーム設定にあります。最初の 4 つはすべて、アプリケーションの[記号のセットアップ]表示で上書きできます。

Ans

ホーム表示または CAS 表示で計算された最終結果を格納します。Ans(n) は、ホーム表示の履歴の n 番目の結果を返します。CAS 表示で Ans が行列の場合には、Ans(m,n) は行 m、列 n 内の要素を返します。

HAngle

ホーム表示の角度形式を設定します。[ホーム設定]で、角度単位に[度]または[ラジアン]を選択します。

プログラムでは、以下のように入力します。

0 ▶ HAngle：ラジアン

1 ▶ HAngle：度

2 ▶ HAngle：グラード

HDigits

ホーム表示で[標準]以外の数値形式の桁数を設定します。[ホーム設定]で、**[数値形式]**の 2 番目のフォールドに値を入力します。

プログラムでは、以下のように入力します。

n ▶ HDigits、ただし、0 < n < 11。

HFormat

ホーム表示で使用する数値表示形式を設定します。[ホーム設定]の**[番号書式]**フィールドで、[標準]、[固定]、[関数]、または[工学]を選択します。

プログラムでは、以下の定数番号（またはその名前）のいずれかを変数 HFormat に格納します。

0 標準

1 固定

2 関数

3 工学

HComplex

実数入力からの複素数結果を許可します。たとえば、HComplex が 0 に設定された場合、ASIN(2)はエラーを返します。HComplex が 1 に設定された場合、ASIN(2)は 1.57079632679 – 1.31695789692*i を返します。

ホーム表示で、**[複素数]**フィールドを選択またはクリアします。プログラムでは、以下のように入力します。

0 ▶ HComplex：オフ。

1 ▶ HComplex：オン。

Date

システム日付を格納します。形式は YYYY.MMDD です。この形式は、[ホーム設定]画面で設定された形式に関係なく使用されます。[ホーム設定]のページ 2 で、Date の値を入力します。

プログラムでは、以下のように入力します。

YYYY.MMDD ▶ Date。YYYY は 4 桁の年、MM は 2 桁の月、DD は 2 桁の日です。

時間

現在の時計の時刻を DMS 形式で返します。これは TICKS 変数に似ています。TICKS 変数には、コンピューターが起動してからのミリ秒単位の時間が格納されています。

時計の時刻を設定するには、Time:= H°MM’ SS’’ と入力します。

Language

システム言語を示す整数を格納します。[ホーム設定]から、**[言語]**フィールドの言語を選択します。

プログラムでは、以下の定数番号のいずれかを変数 Language に格納します。

1 ▶ Language（英語）

2 ▶ Language（中国語）

3 ▶ Language（フランス語）

4 ▶ Language（フランス語）

5 ▶ Language（スペイン語）

6 ▶ Language（オランダ語）

7 ▶ Language（ポルトガル語）

Entry

入力モードを示す整数を格納します。[ホーム設定]で、**[エントリ]**のオプションを選択します。

プログラムでは、以下のように入力します。

0 ▶ Entry：テキストブック

1 ▶ Entry：代数

2 ▶ Entry：RPN

整数

Base

整数ベースを返すか設定します。[ホーム設定]で、**[整数]**の横の最初のフィールドのオプションを選択します。プログラムでは、以下のように入力します。

0 ▶ Base：2 進数

1 ▶ Base：8 進数

2 ▶ Base：10 進数

3 ▶ Base：16 進数

Bits

整数を表すためのビット数を返すか設定します。[ホーム設定]で、**[整数]**の横の 2 番目のフィールドの値を入力します。プログラムでは、以下のように入力します。

n ▶ Bits。n はビット数です。

Signed

整数ワードサイズが符号付きかどうかを示すフラグの状態を返すか、フラグを設定します。[ホーム設定]で、**[整数]**の右の[±]フィールドをチェックまたはチェック解除します。プログラムでは、以下のように入力します。

0 ▶ `Signed`：符号なし

1 ▶ `Signed`：符号付き

その他の一般的なホーム変数

[ホーム設定]を制御するホーム変数のほかに、さまざまなタイプのホーム オブジェクトにユーザーがプログラムによってアクセスできるようにする 4 つの追加のホーム変数があります。

DelHVars

`DelHVars(n)` または `DelHVars("name")` は、指定されたホーム ユーザー変数を削除します。

HVars

ユーザー定義のホーム変数にアクセスできます。

`HVars` は、すべての定義済みのホーム ユーザー変数の名前のリストを返します。

`HVars(n)` は、n 番目のユーザー定義ホーム変数を返します。

`HVars("name")` は、与えられた名前のユーザー定義ホーム変数を返します。

`HVars(n` または`"name", 2)` は、変数がユーザー定義関数の場合はその関数のパラメーターのリストを返し、それ以外の場合は 0 を返します。

`HVars(n):=value` は、値を n 番目のホーム ユーザー変数に保存します。

`HVars("name"):=value` は、「name」というホーム ユーザー変数に値を保存します。そのような変数が存在しない場合、変数を作成します。

`HVars(n or "name", 2):= {"Param1Name", ..., "ParamNName"}` は、指定されたユーザー変数に関数が保存されていると仮定し、その関数のパラメーターが何であるかを指定します。

注記

注記変数では、電卓に保存された注記にアクセスできます。

`Notes` は、電卓に保存されたすべての注記の名前のリストを返します。

`Notes(n)` は、電卓の n 番目の注記の内容を返します（1 から NbNotes）。

`Notes("name")` は、タイトルが「name」である注記の内容を返します。

このコマンドは、注記の定義、再定義、およびクリアにも使用できます。

`Notes(n):="string"` は、注記 n の値を設定します。文字列が空の場合、この注記は削除されます。

`Notes("name"):="string"` は、「name」という注記の値を設定します。文字列が空白の場合、注記は削除されます。「name」という名前の注記が存在しない場合、この文字列の内容で注記が作成されます。

Programs（プログラム）

Programs 変数では、電卓に保存されたプログラムへのアクセスができます。

Programs は、電卓に保存されたすべてのプログラムの名前のリストを返します。

Programs(n) は、電卓の n 番目のプログラムの内容を返します（1 から NbPrograms）。

Programs(n):="string"は、プログラム n のプログラム ソース コードを設定します。文字列が空白の場合、プログラムは削除されます。

Programs("name") は、「name」というプログラムのソースを返します。

Programs("name"):="string"は、「name」というプログラムのソース コードを文字列に設定します。文字列が空白の場合、プログラムは削除されます。「name」という名前のプログラムが存在しない場合、プログラムが作成されます。

TOff

TOff には電卓の次の自動電源オフまでのミリ秒数を定義する整数が格納されます。初期設定は 5 分で、#493E0h（5*60*1000 ミリ秒）です。

有効な値の範囲は#1388h から#3FFFFFFFh までです。

[記号のセットアップ]変数

以下の変数は、アプリケーションの[記号のセットアップ]にあります。ホーム設定で対応する変数の値を上書きするために使用できます。

AAngle

角度モードを設定します。

[記号のセットアップ]から、角度単位に[システム]、[度]、または[ラジアン]を選択します。[システム]（初期設定）を選択した場合は、強制的にホーム設定と同じ角度単位に設定されます。

プログラムでは、以下のように入力します。

0 ▶ AAngle：システム（初期設定）

1 ▶ AAngle：ラジアン

2 ▶ AAngle：度

3 ▶ AAngle：グラード

AComplex

複素数モードを設定します。

[記号のセットアップ]から、[システム]、[オン]、またはオフを選択します。システム（初期設定）を選択した場合は、強制的にホーム設定の対応する設定と同じモードに設定されます。

プログラムでは、以下のように入力します。

0 ▶ AComplex：システム（初期設定）

1 ▶ AComplex：オン

2 ▶ AComplex：オフ

ADigits

アプリケーションの[記号のセットアップ]の固定、関数、または工学の数値形式に使用する小数点以下桁数を格納します。

[記号のセットアップ]で、[番号書式]の 2 番目のフォールドに値を入力します。

プログラムでは、以下のように入力します。

```
n ▶ ADigits
```

ここで、0 < n <11

AFormat

ホーム表示の数値表示およびプロット表示のラベル軸に使用する数値表示形式を定義します。

[記号のセットアップ]の**[番号書式]**フィールドで、[標準]、[固定]、[関数]、または[工学]を選択します。

プログラムでは、以下の定数番号を変数 `AFormat` に格納します。

0 システム

1 標準

2 固定

3 関数

4 工学

例：

```
3 ▶ AFormat
```

結果変数

関数、1 変数統計、2 変数統計、および推論アプリケーションには、これらのアプリケーションの外部（プログラム内など）で再利用できる結果を生成する関数が用意されています。たとえば、関数アプリケーションは関数の根を求めることができ、その根は Root という変数に書き込まれます。その変数は他の場所で使用できます。

結果変数は、それらを生成するアプリケーションと一緒にリストされます。

29 基本整数算術計算

現在の数学で使用される一般的な基数は基数 10 です。初期設定では、HP Prime が行うすべての計算は基数 10 で実行され、すべての結果は基数 10 で表示されます。

ただし、HP Prime での整数算術計算は 4 つの基数で実行できます：10 進数（基数 10）、2 進数（基数 2）、8 進数（基数 8）、および 16 進数（基数 16）。たとえば、基数 16 の 4 と基数 8 の 71 の積は基数 16 の E4 です。これは、基数 10 で 4 と 57 の積が 228 になるのと同じです。

整数算術計算を実行しようとしていることを示すには、数値の前にポンド記号（#、ALPHA alpha 3 π #を押すことで取得される）を付けます。数値に使用する基数を示すには、該当する基数マーカーを付加します。

基数マーカー	基数
[空]	初期設定の基数を使用します（632 ページの初期設定基数を参照）。
d	10 進数
b	2 進数
o	8 進数
h	16 進数

したがって、#11b は 3_{10} を表します。基数マーカー *b* は数値が 2 進数として解釈されることを示します：11_{2}。同様に、#E4h は 228_{10} を表します。この場合、基数マーカー *h* は数値が 16 進数として解釈されることを示します：$E4_{16}$。

整数算術計算では、浮動小数点算術計算で剰余を返す計算の結果は切り捨てられます：整数部だけが提示されます。したがって、#100b/#10b の場合は正確な答えが得られます：#10b（$4_{10}/2_{10}$ は 2_{10} です）。ただし、#100b/#11b の場合は正しい結果の整数部分だけが返されます：#1b。

整数算術計算の精度は整数ワードサイズで制限できます。ワードサイズは、整数を表現できる最大ビット数です。1 から 64 の間の値に設定できます。ワードサイズが小さいほど、正確に表現できる整

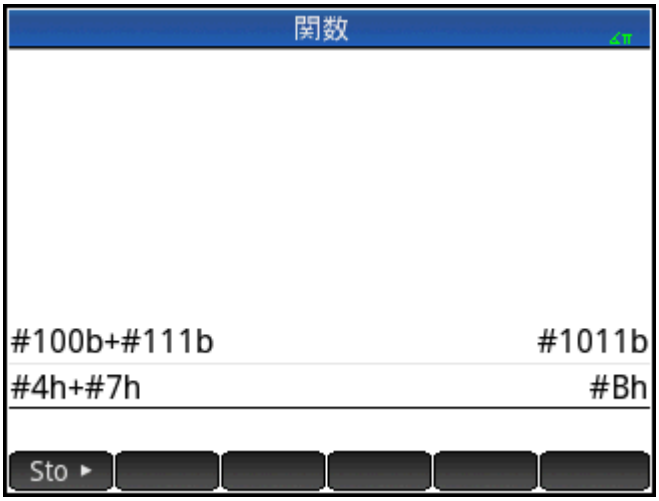
数は小さくなります。初期設定ワードサイズは 32 で、最大で約 2×10^9 までの整数を表現するのには十分です。ただし、それより大きな整数は切り捨てられます。つまり、MSB（最上位ビット）が切り捨てられます。このため、このような数値を含む計算の結果は正確でなくなります。

初期設定基数

初期設定基数を設定した場合、整数算術計算で使用する数値の入力と表示にのみ影響します。初期設定基数を 2 進数に設定した場合、27 および 44 はホーム表示でそのまま表現され、これらの数値を加算した結果も 71 と表現されます。ただし、#27b と入力した場合は、2 と 7 は 2 進数算術計算にない整数であるため、構文エラーになります。27 は#11011b と入力する必要があります（$27_{10}=11011_2$）。

初期設定基数を設定することは、整数算術計算を行うときに数値に常に基数マーカーを指定する必要がなくなることを意味します。初期設定でない基数の数値を含める場合は例外です。基数マーカーを含める必要があります。すなわち、初期設定基数が 2 で、整数算術計算で 27 を入力する場合は、*b* サフィックスなしで#11011 と入力するだけです。しかし、$E4_{16}$ を入力したい場合、サフィックスを付けて入力する必要があります：#E4h。（HP Prime では、計算が履歴に表示されるときに、省略された基数マーカーが追加されます。）

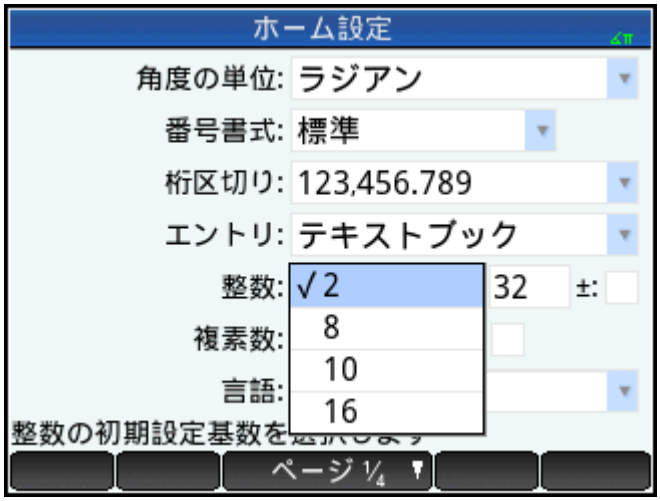
初期設定基数を変更した場合、*基数マーカーを明示的に追加しなかった*整数算術計算を含む計算は、履歴内で新しい基数で再表示されます。以下の図で、1 番目の計算には明示的に基数マーカーが含まれていました（各オペランドに *b* ）。2 番目の計算は 1 番目のコピーでしたが基数マーカーなしです。そのとき、初期設定基数は 16 進数に変更されていました。1 番目の計算はそのままですが、2 番目の計算は、基数マーカーがオペランドに明示的に追加されていないため、基数 16 で再表示されました。



初期設定基数の変更

整数算術計算のための電卓の初期設定基数は 16（16 進数）です。初期設定基数を変更するには、以下の操作を行います。

1. **[ホーム設定]**画面を表示します： Shift Settings



2. **[整数]**メニューから目的の基数を選択します：**[2 進数]**、**[8 進数]**、**[10 進数]**、または**[16 進数]**。
3. [整数]の右側のフィールドはワードサイズ フィールドです。これは、整数を表現できる最大ビット数です。初期設定値は 32 ですが、1 から 64 までの値に変更できます。
4. 符号付き整数を許可する場合は、ワードサイズ フィールドの右側にある**[±]**オプションを選択します。このオプションを選択すると、整数の最大サイズがワードサイズより 1 ビット少なくなります。

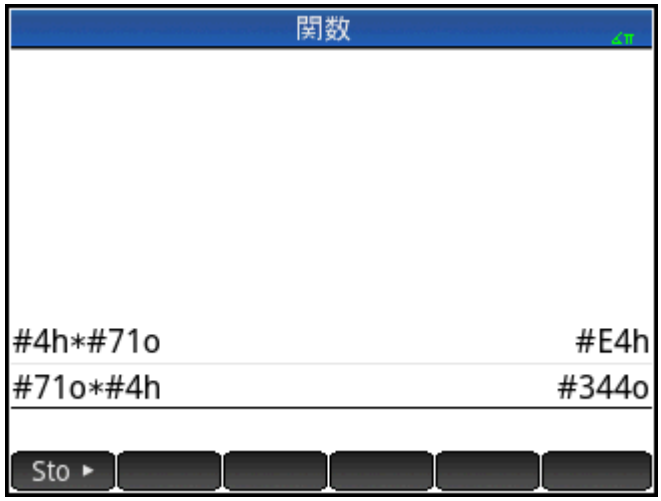
整数算術計算の例

整数算術計算のオペランドは、同じ基数を使用することも、基数を混在させることもできます。

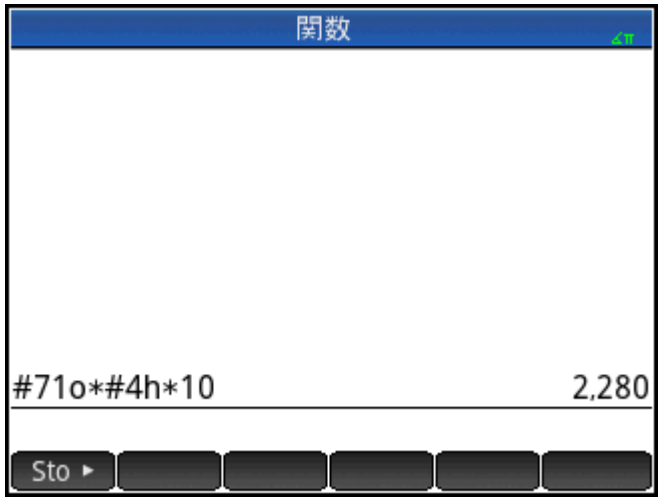
整数計算	相当する 10 進数
#10000b+#10100b =#100100b	16 + 20 = 36
#71o – #10100b = #45o	57 – 20 = 37
#4Dh * #11101b = #8B9h	77 × 29 = 2233
#32Ah/#5o = #A2h	810/5 = 162

基数が混在する算術計算

例外が 1 つあります。オペランドに基数が異なるものがある場合、計算結果は 1 番目のオペランドの基数で表現されます。次の図は 2 つの同等の計算を示しています：最初は 4_{10} と 57_{10} の積、2 番目は 57_{10} と 4_{10} の積です。結果が数学的に同等であることも明らかです。ただし、それぞれは、最初に入力されたオペランドの基数で表現されます。最初の場合は 16 で、2 番目の場合は 8 です。



例外は、整数の前に#が付いていないオペランドの場合です。この場合、結果は基数 10 で表現されます。



整数操作

整数算術計算の結果は、**[整数の編集]**ダイアログに表示することで、さらに分析して操作できます。

1. ホーム表示で、カーソル キーを使用して目的の結果を選択します。
2. Shift −（基数）を押します。

   [整数の編集]ダイアログが表示されます。上部の**[Was]**フィールドには、ホーム表示で選択した結果が表示されます。

   16 進数および 10 進数の方程式は、**[Out]**フィールドの下に表示され、整数のビット単位表現が後に続きます。

   ビット表現の下の記号は、整数を編集するために押せるキーを示します。（ホーム表示の計算結果は変更されません。）キーは以下のとおりです。

- または （シフト）：ビットを左（または右）へ 1 スペース分シフトします。押すたびに、表現される新しい整数は**[Out]**フィールド（およびその下の[hex]および[decimal]フィールド）に表示されます。
- または （ビット）：ワードサイズを増やします（減らします）。新しいワードサイズは**[Out]**フィールドに表示される値に付加されます。
- （否定）：2 の補数を返します（指定されたワードサイズ内の各ビットが反転され、1 が加算されます）。表現される新しい整数は**[Out]**フィールド（およびその下の[hex]および[decimal]フィールド）に表示されます。
- または （基数の循環）：整数を別の基数で**[Out]**フィールドに表示します。

メニュー ボタンにはいくつかの追加オプションがあります。

リセット：すべての変更を元の状態に戻します

ベース：基数を切り替えます。+を押すのと同じです

符号付：符号付きと符号なしの間でワードサイズを切り替えます

否定：1 の補数を返します（指定されたワードサイズ内の各ビットが反転されます）：0 は 1、1 は 0 に置き換えられます。表現される新しい整数は[Out]フィールド（およびその下の[hex]および[decimal]フィールド）に表示されます。

編集：編集モードを有効にします。カーソルが表示され、カーソル キーを使用してダイアログ上を移動できます。ビット表現と同様に、[hex]および[decimal]フィールドを変更できます。1 方のフィールドを変更すると、他方のフィールドが自動的に変更されます。

OK：ダイアログを閉じて、変更を保存します。変更を保存しない場合は、代わりに Esc Clear を押します。

3. 必要な変更を行います。
4. 変更を保存するには、 OK をタップします。そうでない場合は、 Esc Clear を押します。

注記：変更を保存した場合、次にホーム表示で同じ結果を選択すると**[整数の編集]**ダイアログが開き、**[Was]**フィールドに表示される値は保存した値で、結果の値ではありません。

基底関数

整数算術計算に関連する多くの関数は、ホーム表示およびプログラム内から起動できます。

BITAND	BITNOT	BITOR
BITSL	BITSR	BITXOR
B→R	GETBASE	GETBITS
R→B	SETBASE	SETBITS

595 ページの整数を参照してください。

30 付録 A：用語集

アプリケーション

小さなアプリケーション。1 つ以上の関連するトピックを調査するため、または特定の種類の問題を解決するために設計されます。組み込みアプリケーションは、Function（関数）、Advanced Graphing（高度なグラフ作成）、Geometry（幾何）、Spreadsheet（スプレッドシート）、Statistics 1Var（1 変数統計）、Statistics 2Var（2 変数統計）、Inference（推論）、DataStreamer（データストリーマー）、Solve（解く）、Linear Solver（線形ソルバー）、Triangle Solver（三角ソルバー）、Finance（金融）、Parametric（パラメトリック）、Polar（極座標）、Sequence（数列）、Linear Explorer（1 次エクスプローラー）、Quadratic Explorer（2 次エクスプローラー）、および Trig Explorer（三角法エクスプローラー）です。アプリケーションには、特定の問題のデータおよび解を設定できます。アプリケーションは再利用可能で（プログラムに似ていますが、使いやすい）、すべての設定と定義を記録します。

ボタン

画面の下に表示されるオプションまたはメニュー。タッチすることで起動します。**キー**と比較してください。

CAS

数式処理システム。CAS を使用して正確な計算または記号計算を実行します。ホーム表示で行われる計算（数値近似を返すことが多い）と比較してください。CAS とホーム表示の間（およびその逆）で結果と変数を共有できます。

カタログ

項目（行列、リスト、プログラムなど）のコレクション。作成する新しい項目はカタログに保存され、カタログから特定の項目を選択して操作します。アプリケーションをリストする特別なカタログは、アプリケーション ライブラリと呼ばれます。

コマンド

プログラムで使用する操作。コマンドは変数に結果を保存できますが、結果を表示できません。

式

値を生成する数値、変数、または代数式（数値と関数）。

関数

結果を返す演算（引数を取る場合もある）。変数に結果を保存しません。引数は、かっこで囲み、コンマで区切る必要があります。

ホーム表示

電卓の基本開始点。ほとんどの計算はホーム表示で行うことができます。ただし、このような計算は数値近似を返すだけです。正確な結果が必要な場合は、CAS を使用できます。CAS とホーム表示の間（およびその逆）で結果と変数を共有できます。

入力フォーム

値を設定したりオプションを選択したりできる画面。ダイアログ ボックスとも呼ばれます。

キー

キーパッド上のキー（ボタンと比較してください。キーは画面に表示され、有効にするにはタップする必要があります）。

ライブラリ

項目（具体的にはアプリケーション）のコレクション。**カタログ**も参照してください。

リスト

コンマで区切られ、中括弧で囲まれたオブジェクトのセット。リストは通常、統計データを格納したり、関数を複数値で評価したりするために使用されます。リストは、List（リスト）エディターで作成および操作したり、List（リスト）カタログに保存したりできます。

行列

角括弧で囲まれた、実数または複素数の 2 次元配列。行列は、Matrix（行列）エディターで作成および操作したり、Matrix（行列）カタログに保存したりできます。ベクトルも Matrix（行列）カタログおよび Matrix（行列）エディターで処理されます。

メニュー

画面に表示される一連のオプション。画面の下に、リストとして、または一連のタッチ ボタンとして表示できます。

注記

Note（注記）エディターで記述するテキスト。一般的な注記、単独の注記、アプリケーションに固有の注記があります。

開放文

開放文は、=や<などの関係演算子で区切られた 2 つの式（代数または算術計算）で構成されます。開放文の例は $\mathbf{y^2<x^{-1}}$ や $\mathbf{x^2-y^2=3+x}$ です。

プログラム

プログラム エディターを使用して記録する、再利用可能な命令セット。

変数

オブジェクト（数値、リスト、行列、グラフィックなど）に与えられる名前。後で取得するのに役立ちます。 Sto ▸ コマンドは変数を割り当てます。変数メニュー（ 変数 ）から関連付けられた変数を選択することで、オブジェクトを取得できます。

ベクトル

一重角括弧で囲まれた、実数または複素数の 1 次元配列。ベクトルは、Matrix（行列）エディターで作成および操作したり、Matrix（行列）カタログに保存したりできます。

表示

HP アプリケーションの主要環境。アプリケーション表示の例には、[プロット]、[プロットのセットアップ]、[数値]、[数値のセットアップ]、[記号]、[記号のセットアップ]などがあります。

31 付録 B：トラブルシューティング

電卓が応答しない

電卓が応答しない場合、最初にリセットしてみることをお勧めします。これは、PC の再起動に似ています。一部の操作がキャンセルされ、一部の条件が復元され、一時メモリ区域がクリアされます。ただし、保存されたデータ（変数、アプリケーション、プログラムなど）はクリアしません。

リセット方法

電卓を裏返しにして、バッテリー収納部カバーのすぐ上にあるリセット穴にペーパー クリップを挿入します。電卓が再起動してホーム表示に戻ります。

電卓の電源がオンにならない場合

HP Prime がオンにならない場合、電卓がオンになるまで以下の手順を実行してください。手順を完了する前に、電卓がオンになることもあります。それでも電卓がオンにならない場合、カスタマー サポートに連絡して詳細情報を入手してください。

1. 少なくとも 1 時間電卓を充電します。
2. 1 時間の充電後に電卓をオンにします。
3. オンにならない場合は、前のセクションに従って電卓をリセットしてください。

動作限界

動作温度： 0º ~ 45ºC（32º ~ 113ºF）。

保存温度： – 20º ~ 65ºC（ – 4º ~ 149ºF）。

動作湿度 / 保存湿度： 最大 90％の相対湿度、40 °C（104 °F）。電卓が湿らないようにしてください。

バッテリは 3.7V、容量 1500mAh（5.55Wh）で動作します。

ステータス メッセージ

以下の表には、一般的なエラー メッセージとその意味を記載しています。アプリケーションや CAS によっては、読んですぐにわかる、固有のエラー メッセージが出力されることがあります。

メッセージ	意味
引数のタイプが間違っています	この演算の入力が正しくありません。
メモリ不足	演算を続けるには、いくらかのメモリを回復する必要があります。カスタマイズされた 1 つ以上のアプリケーション、行列、リスト、注記、またはプログラムを削除してください。
統計データ不足	計算の標本が十分ではありません。2 変数統計の場合、2 列のデータが必要であり、各列には少なくとも 4 つの数値が必要です。
無効な次元	配列引数の次元が間違っていました。

メッセージ	意味
統計データサイズが等しくありません	等しい数のデータ値を持つ 2 つの列が必要です。
構文エラー	入力した関数またはコマンドの引数または引数の順序が適切ではありません。区切り文字（かっこ、コンマ、ピリオド、およびセミコロン）も正しくなくてはなりません。索引で関数名を調べて、正しい構文を見つけてください。
関数が検査されていません	[プロット表示]を入力する前に、[記号表示]に方程式を入力して検査する必要があります。
受信エラー	別の電卓からのデータ受信の問題。データを再送してください。
未定義の名前	名前の付いたグローバル変数が存在しません。
メモリ不足	演算を続けるには、多くのメモリを回復する必要があります。カスタマイズされた 1 つ以上のアプリケーション、行列、リスト、注記、またはプログラムを削除してください。
2 つの小数点記号がある入力	入力した数値のいずれかに複数の小数点があります。
X/0	ゼロ除算のエラーです。
0/0	除算で未定義の結果です。
LN(0)	LN(0)は未定義です。
単位が一致しません	計算に互換性のない単位が含まれています（長さと質量の加算など）。

索引

記号/数字

[1 変数統計]アプリケーション
- 円グラフ 235
- 茎葉図 235
- コントロール図 234
- 数値表示 246
- [その他]メニュー 227, 246
- ドット図 234

1 変数統計アプリケーション 220
- 記号表示 223
- グラフの操作 236
- 計算される統計 229
- 数値表示 227
- 正規確率プロット 232
- 線プロット 232
- データの削除 228
- データの生成 229
- データの挿入 228
- データの並べ替え 229
- データの入力 226
- データのプロット 230
- データの編集 226, 228
- 箱ひげ図 231
- パレート図 233
- ヒストグラム 231
- プロット 230
- プロットタイプ 231
- [プロットのセットアップ]表示 236
- プロット表示 236
- 棒グラフ 233

2 変数統計アプリケーション 238
- 値の予測 244, 252
- 値の予測、プロット表示 252
- 値の予測、ホーム表示 252
- 回帰モデル 247
- 曲線のトレース 250
- グラフのプロット 243
- 計算される統計 248
- 散布図 250
- データセット 240
- データの入力 239, 245
- データのプロット 249
- データの編集 245
- 統計の操作 241
- トラブルシューティング 253
- トレースの順序 251
- 開く 238
- フィットタイプ 240, 247
- フィットの選択 247
- フィットの定義 248
- プロットのセットアップ 242
- [プロットのセットアップ]表示 252
- プロット表示 251
- 方程式の表示 243

60 進数 15

C

CAS 45
- 計算 46
- 設定 48
- 表示 45
- ホーム表示 50
- ホーム変数 50
- メニュー項目 49

CAS 設定 48
- ページ 1 48
- ページ 2 49

CAS 表示 4

F

Finance（金融）アプリケーション 313
- TVM 計算 317
- 貨幣の時間価値 316
- キャッシュ フロー ダイアグラム 315
- 償却 318
- 償却グラフ 320
- 償却の計算 318
- 償却の例 318

L

Linear Solver（線形ソルバー）アプリケーション 291
- 2x2 系 293
- 開く 291
- メニュー項目 293

P

Parametric（パラメトリック）アプリケーション 294
- Numeric view（数値表示） 298
- 角度の単位 295
- 関数の定義 295
- グラフの操作 297
- 開く 294
- プロットのセットアップ 296, 301

Polar（極座標）アプリケーション 299
- Numeric view（数値表示） 303
- 角度の単位 300
- 関数の定義 299
- グラフの操作 302
- 開く 299

R

RPN 37
- 履歴 38

S

Solve（解く）アプリケーション 283
- 1 つの方程式 283, 287
- 解の情報 289
- 既知の変数 285
- クリア 284
- シード値の入力 288
- 制限事項 289
- 解く 285, 289
- 開く 283, 287
- プロット 286
- 方程式の定義 284, 287

Spreadsheet（スプレッドシート）
アプリケーション
関数 219
形式パラメーター 218
ジェスチャ 209
セル参照 209
セルの名前設定 210
選択 209
データのインポート 212
ナビゲーション 209
変数の参照 215
ボタンおよびキー 216

T

Triangle Solver（三角ソルバー）アプリケーション 321
解が 1 つに決まらない場合 323
解なし 324
角度の単位 321
既知の値 322
三角形のタイプ 323
データが不十分 324
特殊な場合 323
開く 321
未知の値 322

あ

アプリケーション 58
1 変数統計 220
2 変数統計アプリケーション 238
Finance（金融） 313
Inference（推論） 254
Linear Solver（線形ソルバー） 291
Parametric（パラメトリック） 294
Polar（極座標） 299
Solve（解く） 283
Triangle Solver（三角ソルバー） 321
オプション 61
関数 102, 105
削除 60
作成 100
作成例 101
数列 304
注記の追加 100
並べ替え 60
開く 59
変数 102, 103
変数の修飾 104
リセット 60
アプリケーション ライブラリ 59
暗黙的な乗算 25

お

大きい結果 26
オン/オフ 3

か

[カタログ]メニュー 182, 197
affix 197
convexhull 198
distance2 198
division_point 198
equilateral_triangle 198
exbisector 199
extract_measure 199
harmonic_conjugate 199
harmonic_division 199
is_harmonic 200
is_harmonic_circle_bundle 200
is_harmonic_line_bundle 200
is_orthogonal 200
is_rectangle 200
is_rhombus 200
is_square 201
isobarycenter 199
LineHorz 201
LineVert 201
open_polygon 201
orthocenter 201
perpendicular_bisector 201
point2d 202
polar 202
pole 202
powerpc 202
radical_axis 202
vector 203
vertices 203
vertices_abca 203
カタログ メニュー
barycenter 197
かっこ 24
[関数]アプリ
グラフの変更 114
[関数]アプリケーション
スケッチ 113
積分 124, 126
導関数 124
関数アプリケーション 105, 107
2 次方程式 116
2 次方程式の傾き 118
2 次方程式の極値 120
2 つの関数間の面積 119
2 つの関数の交点 117
値への移動 112
演算 123
関数の分析 113
式の定義 106
[数値のセットアップ]表示 110
数値表示 109
[数値表示]の操作 111
ズーム オプション 112
スケールの変更 109
接線の追加 121
その他のオプション 113
トレース機能 107
表内の移動 111
開く 105
プロットのセットアップ 106
[プロット表示]メニュー 113
変数 121, 122

き

キー
EEX 15
計算 12
シフト 10
数学ショートカット 13
数学テンプレート 12
入力 9
編集 9
キーボード 7
幾何 142
幾何アプリケーション 142
グラフのプロット 142
計算の追加 147
準備 142
制限される点の追加 143
接線の追加 144

微分係数の点の作成　145
微分係数のトレース　149
開く　142
[プロット表示]での計算　149
幾何関数　182
幾何コマンド　182
[記号のセットアップ]表示　62
共通操作　73, 74
設定の変更　74
例　66
記号表示　61, 157
Δ三角形　186
⊿三角形　186
ODE　191
円　188
円錐　189
エントリの並べ替え　159
オブジェクトの削除　159
オブジェクトの作成　158
オブジェクトの非表示　159
外接円　188
回転　192
鏡映　192
拡大縮小　192
陰　191
関数　190
[記号のセットアップ]表示　160
軌跡　190
逆数変換　193
共通操作　69
極座標　190
曲線　188
交点　183, 184
勾配場　191
勾配場コマンド　156
[コマンド]メニュー　182
三角形　186
四辺形　186
垂線　185
垂直　185
数列　190
スライダー　191
正接　185
正多角形　187
正方形　188
線　184
線分　184
双曲線　189
相似　192
傍接円　188
楕円　189
多角形　186, 187
中央　183
中点　183
長方形　187
直線　184
点　183
点オン　183
投影　193
内接円　188
中線　185
二等分線　185
パラメトリック　190
半直線　184
反転　193
菱形　187
プロット　190
平行　184
平行移動　192
平行四辺形　186
変換　192
放物線　189
メニュー ボタン　72
リスト　191
例　65
輝度　4
逆ポーランド記法　37
キャンセル　3

く

クイック設定　6
クリップボード　26

け

計算　22
計算例　39
結果の再利用　39

こ

高度なグラフ作成アプリケーション　129
開放文　132
グラフの操作　133
[数値のセットアップ]表示　138
数値表示　136
[数値表示]の操作　137
[数値表示]の表示　137
選択した定義　133
トレース　135
トレース、[数値表示]　138
トレース、特定の点　140
トレース、端　139
開く　131
プロット ギャラリー　141
プロット ギャラリーの操作　141
プロットのセットアップ　132
コマンド
DROPN（以下をドロップ）　42
DUPN（以下を複製）　42
Echo（エコー）　42
PICK（ピック）　41
ROLL（ロール）　42
Stack（スタック）　42
Swap（スワップ）　42
項目の表示　43
全項目の削除　44
特定の項目の削除　43
→LIST（→リスト）　42
コンテキスト依存メニュー　8

し

式　23
再利用　26
試験モード　51
新しい構成　54
解除　56
基本モード　51
構成　56, 57
初期設定試験　52
有効化　55

す

[推論]アプリケーション
ANOVA　282
推論アプリケーション　254
1 標本 T 区間　273
1 標本 T 検定　268
1 標本 Z 区間　270
1 標本 Z 検定　265
1 比率の Z 区間　272
1 比率の Z 検定　266

1 変数統計アプリケーション 260
2 元表に基づく検定 276
2 標本 T 区間 274
2 標本 T 検定 269
2 標本 Z 区間 271
2 標本 Z 検定 265
2 比率の Z 区間 272
2 比率の Z 検定 267
カイ 2 乗検定 275
回帰の推論 276
仮説検証 264
傾きについての信頼区間 278
記号表示 255
グラフでの結果 264
検定結果の表示 258
検定結果のプロット 259
サンプル データ 254
信頼区間 270
推論方式 256
数値での結果 263
切片についての信頼区間 279
線形 t 検定 277
タイプ 262
データのインポート 263
データの入力 258, 260
適合度検定 275
統計のインポート 260
統計の計算 261
開く 254, 261
不要なデータ 260
平均応答についての信頼区間 280
方式 262
予測区間 281
数式処理システム 45
[数値のセットアップ]表示 64
共通操作 99
初期設定の復元 99
例 68
数値表示 64, 160
円周上 196
オブジェクト上 196
角度 195
カスタム表 95
カスタム表、データの削除 96
傾き 195
共通操作 92
共役 197
極座標 194
距離 194
計算の削除 163
計算の編集 162
弧長 195
コピーおよび貼り付け 96
[コマンド]メニュー 182, 193
座標 193
周長 194
垂直 196
ズーム 92
ズーム オプション 93
ズーム キー 94
ズーム ジェスチャ 94
[ズーム]メニュー 94
全オブジェクトのリスト 161
測定 194
[その他]メニュー 98
縦座標 193
デカルト 193
テスト 196
等辺形 197
共線 196
二等辺 196
パラメトリック 194
半径 194
評価 95
[プロット表示]での計算の表示 162
平行 196
平行四辺形 197
方程式 194
メニュー ボタン 98
面積 195
横座標 193
例 68
連携、[プロット表示] 99
[ズーム]メニュー
2 乗 82
X 拡大 80
X 縮小 81
Y 拡大 81
Y 縮小 82
拡大 79
三角法 84
自動スケール 83
縮小 80
小数 83
整数 84
[表示]メニュー 77
分割画面表示 78
ボックス ズーム 77
例 79
数列アプリケーション 304, 305
値の表 312
値の表、セットアップ 310
値の表、操作 309
グラフの操作 308
式の定義 305, 310
数値表示 308
数列のプロット 307, 311
開く 305
プロットのセットアップ 306, 311
明示的に定義された数列 310
スタックの操作 41
[スプレッドシート]アプリ
CHOOSE コマンド 214
スプレッドシート アプリケーション 204
CAS 計算 216
外部関数 212
外部参照 214
基本操作 209
計算での名前の使用 210
コピーおよび貼り付け 213
セルの名前設定 210
直接入力 211
変数の参照 217

せ

設定 18
ホーム 19, 20, 21
ホーム、指定 21

た

代数の優先順位 25
タッチ ジェスチャ 7

て

定義
色 71
構成要素 69
削除 72
選択 71

追加 69
評価 71
変更 69
ディスプレイ 4
データ
共有 32

と
解くアプリケーション 283
トレース
オン/オフ 86
関数の評価 86
プロットの選択 85

な
ナビゲーション 6

に
入力フォーム 17
リセット 18

ひ
表示
例 65

ふ
複素数 29
負数 25
[プロットのセットアップ]表示 63
共通操作 87
グラフ化の方式 90
初期設定の復元 92
[プロット表示]の設定 87
ページ 1 88
ページ 2 89
ページ 3 89
例 67
プロット表示 62, 150
ODE 173
円周上 181
オブジェクト上 181
オブジェクトの移動 152
オブジェクトの色の設定 152
オブジェクトの消去 153, 154
オブジェクトの選択 151
オブジェクトの塗りつぶし 152
オプション メニュー 155
回転 176
鏡映 175
拡大縮小 176
角度 180
陰 173
傾き 180
関数のプロット 172
キー 154
逆数変換 178
共通操作 74
共役 182
極座標 180
極のプロット 172
距離 180
勾配場 173
勾配場コマンド 156
弧長 180
コピーおよび貼り付け 87
[コマンド]メニュー 163
座標 179
ジェスチャ 154
周長 180
垂直 181
ズーム 75, 154
ズーム オプション 75
ズーム キー 76
ズーム ジェスチャ 76
ズーム倍率 75
数列のプロット 173
スライダー 174
相似 177
測定 180
縦座標 179
デカルト 179
テスト 181
点→複素数 179
投影 177
等辺形 182
共線 181
トレース 85
名前の非表示 152
二等辺 181
パラメトリック 180
パラメトリックのプロット 172
半径 180
反転 178
プロット 171
[プロットのセットアップ]表示 157
平行 181
平行移動 174
平行四辺形 182
[変換]メニュー 174
方程式 179
ボタン 154
メニュー ボタン 87
面積 180
横座標 179
リスト 173
例 66
連携、[数値表示] 99
[プロット表示]の[コマンド]メニュー 163
Δ三角形 166
◿三角形 167
∠二等分線 166
円 168
円錐 170
外接円 168
軌跡 170
曲線 168
交点 164
三角形 166
四辺形 167
垂線 166
垂直 165
正接 166
正多角形 168
正方形 168
線 165
線分 165
双曲線 170
傍接円 169
楕円 170
多角形 166, 168
中央 164
中点 164
長方形 167
直線 165
点 163
点オン 164
内接円 169
中線 166
半直線 165
菱形 167

平行　165
放物線　170
ランダム点　164

へ
ヘルプ　34
変数　28
[変数統計 2]アプリ
[関数]メニュー　251
スケッチ　252

ほ
ホーム表示　3

め
明示的な乗算　25
メニュー　16
項目の選択　17
ショートカット　17
ツールボックス　17
閉じる　17
メモリ管理者
使用　33
バックアップ カタログ　33