



## 目次

GRAPES-light for iOS USERS' MANUAL.....	1
第1章 GRAPES-light の基本.....	4
1-0 Windows 版 GRAPES との互換性について.....	4
1-1 GRAPES-light の外観.....	5
1-2 データパネルのカスタマイズ.....	6
1-3 プロジェクトとサンプル.....	7
1-4 プロジェクトの読み出し.....	8
1-5 プロジェクトの共有（保存）.....	9
1-6 ファイルの互換性.....	10
第2章 関数のグラフ.....	11
2-1 関数グラフの作成・追加.....	11
2-2 関数グラフのプロパティ.....	12
2-3 関数グラフの訂正と削除.....	12
2-4 グラフを動かそう.....	13
2-5 グラフを残す —残像—.....	13
2-6 定義域の制限.....	14
2-7 表示スイッチ.....	15
2-8 グラフの伸縮/移動.....	15
2-9 目盛の詳細設定.....	16
第3章 関係式のグラフ.....	17
3-1 関係式のグラフの作成・追加.....	17
3-2 不等式の領域.....	18
3-3 複数の領域の共通部分.....	19
第4章 点と軌跡.....	20
4-1 点を打つ.....	20
4-2 点のプロパティ.....	21
4-3 軌跡を描く.....	21
4-4 点のドラッグ.....	22
4-5 円, 垂直線, 水平線を描く.....	23
第5章 曲線.....	24
5-1 媒介変数表示の曲線.....	24

5-2 極方程式のグラフ .....	25
5-3 関係式と極方程式 .....	26
第 6 章 連結図形 .....	27
6-1 点を結ぶ .....	27
6-2 連結図形のプロパティと訂正・削除 .....	28
6-3 角の表示 .....	29
6-4 線分の長さや角の大きさの表示 .....	30
第 7 章 関数の定義と相互参照 .....	31
7-1 定義関数を利用する .....	31
7-2 関数を参照する .....	32
7-3 グラフをドラッグする 一点の座標の参照 .....	32
第 8 章 ベクトルの利用 .....	33
8-1 点とベクトル .....	33
8-2 座標やベクトルで入力する .....	34
8-3 ベクトルの演算 .....	35
8-4 距離と角度と面積 .....	36
8-5 位置を与える関数 .....	37
8-6 定義関数とベクトル .....	40
第 9 章 関数を使いこなす .....	41
9-1 グラフの詳細設定 .....	41
9-2 関数の詳細設定 .....	42
9-3 GRAPES の関数 .....	43
9-4 特殊な関数 .....	44
9-5 ベクトル（点）に関する関数 .....	45
9-6 評価の優先順位 .....	48
9-7 導関数と不定積分 .....	49

# 第 1 章 GRAPES-light の基本

---

## 1-0 Windows 版 GRAPES との互換性について

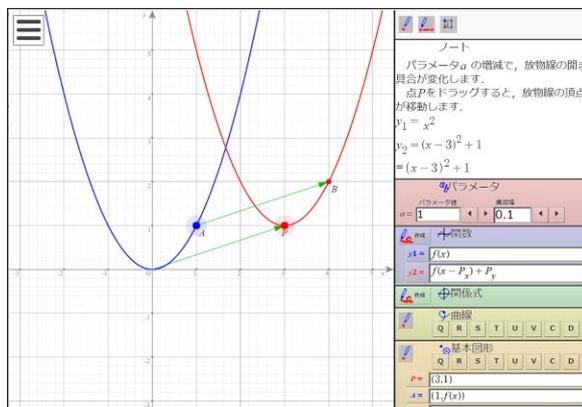
---

- グラフ描画関係
  - 関係式の一部機能を除いて GRAPES とほぼ同等
  - × 関係式では、次の機能に未対応
    - ◇ “and” や “or” を用いた共通部分や和集合の領域表示
    - ◇ 軌跡点機能
    - ◇ 境界線の線種の指定（実線のみ）
  - × 曲線では「パラメータの同期」機能がない
  - × グラフの縦横比の変更ができない
- パラメータの扱い
  - GRAPES とほぼ同等
  - × 関数電卓を用いた値の入力はできない。
  - × パラメータの定義機能がない。
- 点のドラッグ
  - 一部機能を除いて、GRAPES と同等
  - × 任意曲線に貼り付く機能はない
- ステッカー
  - データパネル内に表示
  - × 編集は不可（サンプルや外部から読み込んだプロジェクトの内容表示はできる）
- テーブル
  - × 編集は不可（サンプルや外部から読み込んだプロジェクトのデータの利用はできる）
- ファイル操作
  - クラウドドライブや AirDrop を用いた GRAPES ファイルの読み書きが可能
  - GRAPES とは完全なファイル互換性がある。
- GRAPES-light にはできないこと
  - × スクリプトの実行
  - × 背景機能
  - × マーカー機能
  - × （関数電卓下部などの）ヒントメッセージの表示

## 1-1 GRAPES-light の外観

### ● グラフパネルとデータパネル

GRAPES-light は、2つのパネル、グラフパネルとデータパネルで構成されています。グラフパネルでは、文字通りグラフが表示されます。データパネルでは、関数の式やパラメータの値など、グラフの元になるデータを入力したり操作したりします。



### ● メニューボタン

メニューボタン を押すと、メニューが開き、[プロジェクトの初期化]、[サンプルを開く]、[プロジェクトの共有] (ファイルの書き出し)、各種 [オプション] の設定、[残像消去] など補助的な機能呼び出すことができます。

### ● データパネル

#### ◆ ツールバー

よく使う機能のボタンが配置されます。

#### ◆ ノートパネル

GRAPES のステッカーがここに表示されます。

#### ◆ パラメータパネル

パラメータの操作を行います。

#### ◆ 関数パネル

$y = f(x)$  の形で表された関数のグラフを扱います。

#### ◆ 関係式パネル

$x, y$  についての等式や不等式で表された関係のグラフや領域を扱います。

#### ◆ 曲線パネル

パラメータで表された曲線や、極方程式のグラフを扱います。

#### ◆ 基本図形パネル

点、円、水平線、垂直線を扱います。

#### ◆ 関数定義パネル

関数を定義するとき、ここに式を書きます。

#### ◆ 連結図形パネル

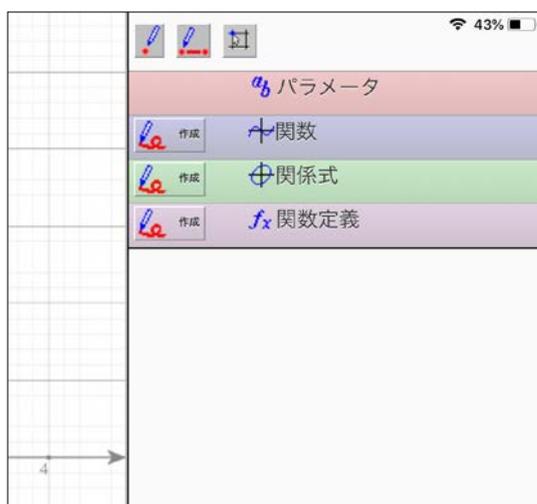
線分や多角形などの連結図形の設定情報が表示されます。

## 1-2 データパネルのカスタマイズ

データパネル内の各小パネルは、表示／非表示を切り替えることができます。

- 各パネルの表示／非表示切替

1. メニュー (☰) の [エリアの表示] をタップ  
各パネル名が表示されます。
2. 非表示にするパネルのチェックを外す (右図)



- 次の機能には対応していません。

- ◆ 各パネルの表示順序の変更
- ◆ 個々のデータの表示／非表示切替
- ◆ 個々のデータの表示順序の変更

## 1-3 プロジェクトとサンプル

### ● プロジェクト

GRAPES-light で作成されたデータをプロジェクトといいます。ひとつのプロジェクトには、グラフの方程式、パラメータ値、残像設定、目盛設定など、GRAPES-light で作ったデータや設定情報がすべて含まれています。

GRAPES のプロジェクトには、拡張子 “.gps” が付いています。

### ● プロジェクトの初期化

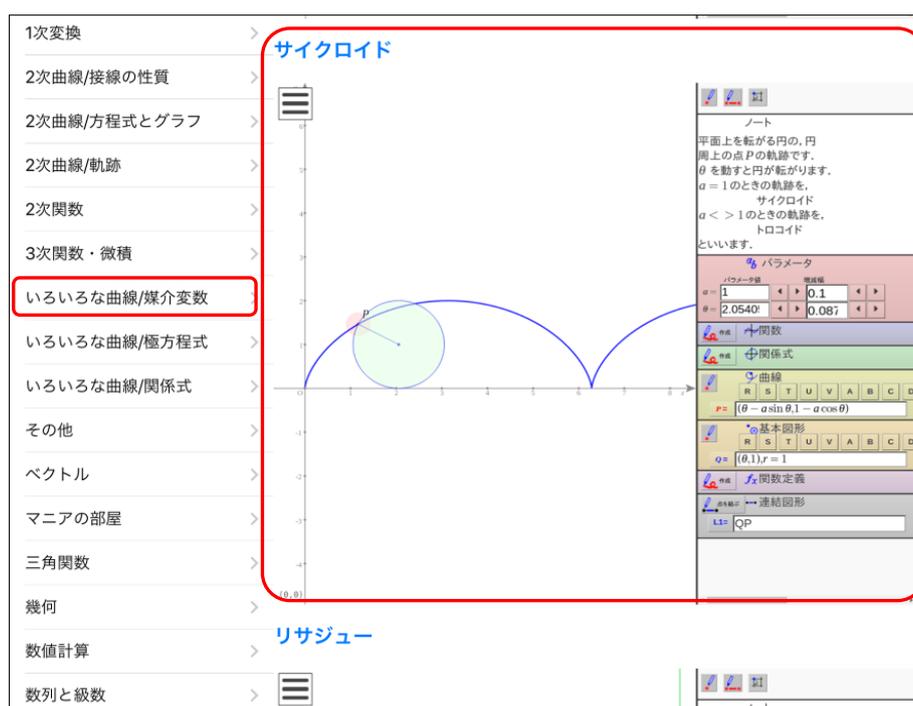
メニュー (☰) から [プロジェクトの初期化] をタップ

操作中のデータはすべて消去され、起動時の状態に戻ります。



### ● サンプルを開く

1. メニュー (☰) から [サンプルを開く] をタップ
2. 画面左に表示されるカテゴリーを選ぶ
3. プロジェクトを選びタップ



## 1-4 プロジェクトの読み出し

### ● プロジェクトの読み出し

Dropbox などクラウドドライブから、プロジェクトを読み出すことができます。

以下は、Dropbox による読み出し方法の説明です。

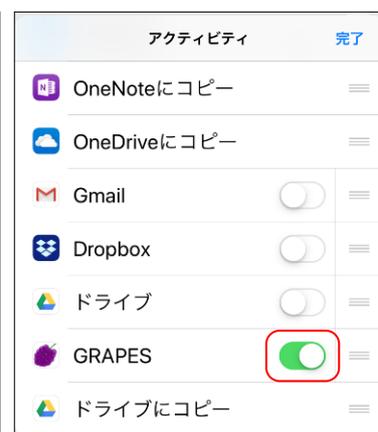
1. クラウドドライブから、GARPEES プロジェクトファイル (“\*.gps”) を選び、[…] をタップする。



2. [エクスポート] を選び [別のアプリで開く] を選ぶ。



3. メニューに「GRAPES」がないときは、上段の [その他...] を選び、「GRAPES」を ON にする。



4. メニューから [GRAPES] を選ぶ。

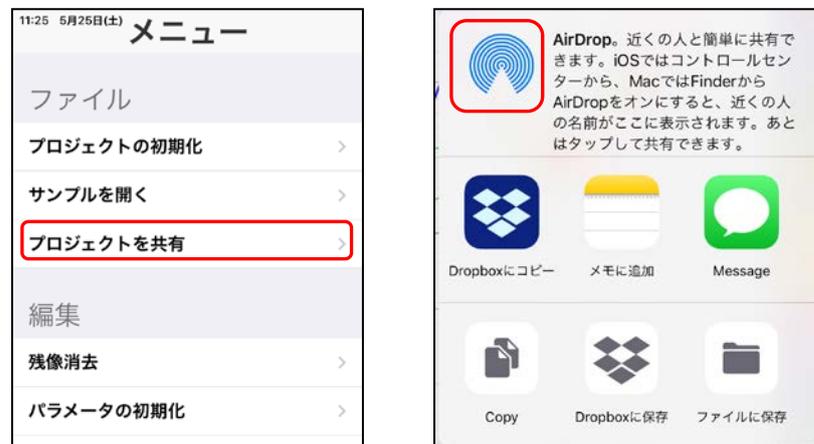


## 1-5 プロジェクトの共有（保存）

GRAPES-light で作成したプロジェクトを AirDrop やクラウドドライブを介して他の iPad (iPhone) と共有することができます。クラウドドライブに書き出したファイル (gps ファイル) は、GRAPES のプロジェクトファイルと完全な互換性があり、GRAPES から読み書きすることもできます。

### ● プロジェクトの共有（AirDrop を利用する場合）

3. メニューの [プロジェクトを共有] をタップ
4. 近くにある iPad や iPhone にファイルを送るときは [AirDrop] をタップし、送付先を選ぶ



### ● プロジェクトの共有（DropBox に保存する場合）

1. メニューの [プロジェクトを共有] をタップ
2. [DropBox に保存] を選ぶ
3. ファイル名と保存先フォルダを確認（必要なら書き替える）
4. [保存] をタップ



---

## 1-6 ファイルの互換性

---

GRAPES のプロジェクトファイルと GRAPES-light のプロジェクトファイル (gps ファイル) には完全な互換性がありますから、GRAPES と GRAPES-light 間で相互にプロジェクトを共有することができます。

- **GRAPES-light 未対応機能の扱い**

GRAPES-light は gps ファイルのすべてを読み込みますが、未対応機能のデータは無視して実行します。そのプロジェクトを書き出す際には、元のデータをそのまま書き出します。

- ◆ **例 1 : スクリプトの扱い**

スクリプトを含んだ gps ファイルを GRAPES-light が読み込んでも、スクリプトは実行されません。ただし、そのプロジェクトを (そのまま、あるいは一部を書き替えて) 保存した場合、元のスクリプトはそのまま書き出されます。

- ◆ **例 2 : ステッカーの扱い**

グラフ画面上にステッカーのある gps ファイルを GRAPES-light で読み込んだ場合、ステッカーはすべてデータパネルに表示されます。このプロジェクトを保存して GRAPES で開くと、ステッカーはグラフ画面上の元の位置に表示されます。

# 第2章 関数のグラフ

## 2-1 関数グラフの作成・追加

### ● 関数のグラフの追加・作成

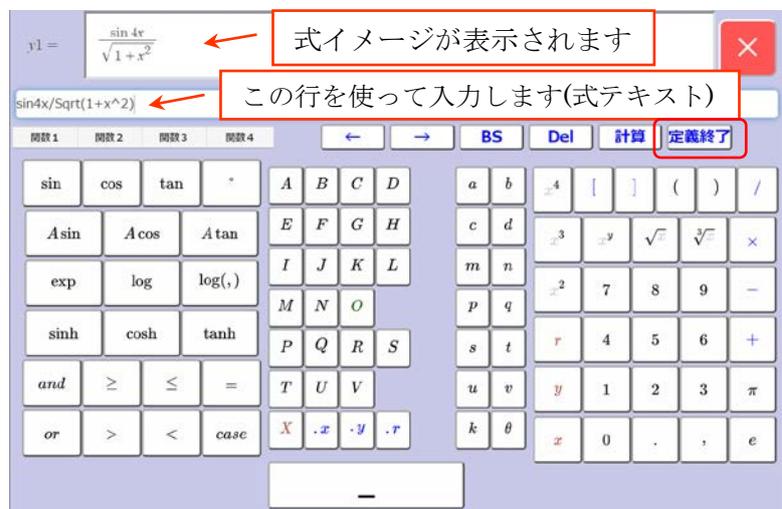
1. 関数パネルの [作成] をタップ



2. 関数電卓で式入力

関数電卓のボタンを使って入れます。

入力を終わったら、[定義終了] をタップします。

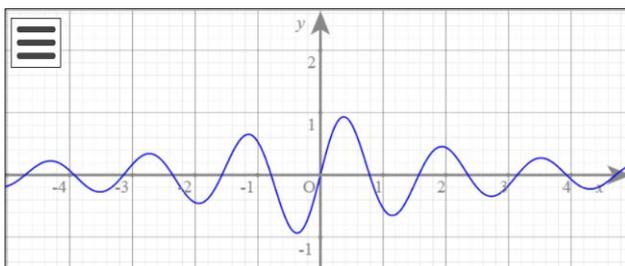


3. グラフスタイルの設定

関数グラフのプロパティが表示されます。

グラフ色や太さの設定方法は、次ページを参考にしてください。

設定を終わったら [OK] をタップします。



## 2-2 関数グラフのプロパティ

関数グラフのプロパティでは、関数式、グラフ色、太さ、線種、残像設定を行うことができます。

### ● 式の訂正

1. 関数式表示窓をタップ
2. 関数電卓で式を訂正

### ● グラフ色の変更

1. グラフ色表示窓をタップ
2. パレットから色を選ぶ

### ● グラフの太さの変更

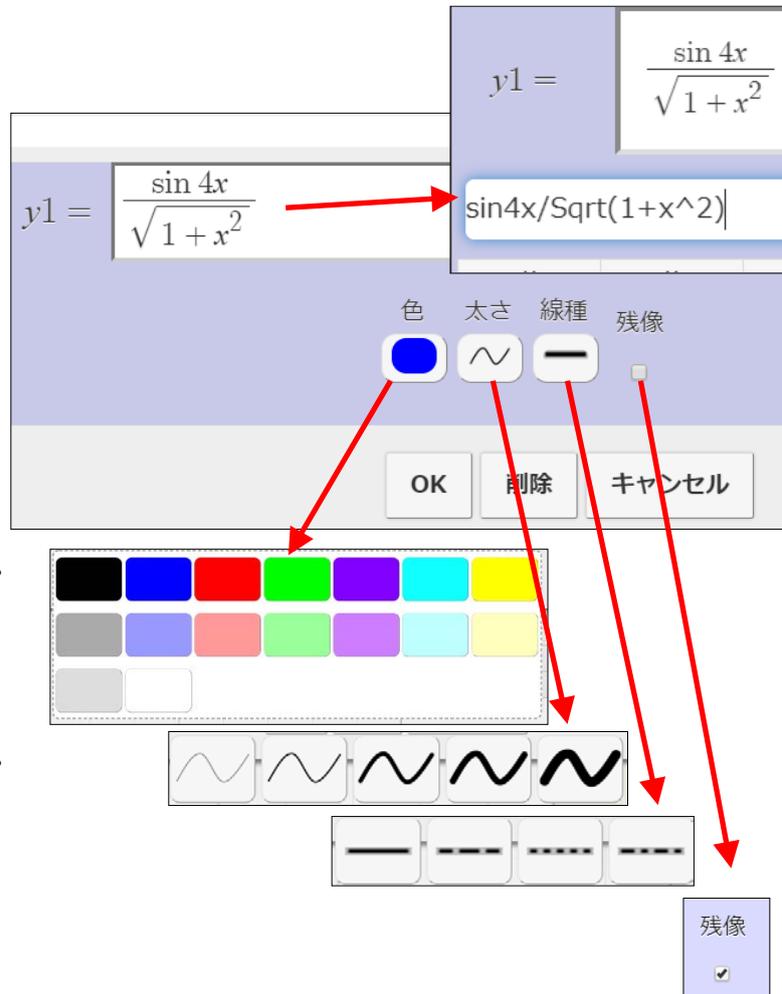
1. グラフの太さ表示窓をタップ
2. パレットから太さを選ぶ

### ● グラフの線種の変更

1. グラフの線種表示窓をタップ
2. パレットから線種を選ぶ

### ● 残像設定の変更

[残像] をチェック



## 2-3 関数グラフの訂正と削除

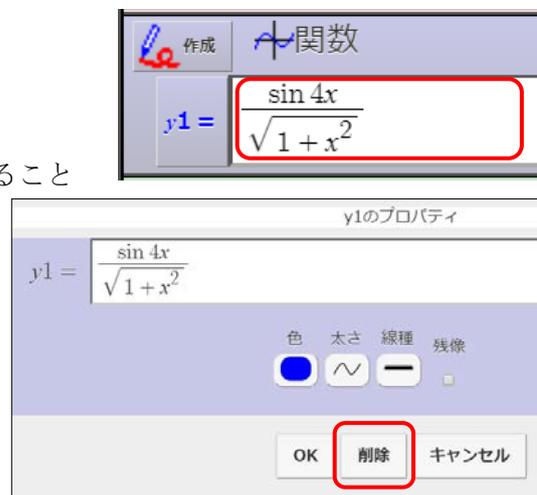
### ● 関数グラフの訂正

1. 関数パネルの関数式表示窓をタップ
2. 関数のプロパティで訂正

グラフの式、色、太さ、残像の有無を変更することができます。

### ● 関数グラフの削除

関数グラフのプロパティで [削除] をタップ



## 2-4 グラフを動かそう

- 14個のパラメータ

パラメータとして、次の14個の文字を使うことができます。

$$a, b, c, d, k, m, n, p, q, s, t, u, v, \theta$$

パラメータは、その値を変化させる意味があるときに表示されます。

- パラメータ値の増減

増減ボタン   をタップ

☆ 押しつづけると連続的に変化します。

★ 増減の速さを変化させるには、増減ボタンを  
押したまま、左右にスライドしてください。



- 増減幅の変更

増減幅変更ボタン   をタップ

- パラメータへの値の代入

パラメータ値表示窓をタップしてキーボードから入力

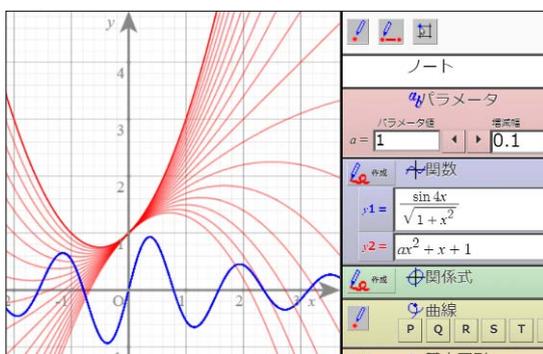
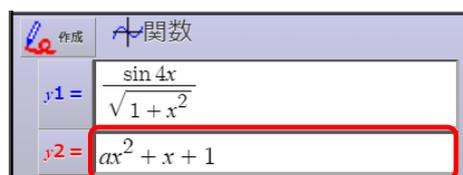


## 2-5 グラフを残す —残像—

パラメータの増減によって動いたグラフを残すことができます。それが「残像」です。

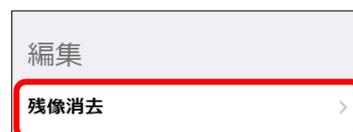
- 残像を残す

1. 残像を残したいグラフの関数式をタップ
2. [残像] をチェック
3. [OK] をタップ
4. パラメータを動かす



- 残像を消す

メニューボタン  をタップし、[残像消去] を選びます。



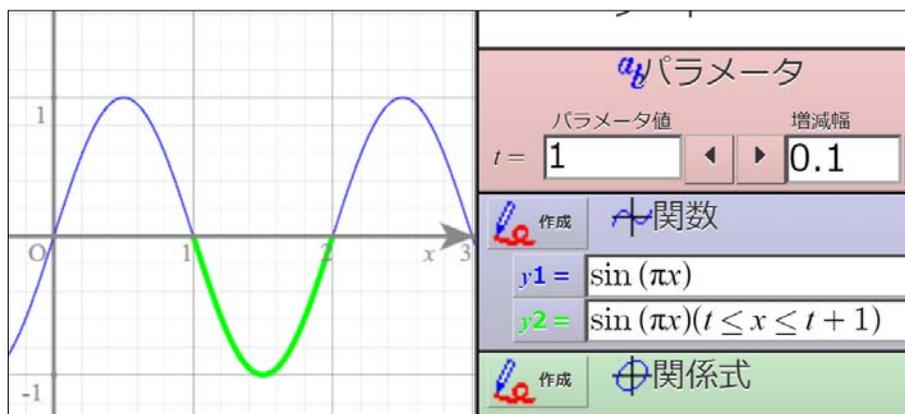
## 2-6 定義域の制限

### ● 定義域の制限

関数の定義域を制限するには、関数式の後または前に定義域を付け加えます。

例：  $y = \sin \pi x (t \leq x \leq t+1)$

(「 $\sin \pi x$ 」と入力してもシステムが  $()$  を付加することがあります)



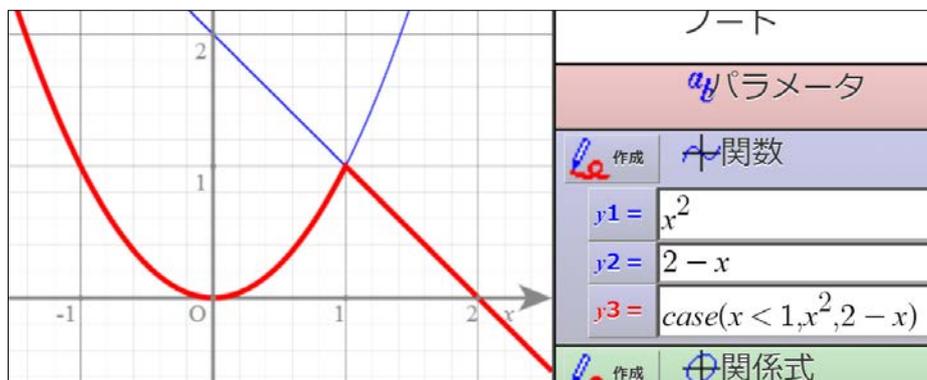
上の例では、  $y2 = y1(t \leq x \leq t+1)$  と書くこともできます。

$t \leq x \leq t-1$  は、  $(t \leq x)$  and  $(x \leq t+1)$  と書くこともできます。

### ● 定義域に制限のある関数の合成

$y = \begin{cases} x^2 & (x < 1) \\ 2-x & (x \geq 1) \end{cases}$  のように、定義域に制限のある2つ以上の関数が組み合わさっている場合、

$y = \text{case}(x < 1, x^2, 2-x)$  (あるいは  $y = x^2 (x < 1) \text{ or } (2-x)(x \geq 1)$ ) とします。



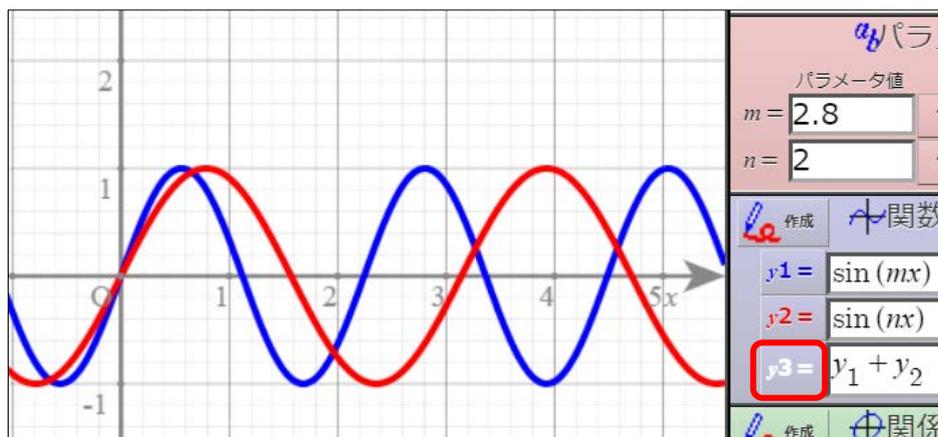
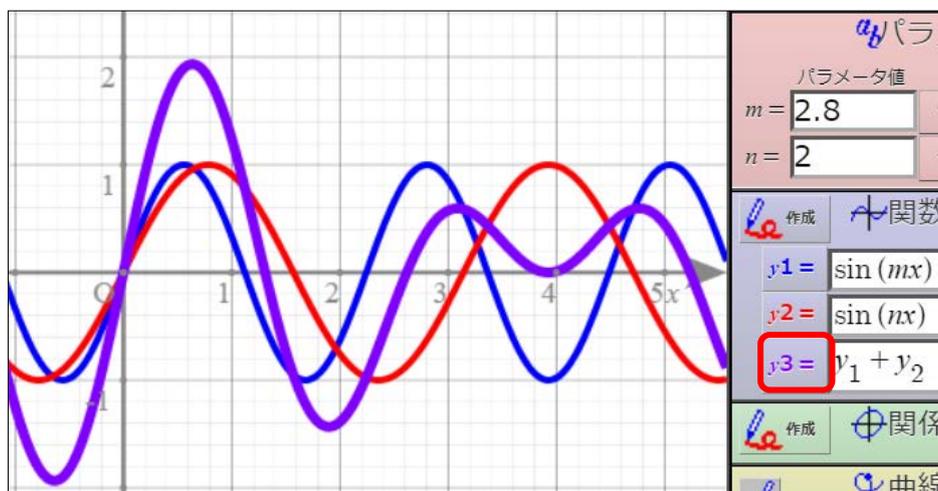
上の例では、  $y3 = x^2(x < 1) \text{ or } (2-x)(1 \leq x)$  と書くこともできます。

## 2-7 表示スイッチ

任意のグラフについて、データパネルのラベルをタップするだけで、表示・非表示を瞬時に切り替えることができます

- グラフの表示と非表示

グラフのラベル文字をタップ（ここでは  $y_3$  をタップ）



表示スイッチで非表示にしたグラフでは、残像も非表示になります。

## 2-8 グラフの伸縮/移動

- ピンチイン・ピンチアウトによる伸縮

ピンチインで縮小，ピンチアウトで拡大

2本指の中央を基準に，ピンチインで縮小，ピンチアウトで拡大される。

- 二本指スライドによる移動

二本指で画面をスライドする。

## 2-9 目盛の詳細設定

- 「目盛」設定の表示

メニューボタン をタップし、[オプション] から [目盛] を選びます。



- 目盛表示の設定

格子線を消すには [格子線] のチェックを外す

目盛文字を消すには [目盛文字] のチェックを外す

- 軸表示の設定

軸を消すには [軸] のチェックを外す

このとき、目盛文字も表示されなくなります。

軸ラベルを変更するにはラベル表示窓を書き換える

軸ラベルを“x”や“y”から任意の文字に変更  
することができます。



- 目盛文字位置の設定

表示したい文字位置を選ぶ

 を選ぶと、格子線をよけて文字が表示されます。

 を選ぶと、格子線の真上に文字が表示されます。

- 軸の種類

軸の太さや矢印の有無を選ぶことができます。

- 次の項目は現時点では未実装です。

「°」チェックボックス，目盛幅基準値

# 第3章 関係式のグラフ

## 3-1 関係式のグラフの作成・追加

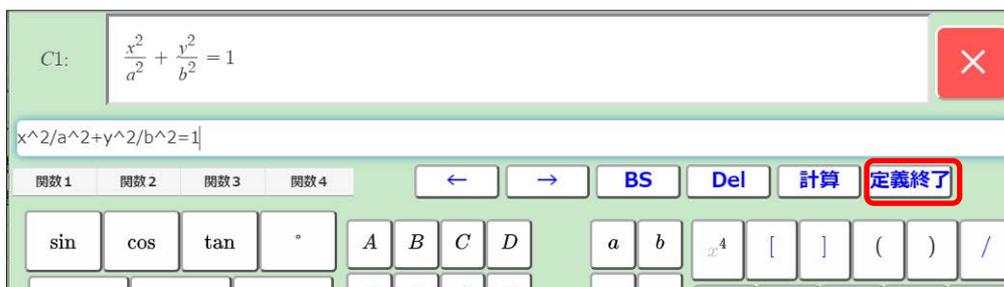
- 関係式のグラフの作成・追加  
関係式パネルの「作成」をタップ



### 2. 関数電卓で式入力

関数電卓のボタンで等式を入力します。

入力を終わったら、「定義終了」をタップします。



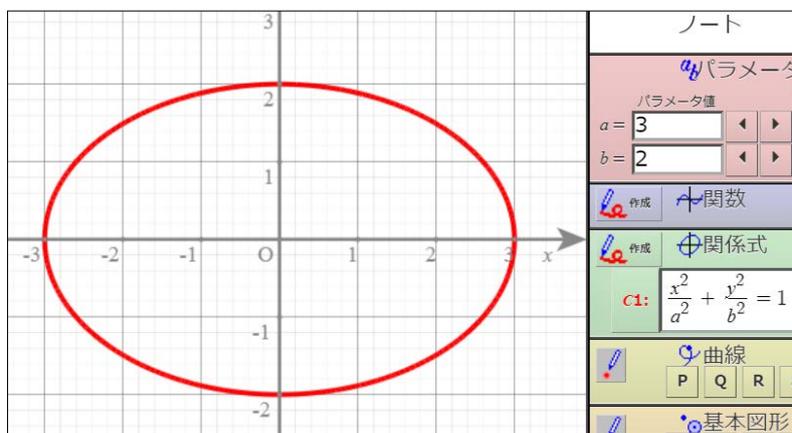
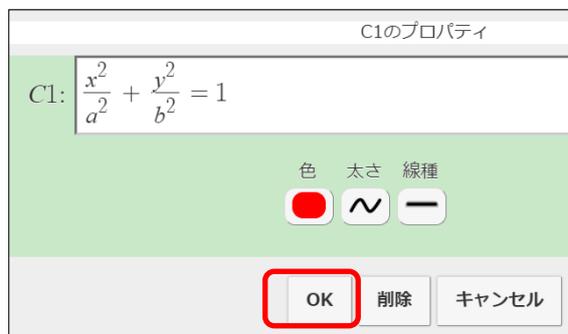
### 3. グラフスタイルの設定

関係式のプロパティが表示されます。

グラフ色や太さなどの設定を終えたら

「OK」をタップします。

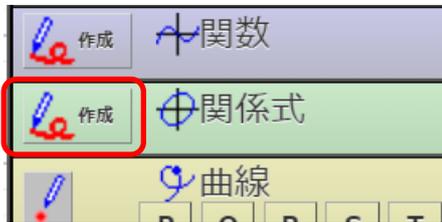
現バージョンでは、線種には対応していません。



## 3-2 不等式の領域

### ● 不等式の領域

#### 1. 関係式パネルの [作成] をタップ



#### 2. 関数電卓で式入力

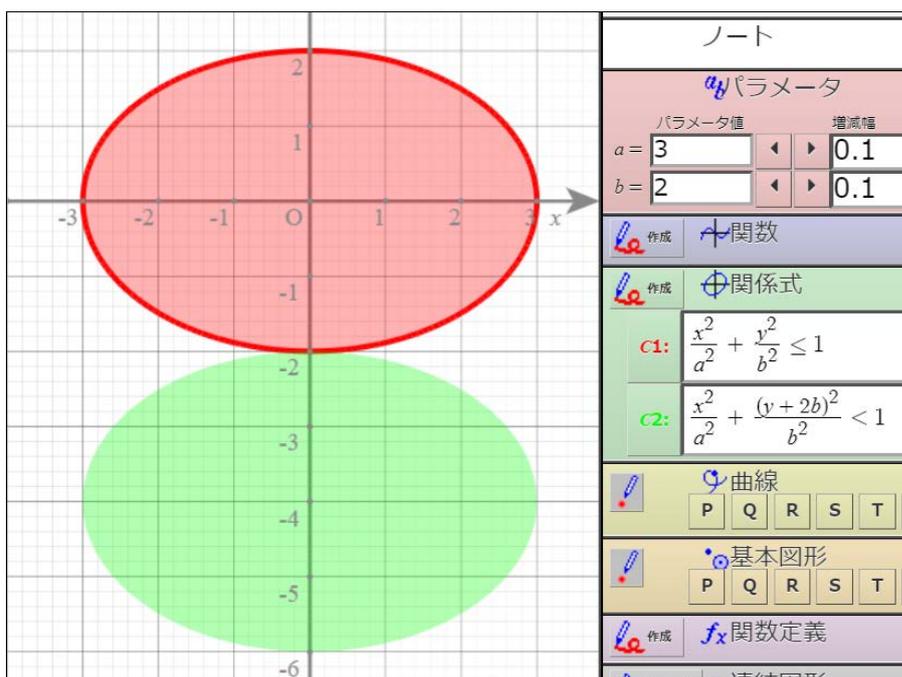
関数電卓のボタンで不等式を入力し、[定義終了] をタップします。



#### 3. グラフスタイルの設定

関係式のプロパティが表示されます。

グラフの色や太さを選択したら、[OK] をタップします。



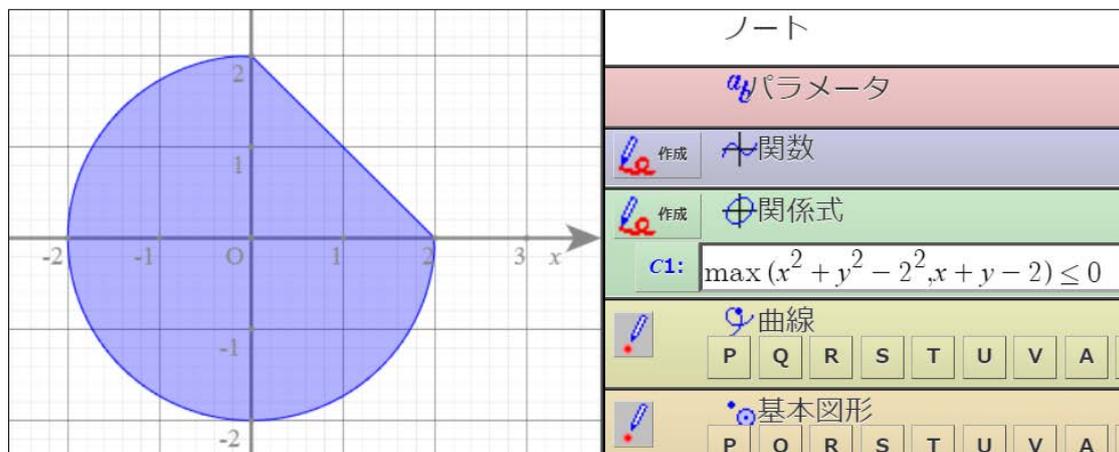
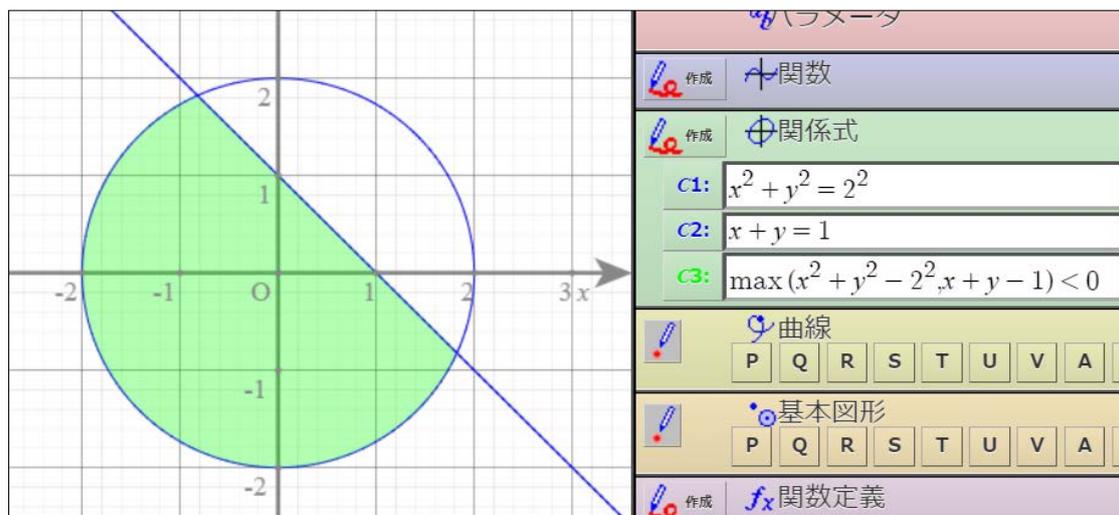
### 3-3 複数の領域の共通部分

Windows 版 GRAPES は, “and” や “or” を用いて共通部分や和集合を描くことができませんが, GRAPES-light にその機能はありません。ただ, 式の工夫で類似のことは可能です。

- 複数の領域の共通部分の表示 (max 関数の利用)

例:  $x^2 + y^2 \leq 2^2$  かつ  $x + y \leq 1$  ならば

$$\max(x^2 + y^2 - 2^2, x + y - 1) < 0 \text{ とします。}$$



# 第4章 点と軌跡

## 4-1 点を打つ

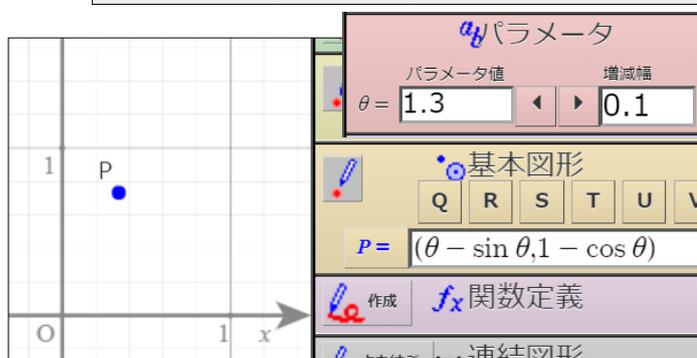
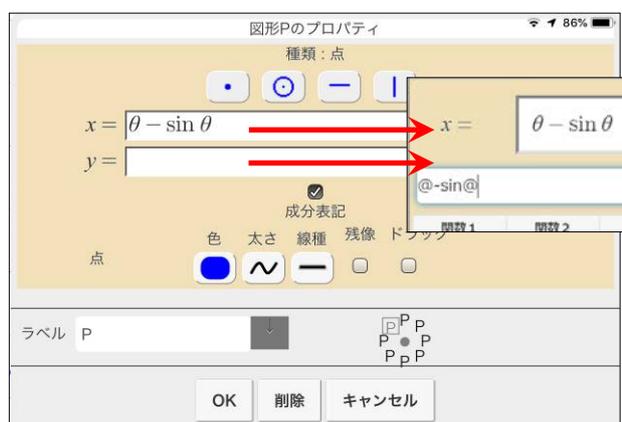
- 名前, 座標, 色などをすべて入力して点を打つ

1. 基本図形パネルに表示されている図形名をタップ
2. 図形の種類を選ぶ。
3. x, y に式を入力  
式表示窓をタップし, 関数電卓で  
入力する。

☆ グラフパネルでタップすると, その  
位置の座標が読み込まれます。

☆ グラフパネルの既存の点をタップす  
ると, その点の座標や式が読み込ま  
れます。

4. 色, 大きさなどを設定して  
[OK] をタップ



- ドラッグできる点を簡単に打つ

1. ツールパネルもしくは基本図形パネルの  
プロットボタン をタップ

「プロットモード」になります。

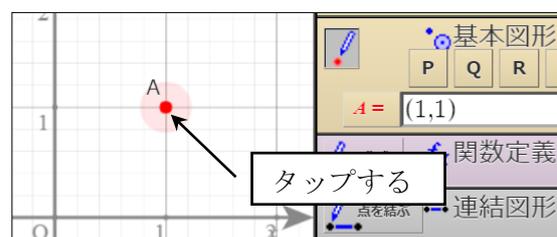
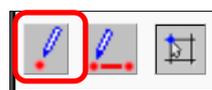
2. グラフパネルでタップ

これで点がプロットされます。

この操作は続けることができます。

3. プロットボタン をタップ

「プロットモード」を解除します。



## 4-2 点のプロパティ

- 座標の入力・訂正

1. 座標の式表示窓をタップ
2. 関数電卓で訂正

- 点の色や大きさの変更

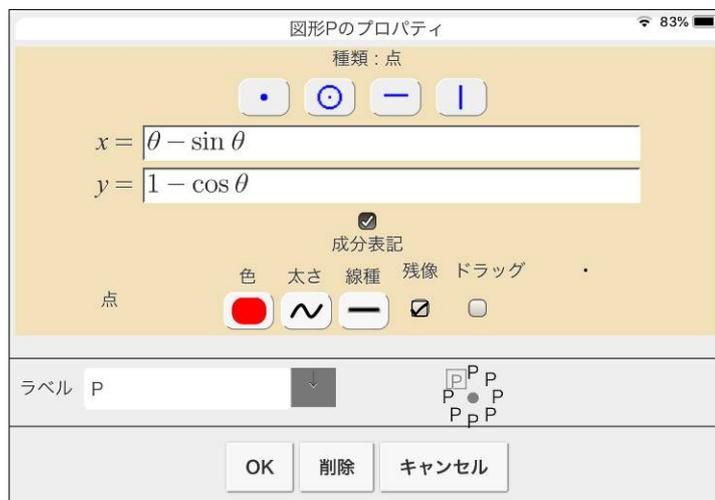
1. 点の色や大きさ表示窓をタップ
2. パレットから選ぶ

- 残像設定

[残像] をチェック

- ラベルの表示設定

ラベル表示内容を入力し（または選び）、表示する位置をタップ



## 4-3 軌跡を描く

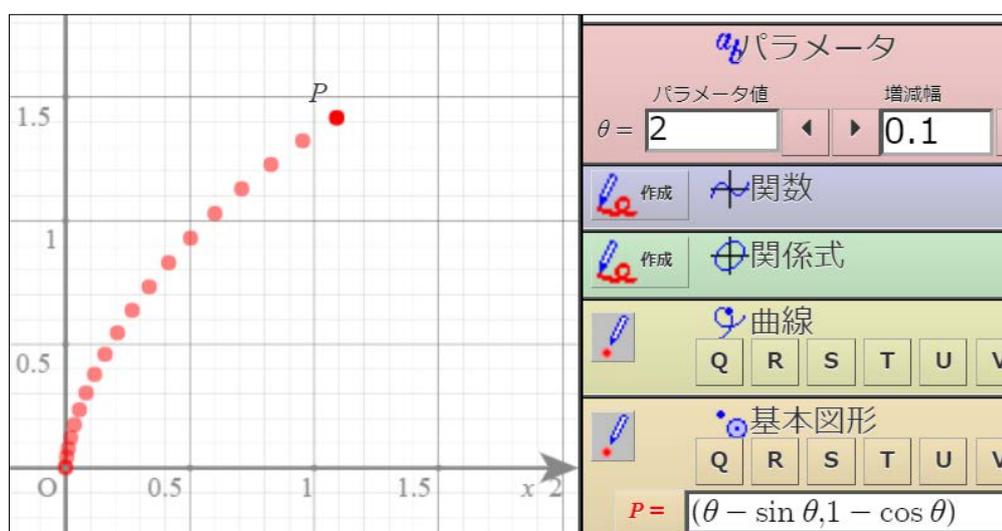
パラメータを動かすと、点が動き軌跡が描かれます。

- パラメータ値の増減

増減ボタン   をタップ

☆ 押しつづけると連続的に変化します。

★ 増減の速さを変化させるには、増減ボタンを  
押したまま、左右にスライドしてください。



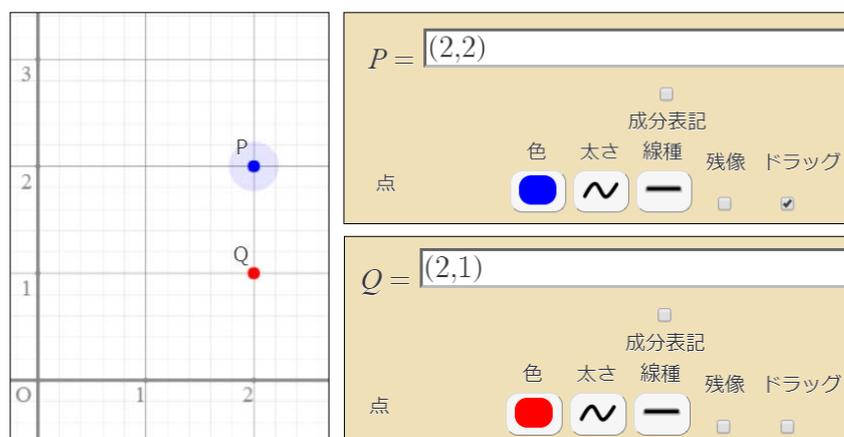
- 残像を消す

メニューボタンをタップし、[残像消去] を選ぶ

## 4-4 点のドラッグ

### ● 点のドラッグ

基本図形は、座標が数値で且つ「ドラッグ」がチェックされているとき、ドラッグして動かすことができます。ドラッグ可能な点は傘をかぶったように表示されます。



### ● 点のドラッグのオプション

#### ◆ 格子線上をドラッグ

領域パレットの格子線追跡ボタン を押し下げた状態でドラッグすると、格子線に吸い寄せられます。

### ● 円のドラッグ

円の中心をドラッグすると円が動きます。円周をドラッグすると半径が変化します。

(次ページ参照)

### ● 曲線上の点のドラッグ

曲線上に点を取り、その点をドラッグすることができます。

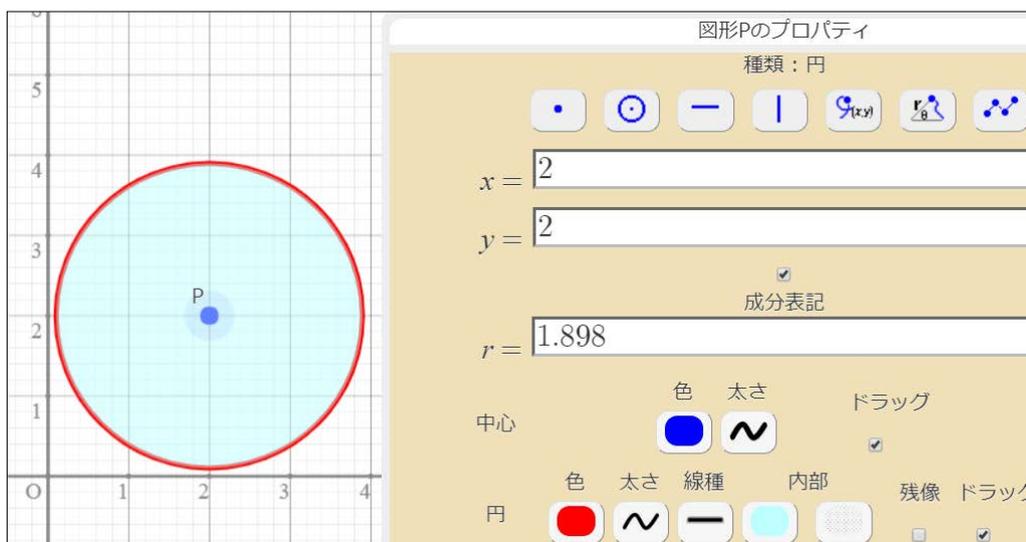
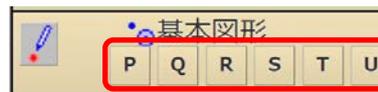
(5-1「媒介変数表示の曲線」参照)

## 4-5 円, 垂直線, 水平線を描く

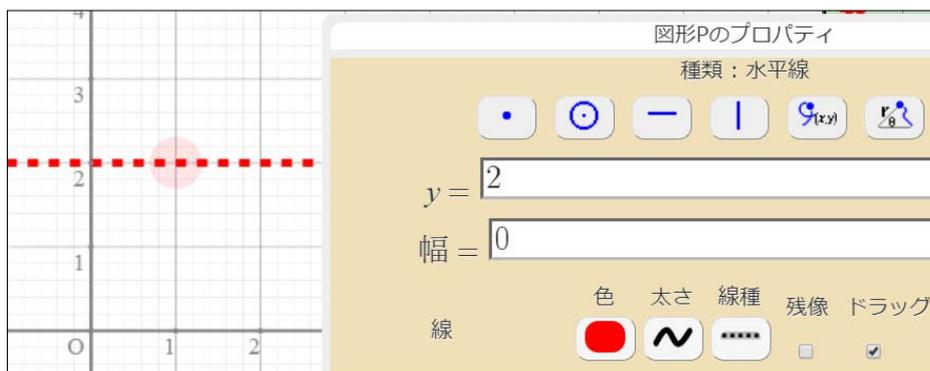
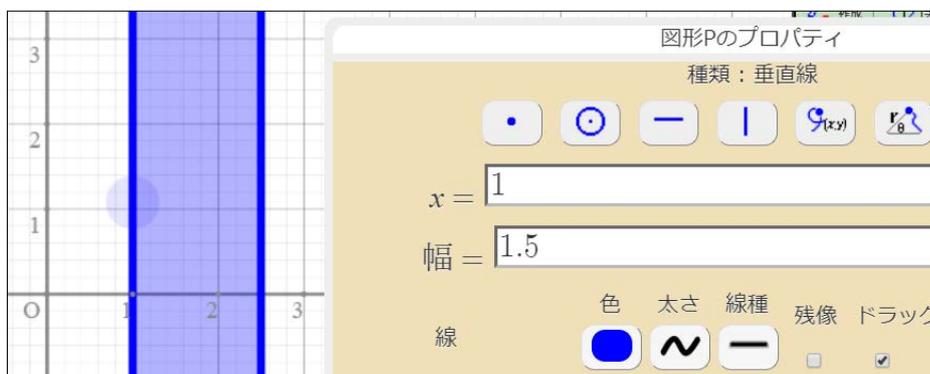
- 図形名を選び, ついで図形種類を選ぶ

あとは, 点の場合とほとんど同じです。

- 円とプロパティ



- 垂直線, 水平線とプロパティ

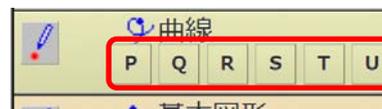


# 第5章 曲線

## 5-1 媒介変数表示の曲線

### ● 媒介変数表示の曲線を描く

1. 曲線パネルの図形名を選ぶ



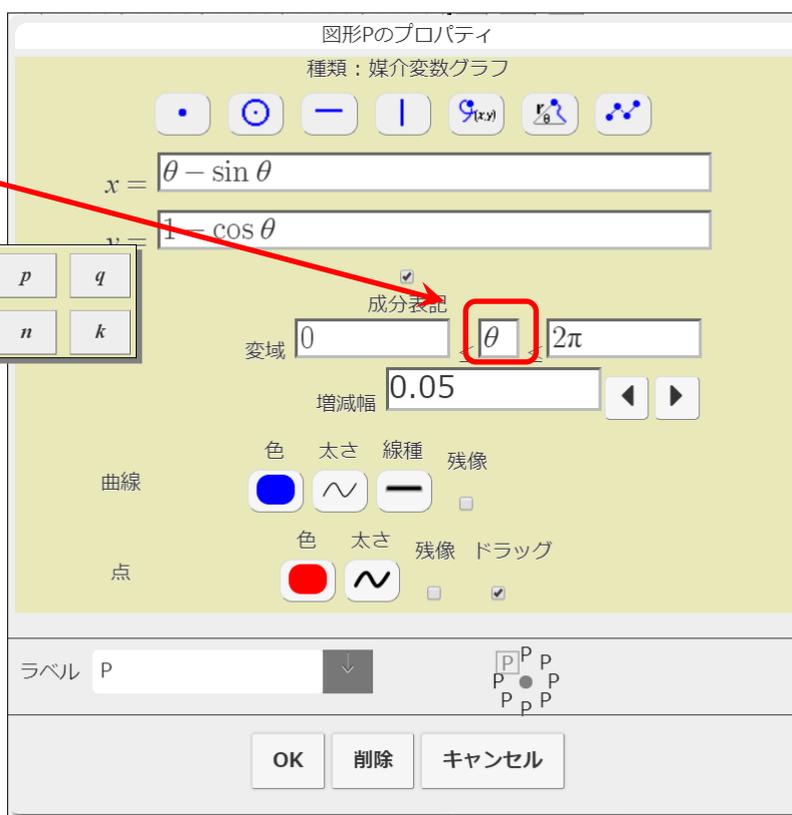
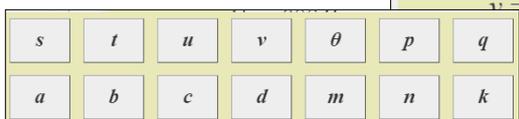
2. 図形の種類の曲線ボタン  をタップ



3.  $x, y$  に関数式を入力

4. 媒介変数の設定

ここをタップして  
媒介変数を変更する。



5. 変域の設定

変域の下減と上限の表示  
窓をタップして関数電卓で  
入力します。

6. 増減幅の設定

増減幅を細かくするほど  
曲線は滑らかになります。

7. 曲線のプロパティの設定

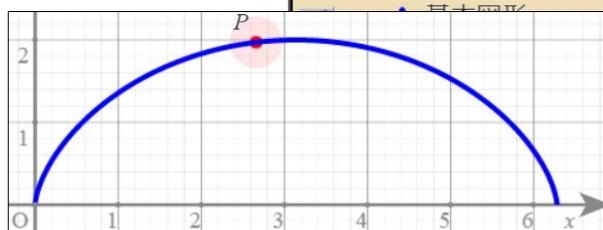
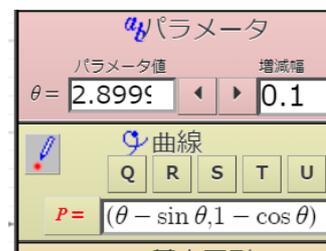
色, 太さ, 線種, 残像を設定します。

8. 点のプロパティの設定

曲線上を動く点を描くことができます。

★ [ドラッグ] をチェックすると, 曲線上  
の点をドラッグすることができます。

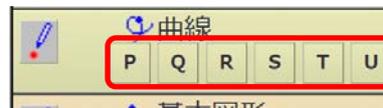
9. [OK] をタップ



## 5-2 極方程式のグラフ

### ● 極方程式のグラフを描く

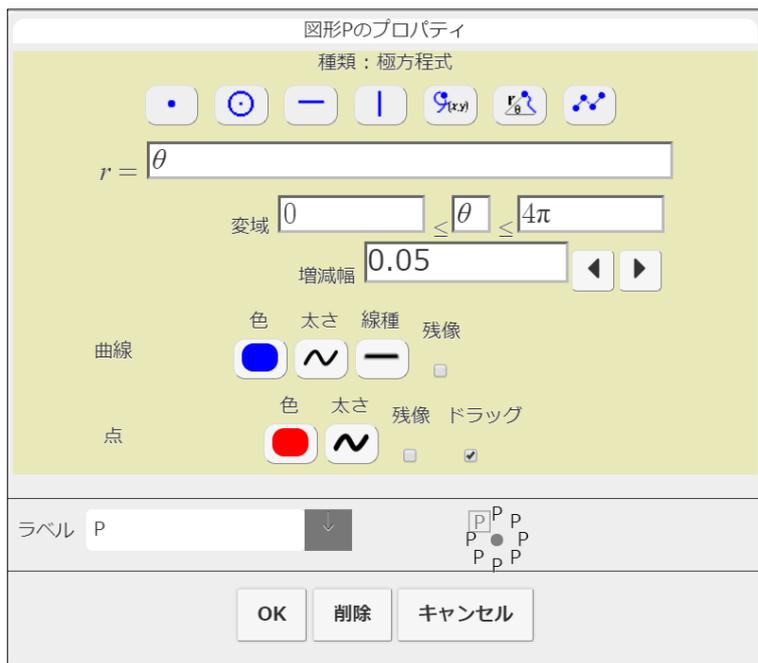
1. 曲線パネルで図形名を選ぶ



2. 図形の種類の「極方程式」ボタン  をタップ



3.  $r$  の式を  $\theta$  を用いて入力

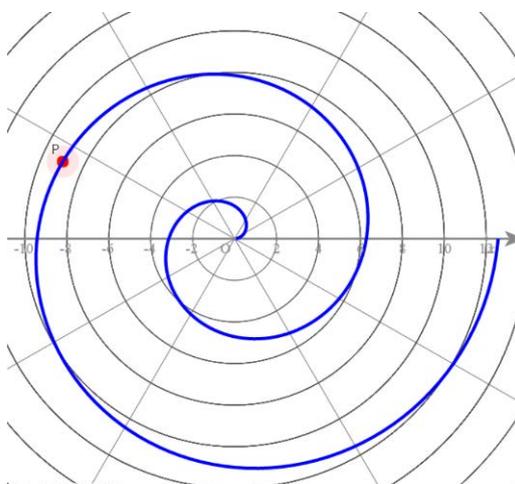


4.  $\theta$  の変域や増減幅の設定

変域表示窓をタップして、  
変域の下限と上限をそれぞれ  
設定します。

5. 曲線や点、ラベルのプロパティの設定

6. [OK] をタップ



## 5-3 関係式と極方程式

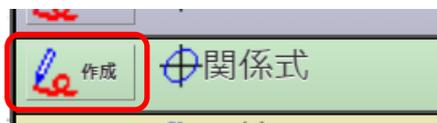
関係式では、動径  $r$  と偏角  $\theta$  使うことができ、極方程式のグラフを描くことができます。

- 関係式と極方程式

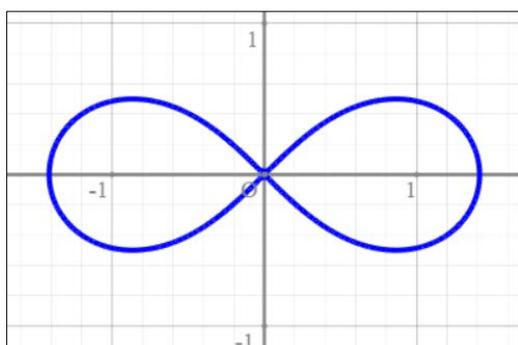
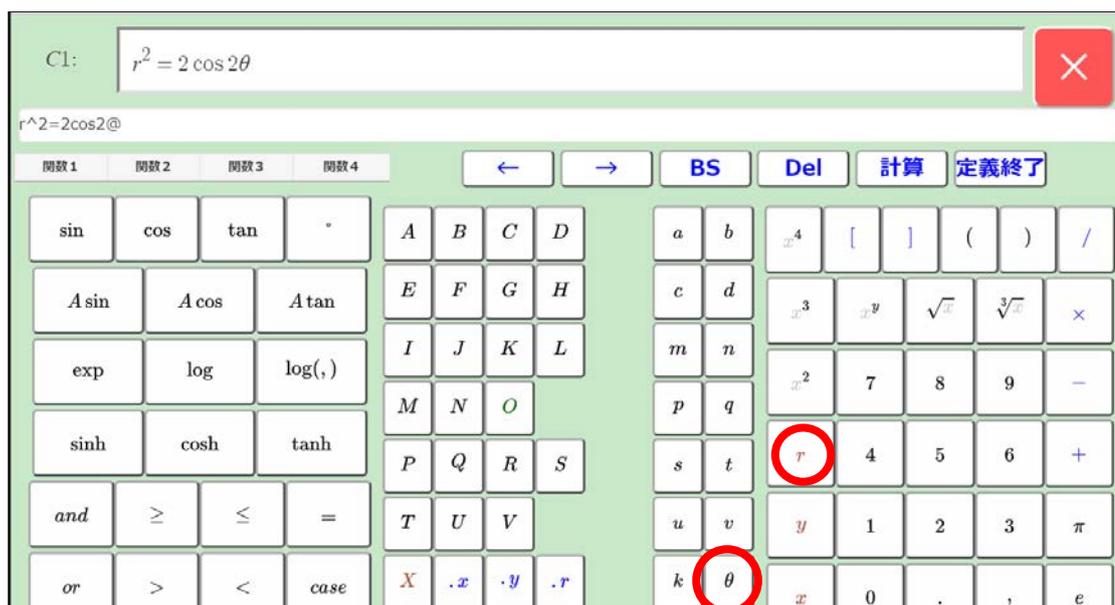
1. オプションの「関数」設定で、「定義関数や関係式中の  $\theta$  は偏角とみなす」をチェック  
以後、 $\theta$  はパラメータではなく、偏角とみなされます。(デフォルトは ON です)



2. 関係式パネルの「作成」をタップ
3. 電卓で式入力

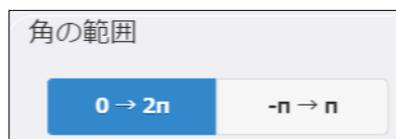


動径  $r$  ( $=\sqrt{x^2 + y^2}$ ), 偏角  $\theta$  を用いて式を入力します。



- 関係式での動径と偏角の扱い

偏角  $\theta$  の範囲は、 $0 \leq \theta < 2\pi$  か  $-\pi \leq \theta < \pi$  です。いずれにするかは、オプションの「関数」設定の「角の範囲」で設定します。



関係式の中では負の動径は認められていません。

# 第6章 連結図形

## 6-1 点を結ぶ

### ● 点を結ぶ

1. コントロールパレットもしくは連結図形パネルの点を結ぶボタン をタップ

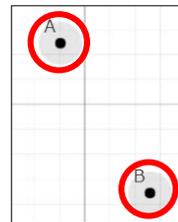


連結図形のプロパティが開きます。

2. グラフパネルで点を選ぶ

結びたい点を順にタップしていきます。

最大 22 個まで結ぶことができます。

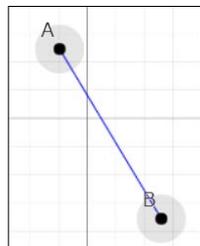


3. 図形の諸設定

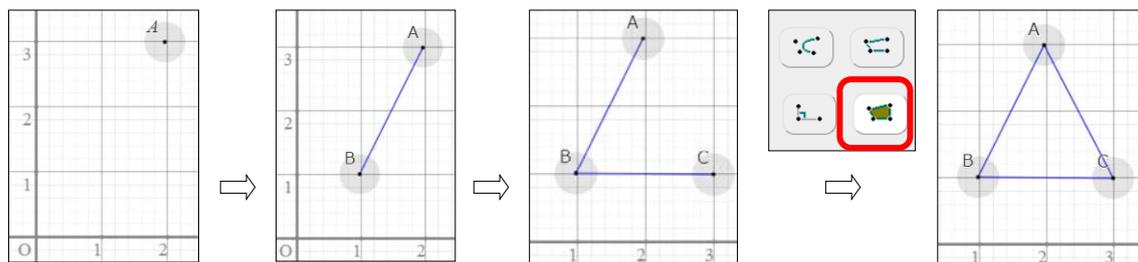
連結図形のプロパティで、図形の種類、色、太さ、領域色、残像の有無を設定します。



4. [OK] をタップして決定



- ★ 上記2の操作中、点のないところをタップすると、新たな点がつくられます。下図はその例で、 $\triangle ABC$  を描画しているところです。



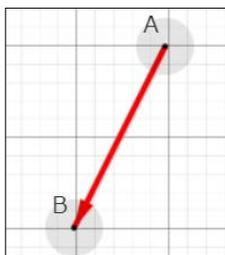
## 6-2 連結図形のプロパティと訂正・削除

- 連結図形の訂正・削除

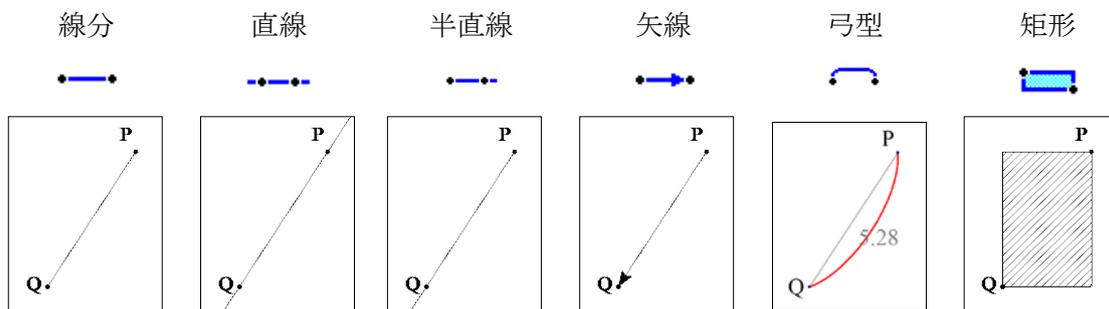
1. 連結図形パネルで、該当する図形をタップ  
または、連結図形上で右タップし、ポップアップメニューから [連結図形のプロパティ] をタップ



2. 図形の種類、色、太さ、線種、領域色、残像を設定  
削除のときは、[削除] をタップ
3. [OK] をタップ



- 図形の種類について (2点を結んでできる図形)



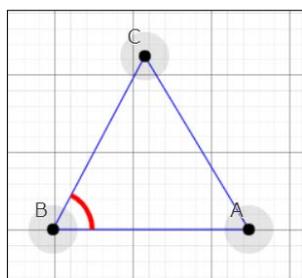
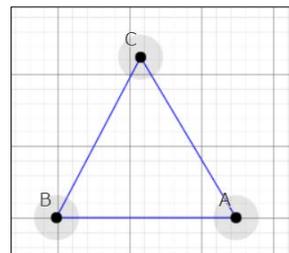
連結図形を構成している点のうちの一つでも削除されると、その連結図形も削除されます。

## 6-3 角の表示

### ● 角の表示

3点 A, B, C について,  $\angle ABC$  を描く方法の説明をします。

1. 点を結ぶボタン  をタップ
2. グラフパネル上で, 点 A, B, C をこの順でタップ
3. [角度] ボタン  をタップ
4. 線色および内部色を設定
5. [OK] をタップ



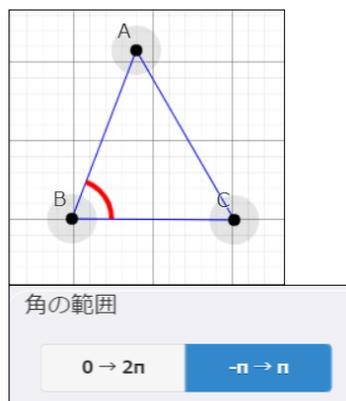
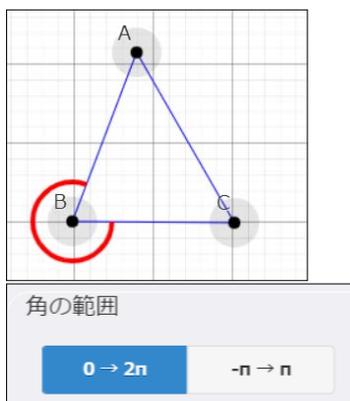
角を表す弧の半径を変えるには, オプションの [グラフ] 設定で, 「角記号の大きさ」を変更します。

角記号の大きさ

### ● 角の向き

角は左回りに表示されます。例えば,  $\angle ABC$  は, 半直線 BA から出発して左回りに半直線 BC に到達するまでの角です。

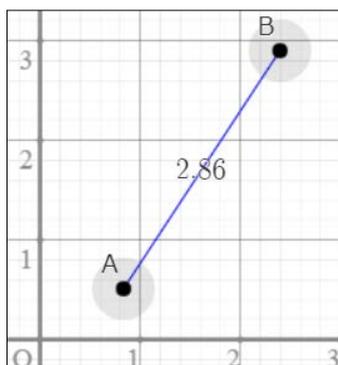
常に  $180^\circ$  以下の角を表示したい場合には, オプションでの [関数] 設定で, 「角の範囲」を  $[-\pi \rightarrow \pi]$  に設定します。



## 6-4 線分の長さや角の大きさの表示

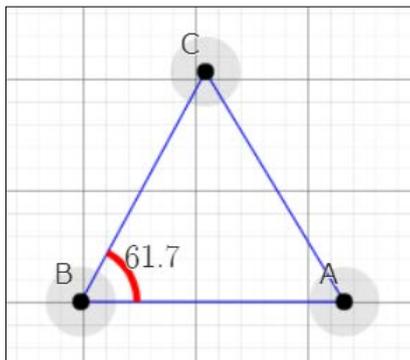
### ● 線分の長さを表示する

1. 連結図形パネルで線分 AB をタップ
2. ラベル表示窓のプルダウンメニューから “ $!{\{AB\}}$ ” を選ぶ
3. ラベル表示位置をタップ



### ● 角の大きさを表示する

1. 連結図形パネルで角 ABC をタップ
2. ラベル表示窓のプルダウンメニューから “ $!{\{arg(A,B,C)\}}3$ ” を選ぶ
3. ラベル表示位置をタップ
4. オプションの [関数] 設定で「角の単位」を [度数法] にする



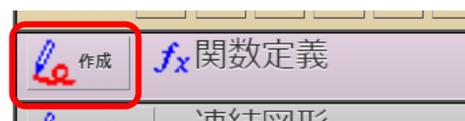
# 第7章 関数の定義と相互参照

## 7-1 定義関数を利用する

同じ関数を繰り返し使うとき、それを  $f(x)$  などと定義しておくことができます。このような関数は5個使うことができます。

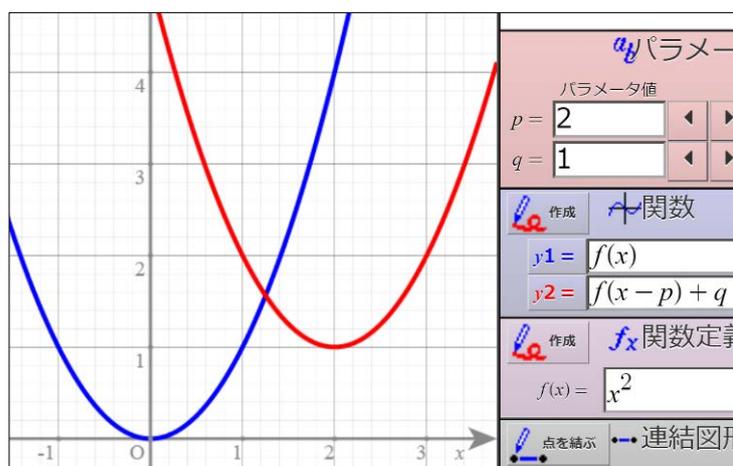
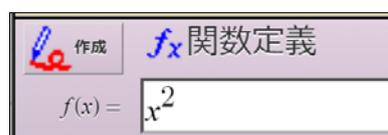
- **定義関数**

1. 関数定義パネルの [作成] をタップ



2. 関数電卓で式入力

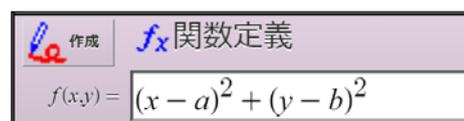
定義関数は、 $f, g, h, f1, \dots, f5$  の8個あります。



2つの関数  $y = f(x)$  ,  $y = f(x-p)+q$  のグラフ

- **2変数関数**

関数に変数  $y$  が含まれているとき、2変数関数として扱われます。このとき関数のラベル部分には、2変数関数であることが  $f(x,y)$  のように表示されます。



2変数関数を参照するときは、 $f(2x, k)$  のように、2個の引数を与えます。なお、第2引数は省略することができ、この場合、 $y=0$  として扱われます。

なお、引数は  $x, y, z, w$  の4つまで利用することができます。すなわち、 $f(x,y,z,w)$  となります。

## 7-2 関数を参照する

### ● 関数の参照

2つの関数

$$y1 = \sin ax, y2 = \cos bx$$

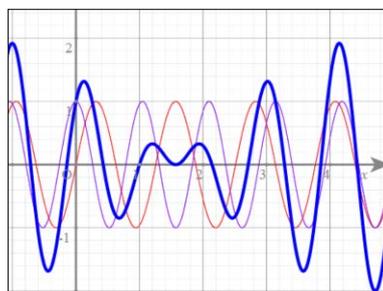
とそれを合成した関数

$$y3 = \sin ax + \cos bx$$

のグラフです。このような場合、

$$y3 = y1 + y2$$

とすることができます。



作成		関数
y1 =	sin	(ax)
y2 =	cos	(bx)
y3 =	y1 +	y2

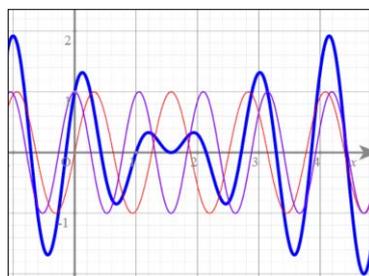
### ● 参照の順序

参照も順序は自由です。

$$y1 = y2 + y3$$

$$y2 = \sin ax, y3 = \cos bx$$

としても、結果は同じです。



作成		関数
y1 =	y2 +	y3
y2 =	sin	(ax)
y3 =	cos	(bx)

## 7-3 グラフをドラッグする 一点の座標の参照

点の座標は関数から参照することができます。

例として、頂点Aのドラッグで動く、2次関数のグラフを描いてみます。

### ● グラフをドラッグする

#### 1. 点Aを打つ

#### 2. 点の座標を使って2次関数のグラフを描く

点Aの座標は、(A.x, A.y)で表されます。

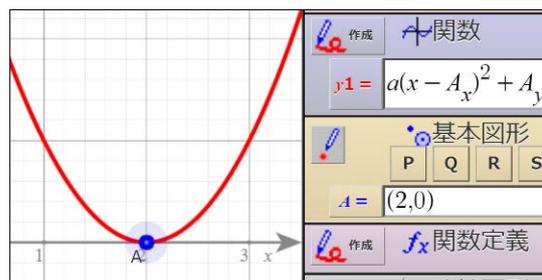
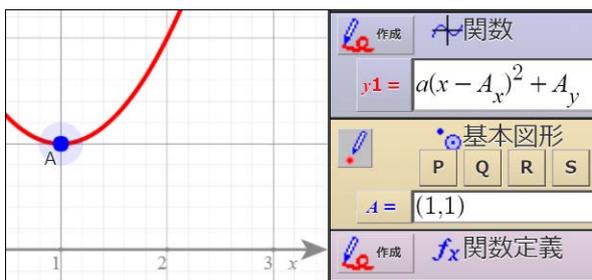
したがって、点Aを頂点にもつ2次関数は、

$$y = a(x - A.x)^2 + A.y \text{ です。}$$

作成		関数
y1 =	a(x - A <sub>x</sub> ) <sup>2</sup> +	A <sub>y</sub>

#### 3. 点Aをドラッグ

点Aをドラッグするとグラフが移動します。



# 第 8 章 ベクトルの利用

## 8-1 点とベクトル

### ● 点とベクトル

三角形 ABC の重心 G は、座標を用いて

$$\left( \frac{A.x+B.x+C.x}{3}, \frac{A.y+B.y+C.y}{3} \right)$$

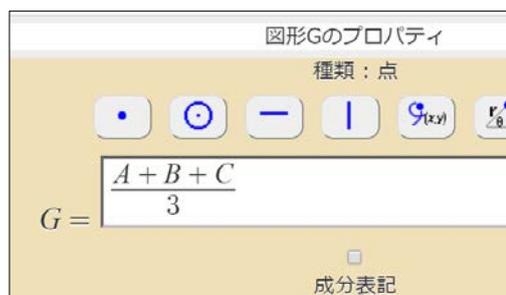
と表されます。これを位置ベクトルで表現すれば、

$$\frac{\vec{OA} + \vec{OB} + \vec{OC}}{3}$$

ですが、同等のことが GRAPES でもできます。

$$\frac{\vec{OA} + \vec{OB} + \vec{OC}}{3} \quad \text{あるいは} \quad \frac{A+B+C}{3}$$

です。GRAPES では、点と位置ベクトルは同一視されています。



★ 点を表す記号は、Windows 版 GRAPES ではローマン体 (ABC...) ですが、GRAPES-light ではイタリック体 (*ABC*...) で表示されます。表示が違っただけで、働きは同じです。

### ● ベクトルと実数

実数からベクトルを作るには、2つの実数の組を括弧でくくります。例えば、x成分が2でy成分が3のベクトルは、

$$(2, 3)$$

と表します。これは、座標の表記と同じです。

逆に、ベクトルが与えられているとき、その成分を求めるには .x や .y を付けます。

$$(2, 3).x = 2$$

$$(2, 3).y = 3$$

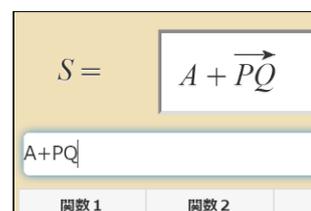
です。



### ● 点と変位ベクトル

基本図形や曲線の点を並べて書くと、変位ベクトルを表します。

例えば、2点 P, Q があるとき、“PQ” は  $\vec{PQ}$  を表します。



## 8-2 座標やベクトルで入力する

ベクトルを用ると、点の位置を  $x$  座標と  $y$  座標に分けずに入力することができます。

- 点の位置をベクトルを用いて入力する

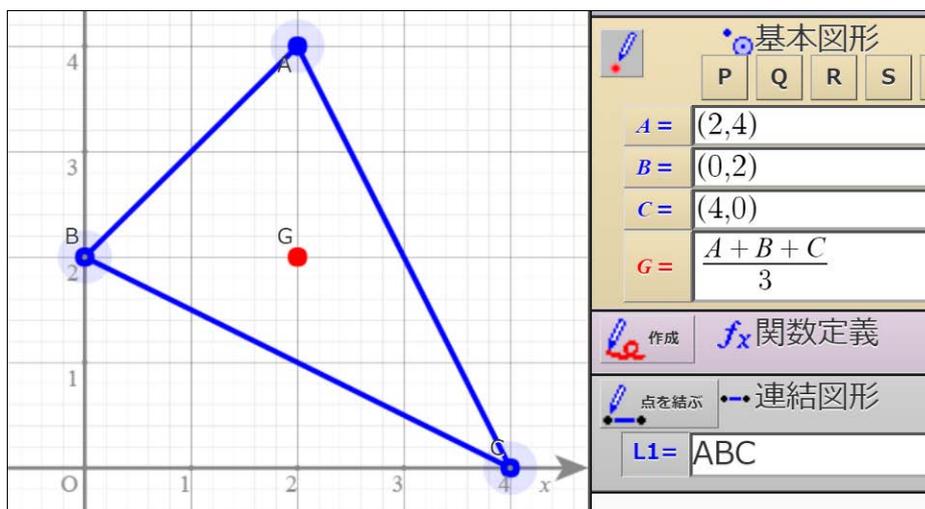
下記の操作に先立って、三角形 ABC を描いておいてください。

1. 基本図形 G を新たに作成し、[点] を選ぶ
2. [成分表示] チェックボックスのチェックを外す  
2つあった式表示窓が1つになります。
3. 式表示部分をタップして、関数電卓で入力



GRAPES は、ベクトルの式を扱うことができません。ベクトルを用いた式の書き方については後で述べます。

5. [OK] をタップ



## 8-3 ベクトルの演算

- ベクトルの加法, 減法と実数倍

ベクトルの加法, 減法, 実数倍については, 数学で用いる標記をそのまま使うことができます。

例:  $P + (-4, 5)$  : 点  $P$  の位置ベクトルとベクトル  $(-4, 5)$  の和

$(2, 3) - (-4, 5)$  : ベクトル  $(2, 3)$  とベクトル  $(-4, 5)$  の差

$3\overrightarrow{PQ}$  : 変位ベクトル  $\overrightarrow{PQ}$  の3倍

$(A + B)/2$  : 2点  $A, B$  の中点 (の位置ベクトル)

- ベクトルの内積

2つのベクトルの内積を求めるには, 演算記号 “ $\cdot$ ” を使います。

例:  $(1, 2) \cdot (3, 4)$  : ベクトル  $(1, 2)$  と  $(3, 4)$  の内積

The screenshot shows a calculator interface with a display area containing  $f() = (1,2) \cdot (3,4)$ . Below the display, the input  $(1,2) \cdot (3,4)$  is visible. A keypad with buttons for function 1, function 2, function 3, and function 4 is shown. The dot product symbol  $\cdot$  is highlighted with a red box.

- 混合演算

括弧を使えば, 複雑な計算にも対応できます。

The screenshot shows a calculator interface with a display area containing  $f() = (\frac{A+B}{2} + (1,2)) \cdot \overrightarrow{PQ}$ . Below the display, the input  $((A+B)/2+(1,2))*PQ$  is visible. The keypad shows function 1, function 2, function 3, and function 4 buttons.

- ベクトルの大きさ

ベクトルの大きさ (長さ) を求めるには, 絶対値記号と同じ記号 “[ ]” を使います。これも, 数学で用いる標記と同じです。

The screenshot shows a calculator interface with a display area containing  $f() = |P + (3,4)|$ . Below the display, the input  $[P+(3,4)]$  is visible. The keypad shows function 1, function 2, function 3, and function 4 buttons. The absolute value brackets [ ] are highlighted with a red box.

- 線分の長さ

“ $[PQ]$ ” と書いた場合, 変位ベクトル  $\overrightarrow{PQ}$  の大きさを表しますが,  $|\overrightarrow{PQ}|$  とは表示されず, 単に  $PQ$  と表示されます。これは線分  $PQ$  の長さを表しています。

The screenshot shows a calculator interface with a display area containing  $f() = \frac{\overrightarrow{PQ}}{PQ}$ . Below the display, the input  $PQ/[PQ]$  is visible.

## 8-4 距離と角度と面積

- 距離を与える関数 len

$\text{len}(P, Q)$  : 2点  $P, Q$  の距離

$\text{len}(P)$  : 点  $P$  の原点からの距離

下記のように座標を使うこともできます。

$\text{len}(P, (2, 3))$  : 点  $P$  と点  $(2, 3)$  の距離

$\text{len}(2, 3)$  : 点  $(2, 3)$  の原点からの距離

他の関数についても同様にして座標を使うことができます。

- 角度を与える関数 arg

$\text{arg}(P, Q, R)$  :  $\angle PQR$  の左回りの大きさ

$\text{arg}(P, Q)$  :  $\angle POQ$

$\text{arg}(P)$  :  $OP$  の偏角

初期状態では、角の単位は弧度法で、範囲は  $0$  から  $2\pi$  までです。角の単位や範囲は、オプションの [関数] 設定で変更することができます。



- 面積を与える関数 areaS

$\text{areaS}(P_1, P_2, \dots, P_n)$  :  $n$  個の点を結ぶ多角形の符号付面積

- 行列式の値

$\text{det}(P, Q)$  : 2つのベクトルが作る行列式の値を与える

関数値は実数です。

## 8-5 位置を与える関数

### ● 線分の分点を与える関数“分点”

関数“分点”は、線分の内分点や外分点を与えます。

分点( $P, Q, m, n$ ): 線分  $PQ$  を  $m:n$  に分ける点

分点( $P, Q$ ) : 線分  $PQ$  の中点

キーボードから入力するときは `mid` を使います。

### ● 図形の交点を与える関数“交点”

交点( $A, B, C, D$ ): 2直線  $AB$  と  $CD$  の交点

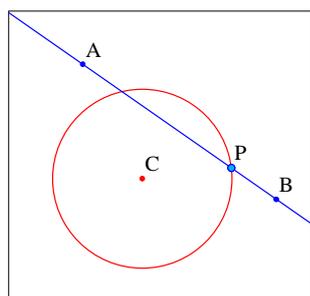
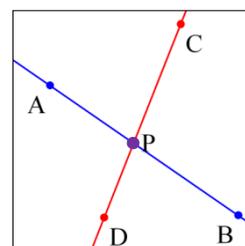
交点( $A, B, C, r$ ): 直線  $AB$  と中心  $C$  半径  $r$  の円の交点

関数“交点”は、2直線の交点を与えます。

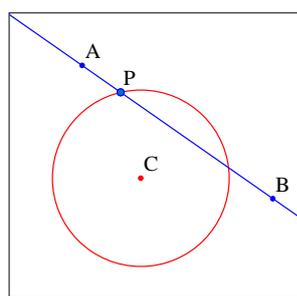
キーボードから入力するときは `intr` を使います。

☆ 直線と円の交点は2つあります。この場合、もうひとつの交点を得るには

交点( $B, A, C, r$ )とします。



交点( $A, B, C, r$ )



交点( $B, A, C, r$ )

交点( $A, a, B, b$ ): 中心  $A$  半径  $a$  の円と中心  $B$  半径  $b$  の円の交点

☆ 2円の交点は2つあります。この場合、もうひとつの交点を得るには、

交点( $B, b, A, a$ )とします。

### ● 垂線の足を与える関数“垂足”

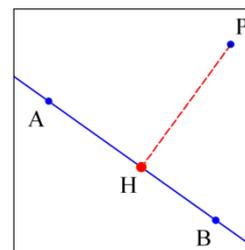
垂足( $P, A, B$ ): 点  $P$  から直線  $AB$  に下ろした垂線の足

関数“垂足”は、点から直線に下ろした垂線の足を与えます。

キーボードから入力するときは `perp` を使います。

★ 垂線の長さを求めるには、“垂長”関数を使います。

垂長( $P, A, B$ ) キーボード入力は `perpL` です。



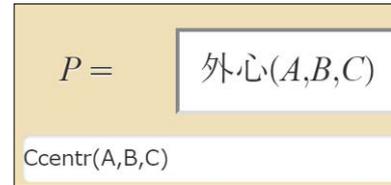
● 三角形の重心, 外心, 垂心, 内心, 傍心を与える関数

重心(A, B, C) : 三角形 ABC の重心

関数“重心”は三角形の重心を与えます。

キーボードから入力するときは **Gcentr** を使います。

他にも, 外心, 内心, 傍心, 垂心などを与える関数があります。



働き	関数名	キー入力名	例	値
重心	重心	Gcentr	重心(A,B,C)	ベクトル
外心	外心	Ccentr	外心(A,B,C)	ベクトル
垂心	垂心	Hcentr	垂心(A,B,C)	ベクトル
内心	内心	Icentr	内心(A,B,C)	ベクトル
傍心	傍心	Ecentr	傍心(A,B,C)	ベクトル
外接円の半径	外径	Crad	外径(A,B,C)	実数
内接円の半径	内径	Irad	内径(A,B,C)	実数

● 回転を与える関数“回転”

回転(A, C, t) : 点 A を点 C の周りに角 t 回転

回転(A, t) : 点 A を原点の周りに角 t 回転

回転(A) : 点 A を原点の周りに 90° 回転

関数“回転”の値はベクトル(すなわち点)です。

キーボードから入力するときは, **rot** を使います。

回転角の単位は初期状態では弧度法です。度数法に変えるには, オプションの [関数] 設定を用います。



● 単位円上の点

roll(t) : 単位円上の点 (cost, sint) を与える

関数値はベクトルです。

極座標で (a, θ) で表される点は, roll 関数を用いれば,

a roll(θ) で表されます。

● 単位ベクトル

unit(P) :  $\overrightarrow{OP}$  方向の単位ベクトル

● 多角形の辺上の点

多角形( $k, P_0, P_1, \dots, P_{n-1}$ ) :  $n$  角形の辺上の点。  $0 \leq k < n$  で多角形を一周する。 ( $n \leq 20$ )

キーボードから入力するときは, polygon を使います。

$k$  が整数のとき, 多角形( $k, P_0, P_1, \dots, P_{n-1}$ ) =  $P_k$

非整数のとき, 辺上の点を表す。

$k = k_0 + ln$  ( $l$  は整数) のとき,

多角形  $c$  多角形( $k_0, P_0, P_1, \dots, P_{n-1}$ )

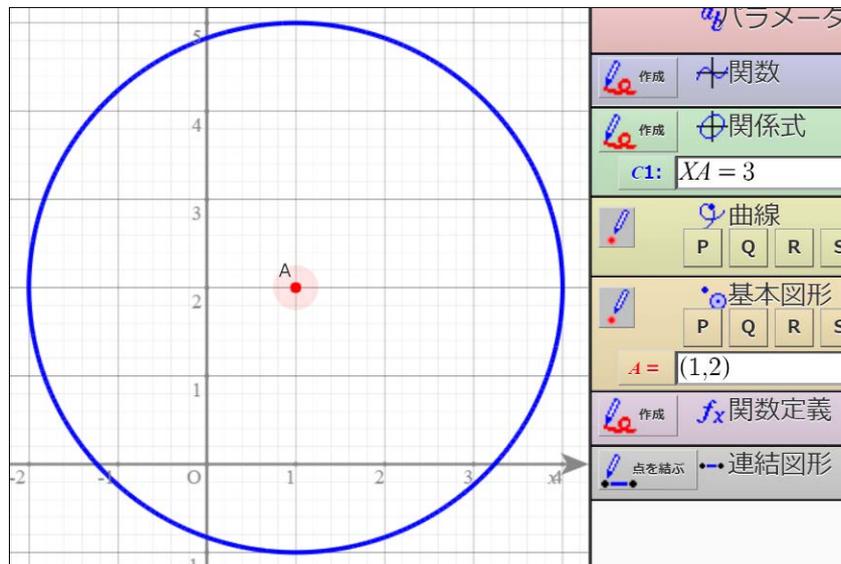
とくに,

多角形( $n, P_0, \dots, P_{n-1}$ ) = 多角形( $0, P_0, \dots, P_{n-1}$ ) =  $P_0$

● X は 点(x, y) を表す

関係式や定義関数の中では, 変数  $x, y$  を使うことができます。点 X はこれら変数を組にした点  $(x, y)$  を表します。

例えば, 関係式で  $XA = 3$  とすれば, 中心 A 半径 3 の円を表します。



## 8-6 定義関数とベクトル

- ベクトル（点）を引数として扱う

例えば,

$$f(x, y) = 2x + 3y$$

であるとしましょう。このとき,

$$f(4, 5) = 2 \times 4 + 3 \times 5$$

ですが,  $P = (4, 5)$  のときも, やはり,

$$f(P) = 2 \times 4 + 3 \times 5$$

となります。

- ベクトル  $X$  は  $(x, y)$  を表す

例えば, 関数  $f(x, y) = 2x + 3y$  であれば,  $2x + 3y = (2, 3) \cdot (x, y)$  ですから,

$$f(x, y) = (2, 3) \cdot X$$

と表すことができます。

同様に, ベクトル  $Y$  は  $(z, w)$  を表します。

- ベクトルを関数値として返す

定義関数の値は, ベクトルでも構いません。

例えば,

$$f(x) = xA + (1-x)B$$

とすれば, 関数  $f(x)$  は線分  $AB$  の内分点を与えます。

- ベクトル（点）の変換

定義関数を利用して, 平面上の点の変換をひとつの関数で表現することができます。

例えば,

$$f(x, y) = (ax + by, cx + dy)$$

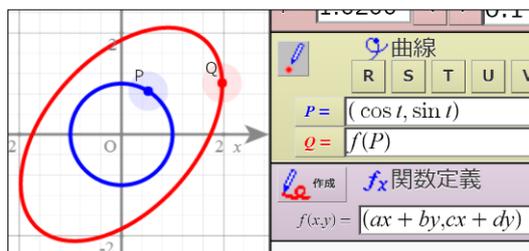
は, 1次変換です。

GRAPES の定義関数では  $f(x, y) = f((x, y))$  なので,

$$Q = f(P)$$

とすれば, 点  $Q$  は点  $P$  の像を表します。

同様に,  $f(x, y, z, w) = f((x, y), (z, w))$  です。



# 第9章 関数を使いこなす

## 9-1 グラフの詳細設定

### ● [グラフ] 設定の表示

メニューボタン をタップし、[オプション] から [グラフ] を選びます。

### ◆ 連結図形の表示は端点の表示に依存

これをチェックすると、連結図形を構成する点のうちひとつでも非表示のものと、その連結図形は表示されなくなります。

### ◆ 削除に確認を求める（無効）

### ◆ 図形内部の残像を残す（現在、無効）

### ◆ 左ボタン up 時のみ残像を残す

基本図形が残像を残す設定になっているとき、ドラッグ中にも残像が残りますが、これをチェックすることで、ドラッグ終了後のみ残像が残るようになります。

### ◆ 角記号の大きさ

連結図形の「角」や「直角」を表示する際の弧の半径を指定します。

### ◆ 関係式の計算密度

一般の関係式では、グラフ表示エリア内のすべての点について関数の値を調べ、グラフを描いています。これは関数によってはかなりの時間を要します。そのために、点の密度を落として計算するようにしています。

### ◆ 関数の描画方法

関数のグラフでは、 $x$  に対する  $y$  の値を計算してグラフを描いています。このとき、これらの点をすべて線で結んで連続したグラフにするか、そのまま点をプロットするか、一定の条件のもとでのみこれらを結ぶようにするかを選択することができます。

### ◆ 残像の濃さ

残像や軌跡の色は、元になるグラフ色を薄くしたものですが、濃さを変えることができます。



## 9-2 関数の詳細設定

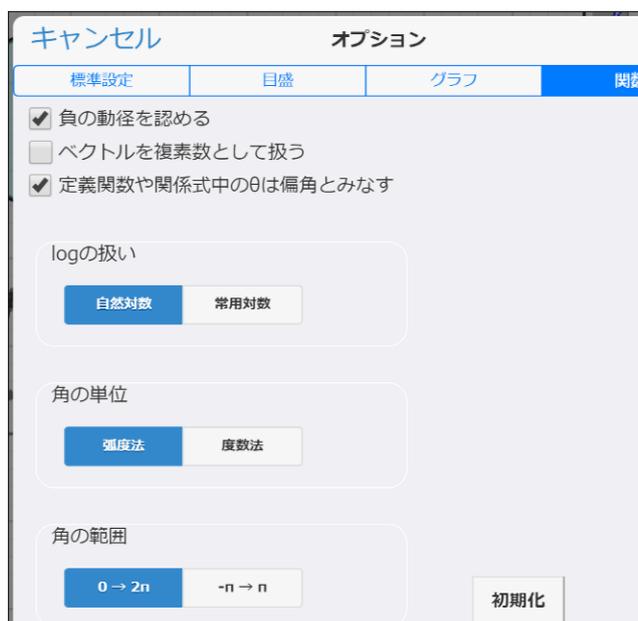
### ● [関数] 設定の表示

メニューボタン をタップし、[オプション] から [関数] を選びます。

#### ◆ 負の動径を認める

極方程式の動径や円の半径は 0 以上の数であるのが通常ですが、負の動径や半径を認めると都合のよいことが多くあります。

初期設定では、負の動径や半径を認めるようになっていますが、これを変更することができます。



#### ◆ ベクトルを複素数として扱う（現在無効）

#### ◆ 定義関数や関係式中の $\theta$ は偏角とみなす

関係式や定義関数中では、 $\theta$  は点  $(x, y)$  についての偏角を表します。しかし、 $\theta$  は関数や基本図形ではパラメータとして働きます。もし、関係式や定義関数でも $\theta$ をパラメータとして扱うときには、このチェックを外します。

#### ◆ log の扱い

関数 log について、これを自然対数として扱うか、あるいは常用対数として扱うかを選択することができます。初期設定は自然対数です。

関数電卓にはありませんが、自然対数を表す関数  $\ln$  (エルエヌ) も使うことができます。

#### ◆ 角の単位

角を弧度法で扱うか度数法で扱うかを選択することができます。これは、三角関数や逆三角関数、および  $\arg$  関数に影響します。初期設定は弧度法です。

#### ◆ 角の範囲

逆三角関数や  $\arg$  によって得られる角の範囲を与えます、初期設定は  $0 \leq \theta < 2\pi$  です。

また、この設定は、連結図形の「角」の表示にも影響します。角の範囲が  $-\pi \leq \theta < \pi$  のとき、「角」は常に劣弧（小さい方の角）を表示します。

---

## 9-3 GRAPES の関数

---

### ● 左関数

通常関数のことで、関数の右に引数がきます。三角関数や対数関数では、引数の範囲を表す括弧を省略することができます。例えば、 $\sin(2x)$  を  $\sin 2x$  と書いても構いません。他の関数では、引数の範囲を括弧記号で明示する必要があります。

- ◆  $\sin x, \cos x, \tan x$  : 三角関数

- ◆  $A \sin x, A \cos x, A \tan x$  : 逆三角関

三角関数と逆三角関数について、角の単位は初期設定では弧度法。弧度法か度数法かの選択はオプションの [関数] 設定で行います。

- ◆  $\exp(x)$  :  $e^x$

- ◆  $\log(x)$  : 自然対数または常用対数

初期設定では自然対数。変更はオプションの [関数] 設定で行います。

- ◆  $\log(a, x)$  :  $a$  を底とする  $x$  の対数

- ◆  $\ln(x)$  : 自然対数

- ◆  $\sinh x, \cosh x, \tanh x$  : 双曲線関数

- ◆  $\sqrt{x}, \sqrt[3]{x}$  : 平方根, 立方根 (キー入力は, Sqrt, Cbrt)

- ◆  $\text{int}(x)$  : 整数部分 (ガウス記号と同等)

- ◆  $\text{round}(x)$  : 四捨五入

- ◆  $\text{frc}(x)$  : 小数部分  $\text{frc}(x) = x - \text{int}(x)$

- ◆  $|x|$  もしくは  $\text{abs}(x)$  : 絶対値 (| | のキー入力は [ ] )

- ◆  $\text{sign}(x)$  : 符号

- ◆  $\text{rnd}(x)$  : 0 以上  $x$  未満の整数の乱数 ( $x=1$  の場合は, 0 から 1 の間の実数の乱数)

- ◆  $\Gamma(x), B(x, y)$  : ガンマ関数とベータ関数

- ◆  $f(x), g(x), h(x), f1(x) \sim f5(x)$  : 定義関数

- ◆  $f'(x) \sim f5'(x), f''(x) \sim f5''(x)$  : 定義関数の導関数および 2 次導関数

- ◆  $F(x) \sim F5(x)$  : 定義関数の不定積分

- ◆  $\text{nCr}(x, y)$  : 二項係数

- ◆  $\text{gcd}(x, y), \text{lcm}(x, y)$  : 最大公約数と最小公倍数

- ◆  $f(x, y), f(x, y, z), f(x, y, z, w)$  : 定義関数 ( $g, h, f1 \sim f5$  も同様)

- ◆  $\text{Max}(x_1, \dots, x_n), \text{Min}(x_1, \dots, x_n)$  : 最大値, 最小値

引数の個数は 1 以上 20 以下で、カンマで区切って記述する。

- ◆  $\text{case}(p_1, x_1, \dots, p_n, x_n), \text{case}(p_1, x_1, \dots, p_n, x_n, x_{n+1})$  : 場合分け関数

条件  $p_1 \dots p_n$  のうちで最初に真であるものを  $p_m$  とするとき値は  $x_m$ , 真のものが無いときは Error (あるいは  $x_{n+1}$ ) となる。

---

## 9-4 特殊な関数

---

- 右関数

引数の後に関数記号がある関数です。

- ◆  $x!$  : 階乗
- ◆  $x^\circ$  : 度を弧度に変換 (“°” のキー入力は degree)
- ◆  $.x, .y$  : ベクトルの  $x$  成分および  $y$  成分
- ◆  $.r$  : 円 (基本図形) の半径

- 引数なし関数

- ◆  $y1 \sim y20$  : 関数の値 (関数や関係式の中からのみ参照できます)
- ◆  $P, Q$  など : 点
- ◆  $O$  : 原点
- ◆  $X$  : 変数の組  $(x, y)$
- ◆  $r, \theta$  :  $r = \text{len}(x, y)$ ,  $\theta = \text{arg}(x, y)$  (関係式や定義関数の中でのみ使用可)

- 関数やパラメータ文字を引数に持つ関数

- ◆  $\text{sol}(f(x) = g(x), n)$  : 4 次以下の方程式  $f(x) = g(x)$  の解のうち、実数部分の小さい順に左から  $n$  番目 ( $n = 1, 2, 3, 4$ ) のものを与える。 $n$  を省略した場合、 $n = 1$  となる。
- ◆  $\text{cof}(n, f(x))$  :  $f(x)$  が 6 次以下の整式のとき、 $n$  次の係数を与える。

例 :  $\text{cof}(2, (x-2)^3) = -6$

- ◆  $\Sigma(\text{パラメータ}, \text{初期値}, \text{終了値}, \text{式})$  : 級数の和 (“ $\Sigma$ ” のキー入力は Sum)

例 :  $\sum_{k=1}^5 k^2$  ならば、 $\text{Sum}(k, 1, 5, k^2)$

初期値や終了値に変数  $x$  を含んだ式を代入することはできません。

- ◆  $\Pi(\text{パラメータ}, \text{初期値}, \text{終了値}, \text{式})$  : 級数の積 (“ $\Pi$ ” のキー入力は Product)

初期値や終了値に変数  $x$  を含んだ式を代入することはできません。

- ◆  $\int(\text{積分変数}, \text{下限}, \text{上限}, \text{式})$  : 定積分 (“ $\int$ ” のキー入力は Intgr)

例 :  $\int_0^\pi \sin 2x dx$  ならば、 $\text{Intgr}(x, 0, \pi, \sin 2x)$

積分変数にはパラメータや  $x$  を使います。

---

## 9-5 ベクトル（点）に関する関数

---

### ● 実数値関数

- ◆  $\text{path}(x, P_1, \dots, P_n)$  : 与えられた  $n$  個の点を通る ( $n-1$ ) 次関数
- ◆  $|P|$  もしくは  $\text{len}(P)$  : ベクトルの大きさ (ノルム記号 “ $|$ ” のキー入力は  $[]$ )
- ◆  $\text{len}(P_1, P_2)$  : 2 点間の距離
- ◆  $\text{arg}(P)$  : 偏角 (角の範囲は [関数設定] で設定 ( $0 \leq \text{arg} < 2\pi, -\pi \leq \text{arg} < \pi$ ) します)
- ◆  $\text{arg}(P_1, P_2)$  : 角  $\angle P_1OP_2$   
 $\text{arg}(P_1, P_2, P_3)$  : 3 点の作る角  $\angle P_1P_2P_3$
- ◆  $\text{det}(a, b, c, d)$  : 行列式の値  $ad - bc$   
 $\text{det}(P, Q)$  : 行列式の値  $P.x \cdot Q.y - P.y \cdot Q.x$
- ◆ 垂長( $P, A, B$ ) : 点  $P$  から直線  $AB$  に下ろした垂線の長さ (“垂長” のキー入力は  $\text{perpL}$ )
- ◆ 外径( $A, B, C$ ) : 三角形  $ABC$  の外接円の半径 (“外径” のキー入力は  $\text{Crad}$ )
- ◆ 内径( $A, B, C$ ) : 三角形  $ABC$  の内接円の半径 (“内径” のキー入力は  $\text{Irad}$ )
- ◆ 面積 ( $P_1, \dots, P_n$ ) : 多角形の符号付き面積 (“面積” のキー入力は  $\text{areaS}$ )

### ● ベクトル値関数

- ◆  $\text{roll}(t)$  : 単位円上の点  $(\cos t, \sin t)$  を与える。
- ◆  $P(n)$  :  $n$  番目の点の座標 (点の番号とは,  $P, Q, \dots, V, A, \dots, N$  に対して,  $1, 2, 3, \dots, 21$ )  
この関数は, ステッカーやスクリプトの中だけで使うことができます。
- ◆  $\text{unit}(P)$  :  $\overrightarrow{OP}$  方向の単位ベクトル
- ◆ 重心( $A, B, C$ ) : 三角形  $ABC$  の重心 (“重心” のキー入力は  $\text{Gcentr}$ )
- ◆ 外心( $A, B, C$ ) : 三角形  $ABC$  の外心 (“外心” のキー入力は  $\text{Ccentr}$ )
- ◆ 垂心( $A, B, C$ ) : 三角形  $ABC$  の垂心 (“垂心” のキー入力は  $\text{Hcentr}$ )
- ◆ 内心( $A, B, C$ ) : 三角形  $ABC$  の内心 (“内心” のキー入力は  $\text{Icentr}$ )
- ◆ 傍心( $A, B, C$ ) : 三角形  $ABC$  の傍心 (“傍心” のキー入力は  $\text{Ecentr}$ )  
他の傍心は, 傍心( $B, C, A$ ), 傍心( $C, A, B$ )
- ◆ 交点( $A, B, C, D$ ) : 2 直線  $AB, CD$  の交点 (“交点” のキー入力は  $\text{intr}$ )
- ◆ 交点( $A, B, C, r$ ) : 直線  $AB$  と中心  $C$  半径  $r$  の円との交点  
2 つある交点のもう一方は, 交点( $B, A, C, r$ )
- ◆ 交点( $A, a, B, b$ ) : 中心  $A$  半径  $a$  の円と中心  $B$  半径  $b$  の円の交点  
2 つある交点のもう一方は, 交点( $B, b, A, a$ )
- ◆ 垂足( $P, A, B$ ) : 点  $P$  から直線  $AB$  に下ろした垂線の足 (“垂足” のキー入力は  $\text{perp}$ )

- ◆ 分点(A, B) : 線分 AB の中点 (“分点” のキー入力は mid)
  - 分点(A, B, m, n) : 線分 AB を  $m:n$  に分ける点
- ◆ 回転(P) : 点 P を原点を中心に  $90^\circ$  回転した点 (“回転” のキー入力は rot)
  - 回転(P, t) : 点 P を原点を中心に角  $t$  回転した点
  - 回転(P, C, t) : 点 P を点 C を中心に角  $t$  回転した点
- ◆ 多角形( $t, P_1, P_2, \dots, P_n$ ) :  $n$  角形の辺上の点。  $0 \leq t < n$  で多角形を一周する。 ( $n \leq 20$ )
  - (“多角形” のキー入力は polygon)
- ◆ proj( $x, y, z, s, t, d$ ) : 空間の点  $(x, y, z)$  を平面に射影したときの座標を与える
  - $s, t, d$  は, 原点から視点までの平面角, 立体角, 距離
  - $d=0$  もしくは  $d$  を省略したときは, 平行投影になります。
- ◆ +, -, ×, / : 四則演算
- ◆ · : 実数や複素数の積, ベクトルの内積
- ◆ ^ : 実数乗
  - $x^y = x^y$  です。
  - $x < 0$  の場合,  $y$  が整数のときだけ値を持ちます。
  - 三角関数の直後に付けた場合, 関数値の冪と解釈されます。(例:  $\sin^n x = (\sin x)^n$ )
- ◆  $x \bmod y$  : 整数剰余
- ◆  $>, \geq, <, \leq, =, \neq$  : 比較演算子
  - これらは, true のとき値 1 をとり, false のときにはエラー値を返します。
  - 関数の値がエラー値の場合, グラフは描画されません。
  - 例えば,  $y = x(x \geq 0)$  では,  $x \geq 0$  の範囲だけグラフが描かれます。
- ◆ and, or, not : 論理演算子
  - これらは, true のとき値 1 をとり, false のときにはエラー値を返します。
- ベクトルの演算子
  - ◆ +, - : ベクトルの加法, 減法
  - ◆ · : ベクトルの内積 (キー入力 “\*”)
  - ◆ × : 実数×ベクトル (×は省略可) (キー入力は “&”)
  - ◆ / : ベクトル/実数
- パラメータ
  - ◆  $a, b, c, d, k, m, n, p, q, s, t, u, v, \theta$
  - $\theta$  は, 関係式や定義関数では偏角を表す関数として扱われる。

- 変数

- ◆  $x, y, z, w$  :  $z, w$  は, 定義関数やスクリプトでのみ使用可能
- ◆  $\theta$  : 極方程式の偏角 (関係式および定義関数でのみ使用可能)
- ◆  $X$  : 点 $(x, y)$ を表す。
- ◆  $Y$  : 点 $(z, w)$ を表す。(定義関数やスクリプトでのみ使用可能)

- 定数

- ◆  $e$  : 自然対数の底
- ◆  $\pi$  : 円周率

---

## 9-6 評価の優先順位

---

- 優先順位

- ◆ 括弧に囲まれた部分は、先に評価される。
- ◆ 関数や演算子は、次の優先順位で評価される。
  1. 右関数
  2. 冪
  3. 乗算記号省略の乗算
  4. 左関数
  5. 乗除算
  6. 加減算
  7. 比較演算, 論理演算 (and は or に優先する)
- ◆ 上記以外は、式の左から順部に評価される。

- 引数が括弧で囲まれていない関数について

次の関数については、引数を括弧で囲む必要がありません。

$$\sin, \cos, \tan, \operatorname{Asin}, \operatorname{Acos}, \operatorname{Atan}, \log, \sqrt{\quad}, \sqrt[3]{\quad}$$

これら関数の引数の範囲は次のように定められています。

- ◆ 四則演算子, 論理演算子, 比較演算子, 空白, 左関数の何れかが現れるまで。

例:  $\sin a \cos x^2 + 1 = (\sin a)(\cos x^2) + 1$

$$\sin a(x+1)^2 + 1 = \sin(a(x+1)^2) + 1$$

$$\sin a (x+1)^2 + 1 = (\sin a)(x+1)^2 + 1$$

- ◆ 三角関数の直後に冪がある場合、それらはまとめて一つの関数とみなされる。
- ◆ 右関数および冪の引数は、直前の文字, 値, 括弧のみである。

例:  $2n! = 2(n!), 2a x^2 = 2a(x^2)$

☆ 優先関係がわかりにくい場合には、括弧をつけるか、スペースで区切って記述します。

例:  $(\sin a)x$ ,  $\sin a x$  など

## 9-7 導関数と不定積分

GRAPES で微分や積分を使うには、対象となる関数を定義関数に登録し、次いでその導関数や不定積分を利用します。

### ● 導関数の利用

#### 1. 微分したい関数を定義関数に登録する

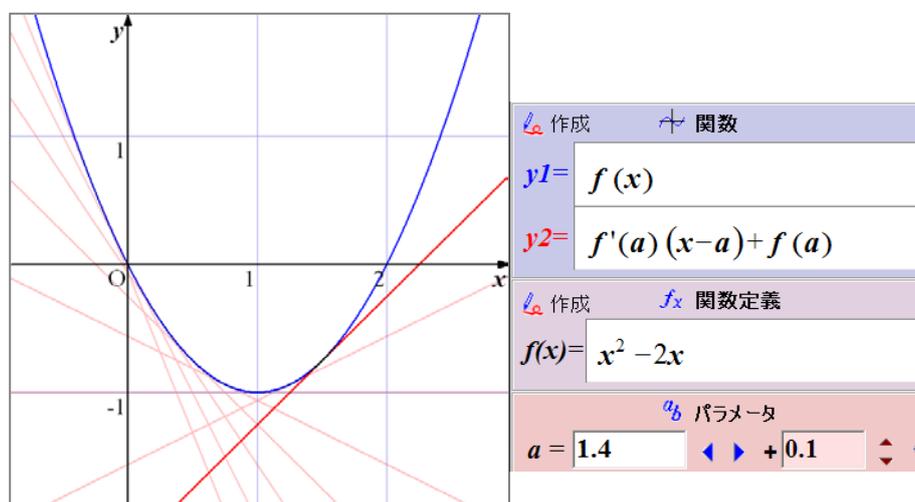
微分して使いたい関数を定義関数に登録します。

例えば、 $f(x) = x^2 - 2x$  などとします。

#### 2. 定義関数の導関数を利用してグラフの式を書く

定義関数の導関数  $f'$ ,  $g'$ ,  $h'$  など使って式を書きます。

例えば、 $y2 = f'(a)(x-a) + f(a)$  などとします。



☆ 2次導関数まで利用することができます。

### ● 不定積分の利用

#### 1. 積分したい関数を定義関数に登録する

例えば、 $f(x) = x^2 - 2x$  などとします。

#### 2. 定義関数の積分関数を利用してグラフの式を書く

定義関数の積分関数  $F, G, H$  などを使って式を書きます。

$F(x)$  は、 $\int_0^x f(t) dt$  を表しています。

注意：積分関数  $F, G, H$  などを関数や関係式内に記述するのは、避けてください。正しい値を表示しないことがあります。