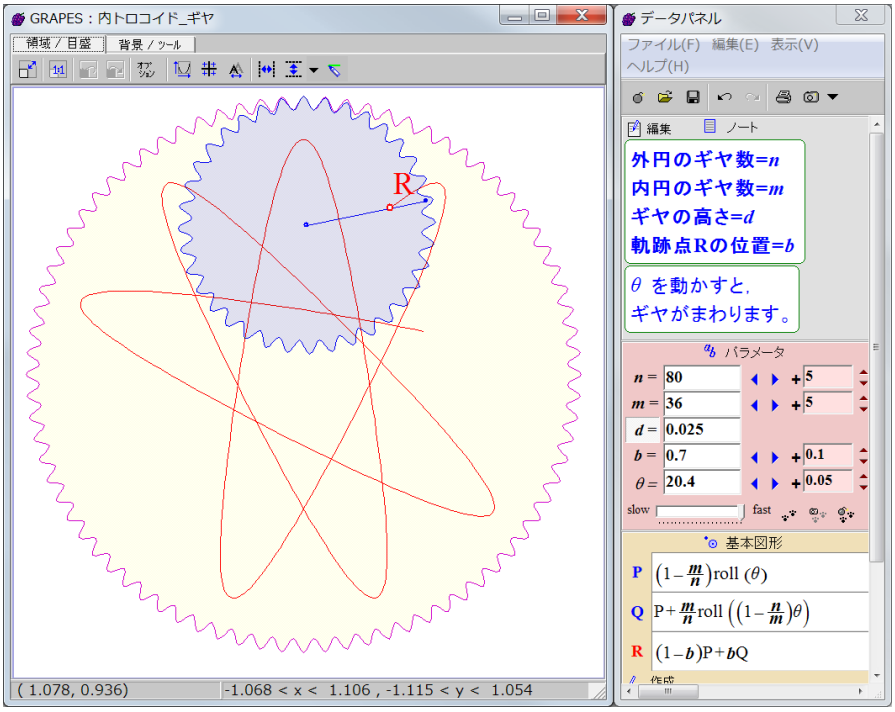
GRAPES USER'S MANUAL

GRAPES Ver 7.17



TOMODA Katsuhisa

（友田　勝久）

Aug. 2016

確認事項

動作環境・著作権・最新版

- 動作環境

a. OS

Windows Vista/7/8/10

Windows 2000/XP については動作確認をしていません。(6.91 をお使いください)

Windows95/98/Me については，サポートの対象外です。これらの OS については，Ver6.16 以前のものを利用していただくのが確実です。

Windows8 への対応については，15-6「windows8 への対応について」をご覧ください。

b. モニター

XGA（1024×768） 以上

65000 色（16 ビットカラー）以上

- 著作権について
  - GRAPES はフリーソフトウェアです。複写・配布・使用を自由に行うことができます。ただし，商業目的で GRAPES を配布されるときは，ご一報をお願いします。
  - GRAPES およびこのマニュアルの著作権は，友田勝久に帰属します。
  - GRAPES を用いて作成したデータや画像の著作権は，そのデータの作成者に帰属します。

- 利用上の留意事項
  - このソフトの利用によって生じるいかなる結果についても，製作者は責任を負いません。
  - バグに関しては解消に努めていますが，対処できないこともあります。

- 最新版について

最新版は下記の URL にあります。

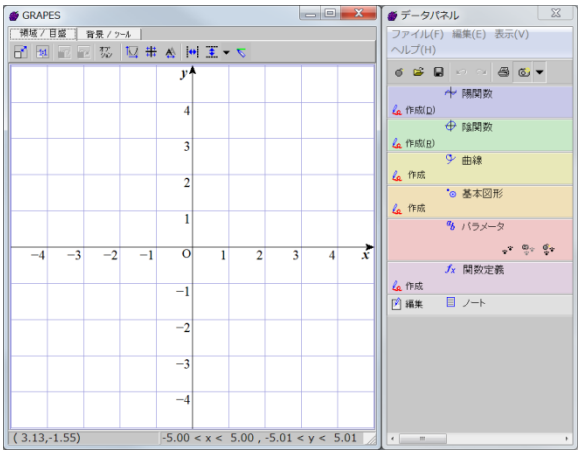
http://www.osaka-kyoiku.ac.jp/~tomodak/grapes/

第１章　GRAPES の基本

1-1 GRAPES の外観

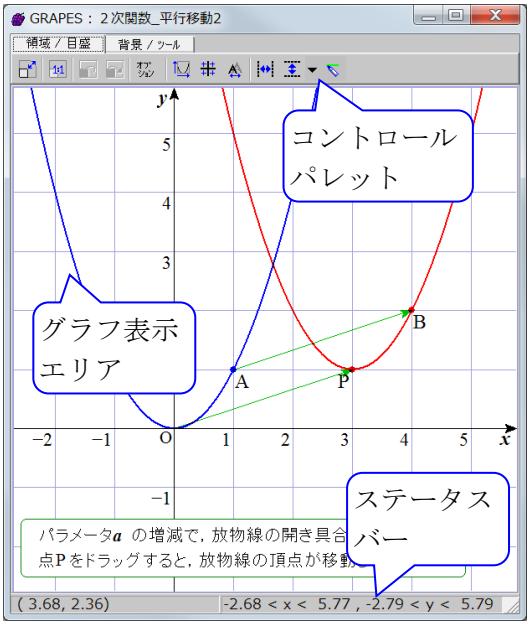
- **グラフウィンドウとデータパネル**

GRAPES は，2 つのウィンドウ，グラフウィンドウとデータパネルで構成されています。グラフウィンドウでは，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。



- **グラフウィンドウ**

グラフウィンドウは上部から，コントロールパレット，グラフ表示エリア，ステータスバーによって構成されています。



- **グラフ表示エリア**

グラフ表示エリアには，データパネルで定めた関数のグラフが表示されます。

ここでのマウス操作によって，表示領域の変更，点など基本図形の移動，ステッカーの移動のほか，右クリックによるポップアップメニューが利用できます。

- **グラフウィンドウのサイズ**

グラフウィンドウのコーナーや枠をドラッグして，ウィンドウサイズを変更すると，グラフ表示エリアのサイズが変わります。

- **コントロールパレット**

コントロールパレットには，表示領域を設定する**領域パレット**をはじめ，**目盛パレット**，**背景パレット**，そして，グラフを調べるためのツールのある**ツールパレット**があります。

- **ステータスバー**

下部のステータスバーには，マウスカーソル位置の座標のほか，グラフの表示領域や操作のヒントが表示されます。

- **データパネル**
  - **メニューバー**

    ［ファイル］,［編集］,［表示］,［ヘルプ］の４つのメニューがあります。
  - **ツールバー**

    ファイル操作を中心として，よく使う７つの機能のボタン,［プロジェクトの初期化］,［ファイルを開く］,［上書き保存］,［UNDO］,［REDO］,［印刷］,［画面コピー］がまとめられています。
  - **陽関数エリア**

    $y = f(x)$ の形で表された関数のグラフを扱います。
  - **陰関数エリア**

    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。
  - **曲線エリア**

    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。
  - **基本図形エリア**

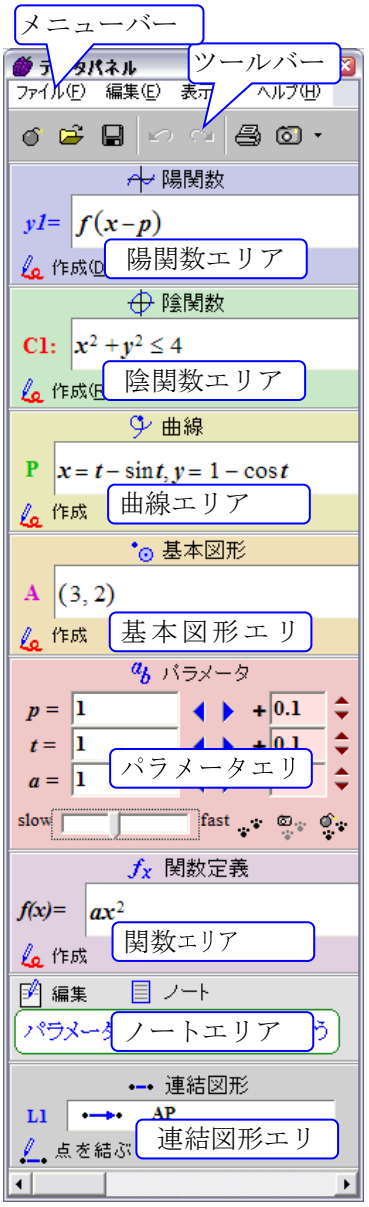
    点，円，水平線，垂直線を扱います。
  - **パラメータエリア**

    パラメータや残像の操作を行います。
  - **関数エリア**

    関数を定義するとき，ここに式を書きます。
  - **ノートエリア**

    画面に説明を表示するステッカー，プログラムの一種であるスクリプト，表形式のデータテーブル，テキストを記録するノートをここで編集します。
  - **連結図形エリア**

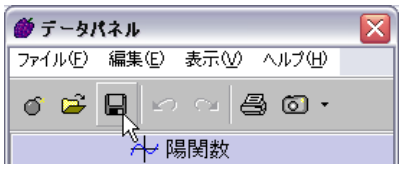
    線分や多角形などの連結図形の設定情報が表示されます。



1-2 プロジェクトの保存と読み出し

- **プロジェクト**

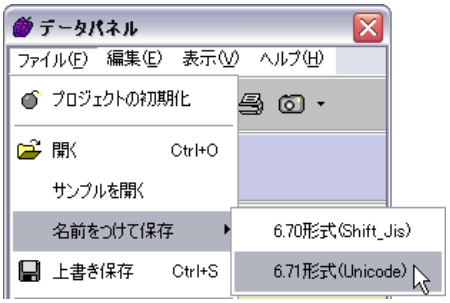
  GRAPES で作成されたデータをプロジェクトといいます。ひとつのプロジェクトには，グラフの方程式，パラメータ値，残像，表示エリア，目盛設定など，GRAPES で作ったデータや設定情報がすべて含まれています。



- **新規プロジェクトの保存**

  ツールバーの上書き保存ボタン をクリック

  または，［ファイル］メニューの［上書き保存］をクリック



- **別名で保存**

  ［ファイル］メニューの［名前を付けて保存］

  ［6.70 形式(Shift_Jis)］と［6.71 形式(Unicode)］のいずれかを選びます。Unicode で保存することにより，外国語版とのファイル互換性が保たれます。

- **プロジェクトの読み出し（ファイルを開く）**

  開くボタン をクリック

  または，［ファイル］メニューの［開く］をクリック

- **サンプルを開く**

  ［ファイル］メニューの［サンプルを開く］をクリック

  GRAPES 本体と同じフォルダにサンプルフォルダがない場合，このメニューを選択することはできません。

GRAPES プロジェクトファイルとは

GRAPES のプロジェクトは，拡張子“.gps”をつけて保存されます。GRAPES プロジェクトファイルは，GRAPES を実行するにあたって必要なすべてのデータを含んでいます。一方，GRAPES の残像データは，プロジェクトファイルと同名で，拡張子“.gpp”をつけて保存されます。残像ファイルを失ってもプロジェクトファイルがあれば，実行に支障はありません。

GRAPES のバージョンアップの際には，プロジェクトファイルの上位互換性は保たれますから，今まで作ったファイルはそのまま利用することができます。

1-3 プロジェクトの初期化と既定値の変更，ファイルの関連付け

- **プロジェクトの初期化**

プロジェクトの初期化ボタン をクリック

または，［ファイル］メニューの［プロジェクトの初期化］をクリック

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。すでに保存されているデータが消されることはありません。

- **既定値の変更**

起動時や初期化時には，既定値に初期化されます。

1. **既定値にしたい状態にする。**

   既定値として変更可能なデータは，表示領域，ウィンドウの広さ，目盛りに関する諸設定など，オプション設定のほとんどが含まれています。

2. **［編集］メニューの［環境設定］から［今の状態を既定値に］をクリック**

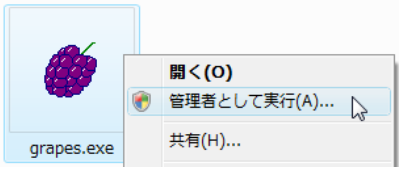
- **既定値の初期化**

［編集］メニューの［環境設定］から［既定値の初期化］をクリック

ユーザーが定めた既定値を，もとの状態に戻します。

- **ファイルの関連付け**

ファイルの関連付けを行うと，GRAPES プロジェクトファイル“*.gps”のダブルクリックだけで，GRAPES が起動し，そのファイルが開きます。



1. **Windows 10/ 8/ 7/ Vista では，GRAPES をいったん終了し，アイコンを右クリックして［管理者として実行］を選んで起動**

   Windows XP/2000 では，通常起動します。

2. **［編集］メニューの［環境設定］から［ファイルの関連付け］をクリック**

GRAPES プロジェクトファイルのアイコンが GRAPES を表す葡萄アイコンに変更されます。ただし，アイコンの変更は，再起動後に有効になる場合があります。

第２章　関数のグラフ

2-1 関数グラフの作成・追加

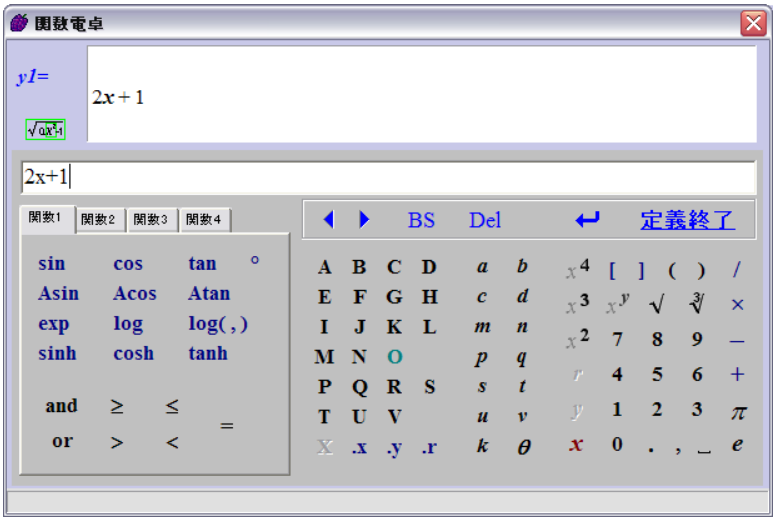
- 関数のグラフの追加・作成

1. 陽関数エリアの［作成］をクリック

2. 関数電卓で式入力

　キーボードもしくは関数電卓のボタンで入力します。

　入力を終えたら，［定義終了］をクリックします。

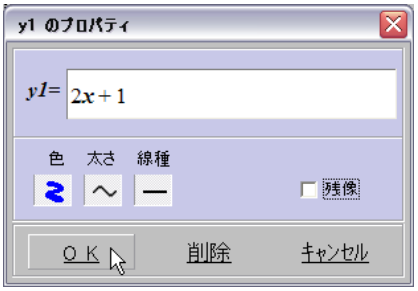


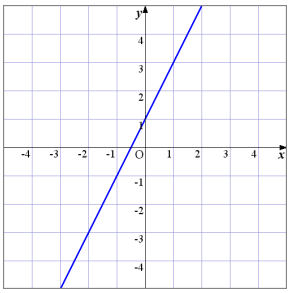
3. グラフスタイルの設定

　関数グラフのプロパティウィンドウが表示されます。

　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。

　設定を終えたら［OK］をクリックします。



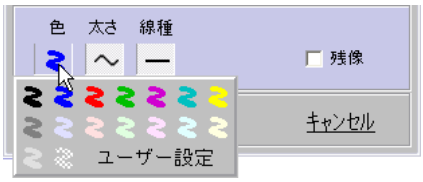


関数のグラフは，２０個まで描くことができます．

2-2 関数グラフのプロパティ

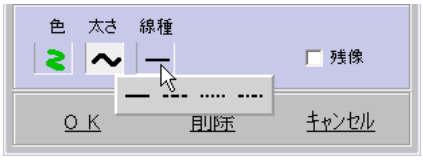
関数グラフのプロパティウィンドウでは，関数式，グラフ色，太さ，線種，残像設定を行うことができます。

- 式の訂正
  1. 関数式表示窓をクリック
  2. 関数電卓で式を訂正



- グラフ色の変更
  1. グラフ色表示窓をポイント
  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ポイント
  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ポイント
  2. パレットから線種を選ぶ

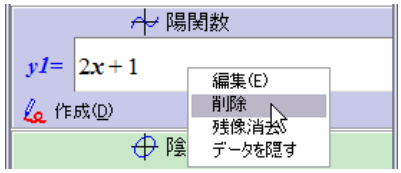
- 残像設定の変更
  ［残像］をチェック

2-3 関数グラフの訂正と削除

- 関数グラフの訂正
  1. 陽関数エリアの関数式表示窓をクリック
  2. 関数のプロパティウィンドウで訂正

     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
  1. 陽関数エリアの関数式表示窓を右クリック
  2. ポップアップメニューから［削除］を選ぶ
  3. 削除の確認を求められるので［OK］をクリック



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。

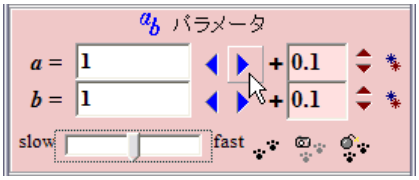
  $a\,,b\,,c\,,d\,,k\,,m\,,n\,,p\,,q\,,s\,,t\,,u\,,v\,,\theta$

  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 ◀ ▶ をクリック

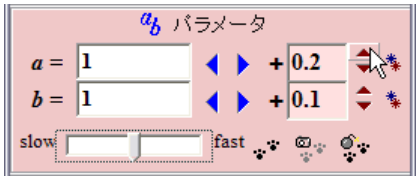
  ☆　押しつづけると連続的に変化します。



- **増減幅の変更**

  増減幅変更ボタン ▲ ▼ をクリック

  ☆　増減幅表示窓をクリックすることによってキーボードから数値を入力することができます。



- **増減速度の変更**

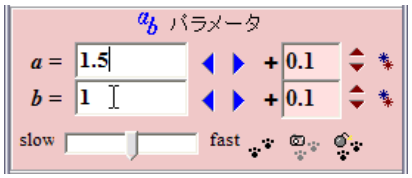
  増減速度のスライドを動かす

  増減ボタンを押しつづけているときの，速度を変更できます。

- **パラメータへの値の代入**

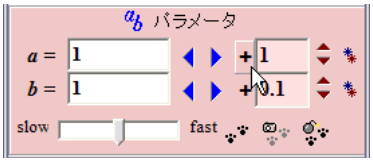
  パラメータ値表示窓をクリックしてキーボードから入力

  ☆　ダブルクリックで関数電卓を用いて入力できます。

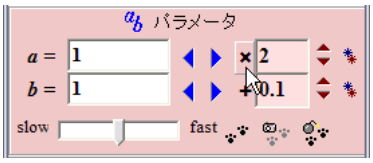


- **増減方法の変更**

  ［＋］をクリック



パラメータは，1 →2 →3 →4　と変化



パラメータは，1→ 2 →4 →8 と変化

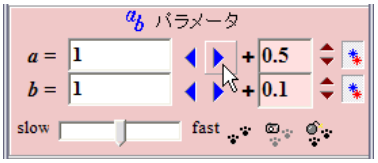
- **パラメータの同期**

  ひとつのパラメータを動かすだけで，複数のパラメータを同時に変化させることができます。

  1. **同期させるすべてのパラメータについて，同期ボタン をクリック**
  2. **パラメータ値を増減する**

     同期している他のパラメータの値も増減します。

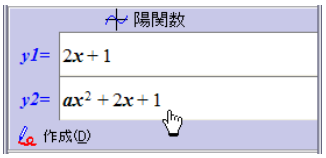
     右の例では，a の値を増減すると b の値も増減します。



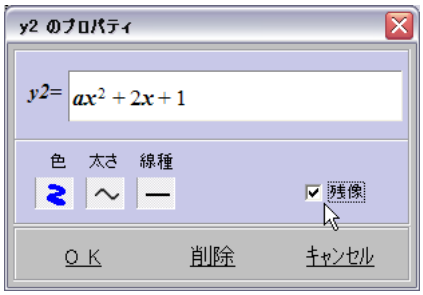
2-5 グラフを残す　―残像―

パラメータの増減によってグラフを動かすことができます。このとき，「残像」を使えば動いたグラフを残すことができます。

- **残像を残す**



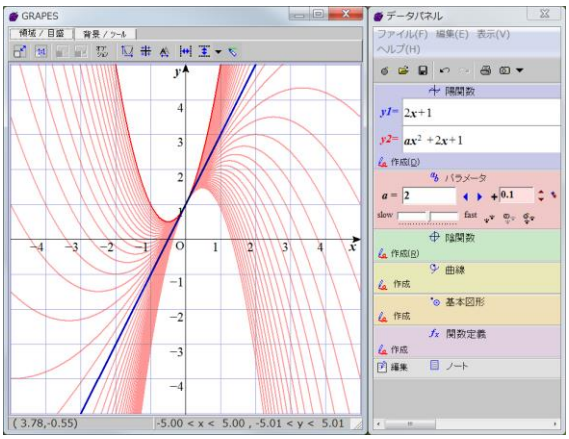
1. 残像を残したいグラフの関数式をクリック



2. ［残像］をチェック

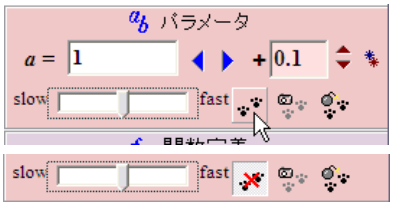
3. ［OK］をクリック

4. パラメータを動かす



- **残像の記録を一時停止する**

パラメータエリアの残像 off ボタン をクリック



ボタンが押し下げられた状態になり，残像が残らなくなります。再度のクリックで解除されます。

- **残像化**

パラメータエリアの残像化ボタン をクリック

現在描かれているすべてのグラフの残像を残すことができます。

- **残像を消す**

パラメータエリアの残像消去ボタン をクリック

第３章　表示領域と目盛りの調整

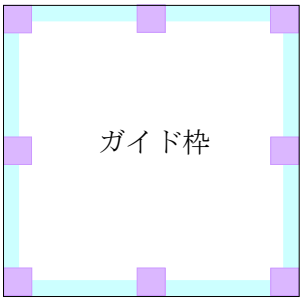
3-1 グラフの移動

- <u>領域モードにする</u>

  伸縮/移動ボタン **をクリック**

  伸縮/移動ボタンは押し下げられた状態になり，グラフ表示エリアにマウスを移動すると，薄い色のガイド枠が表示されます。
  この状態を領域モードといいます。

  もう一度クリックすると，領域モードは解除されます。



- <u>グラフの移動</u>

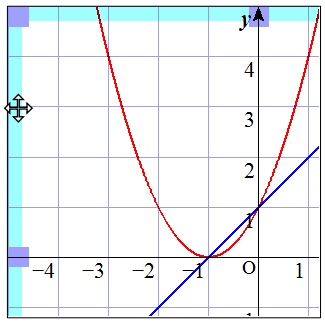
  a. **領域モードになっていることを確認する。**

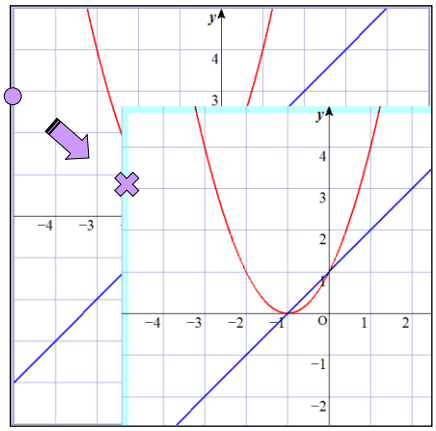
  b. **ガイド枠の水色部分をポイントする。**

  ガイド枠の色が濃くなり，
  カーソルの形状が ✥ に変化します。

  c. **左ボタンを押し**

  d. **移動したい方向にドラッグして，ボタンを離す。**



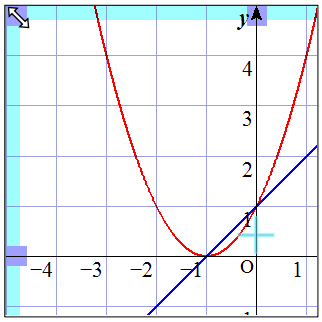


- 任意の位置から移動するには，Shift キーを押しながらドラッグします。
  Shift+ドラッグによる移動は，下記の縮小や拡大の途中でも有効です。
- 右ボタンのドラッグでも，任意の位置から移動できます。
- Shift+ドラッグや右ドラッグによる移動は，領域モード以外でも有効です。

3-2 グラフの縮小

- **グラフの縮小**

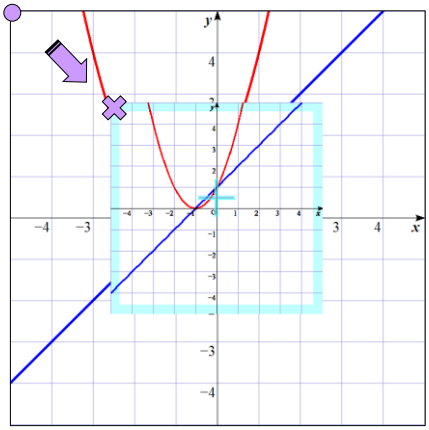
a. **領域モードになっていることを確認する。**
（「3-1 グラフの移動」参照）

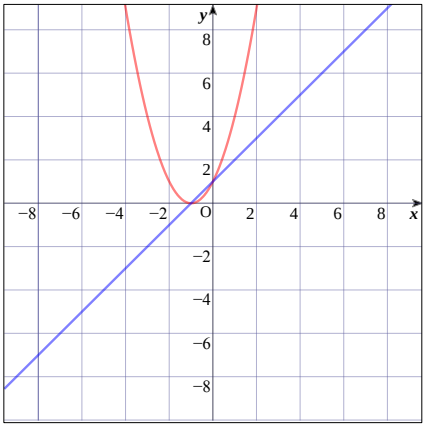


b. **ガイド枠の紫色部分をポイントする。**
ガイド枠の色が濃くなり，縮小の中心となるズームポイント＋が表示されます。また，カーソルの形状が ⇔⇕⇖⇗ に変化します。

c. **左ボタンを押し，**

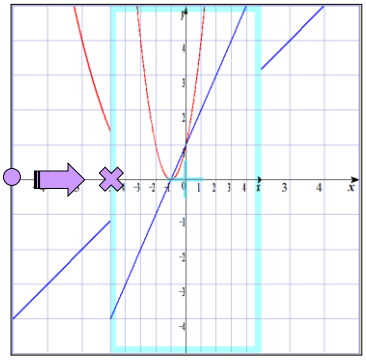
d. **画面内側に向かってドラッグして，ボタンを離す。**

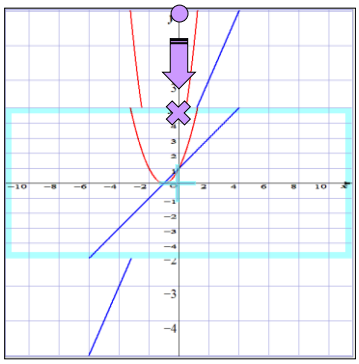




- **水平方向や垂直方向への縮小**

ガイド枠の辺の中央からドラッグすると，横方向か縦方向にだけ縮小されます。



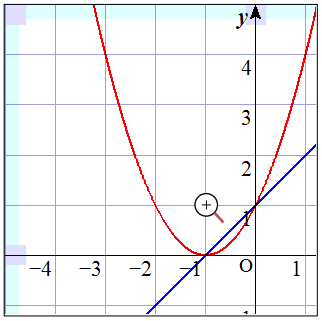


- ズームポイントの位置は画面上をクリックして変更することができます。また，拡大操作を行うと，拡大の中心にズームポイントが移動します。
- Ctrl キーを押しながらドラッグすると，縦横比を自由に変えることができます

3-3 グラフの拡大

- **グラフの拡大**

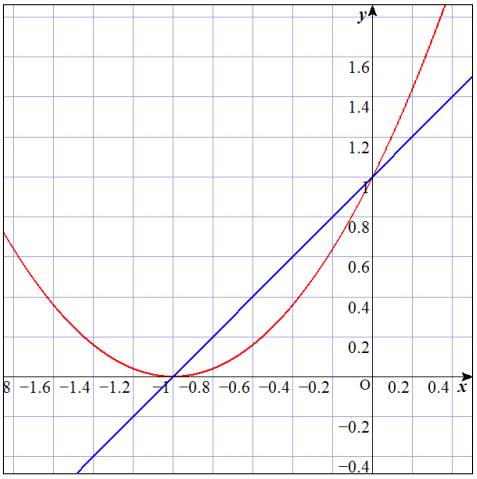
a. **領域モードになっていることを確認する。**
（「3-1 グラフの移動」参照）

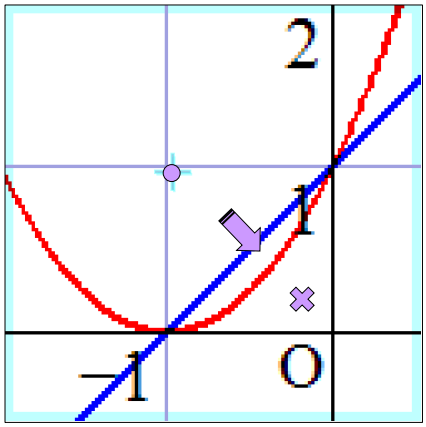


b. **拡大の中心をポイントする。**
ガイド枠内部をポイントすると、ガイド枠の色が薄くなり，カーソルの形状が ⊕ に変化します。

c. **左ボタンを押し，**

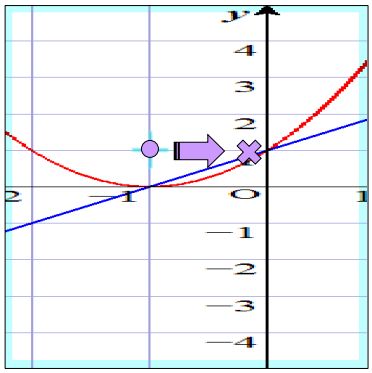
d. **斜め方向にドラッグして，ボタンを離す。**

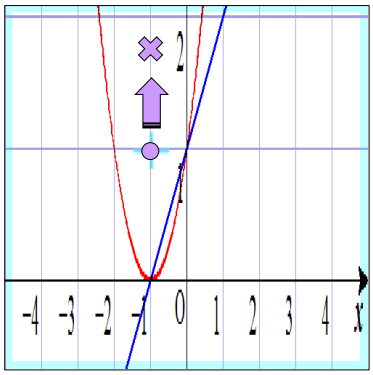




- **水平方向や垂直方向への拡大**

水平方向にドラッグすると横方向だけに拡大されます。
垂直方向にドラッグすると縦方向だけに拡大されます。





- Ctrl キーを押しながらドラッグすると，縦横比を自由に変えることができます。
- 最初にドラッグした方向と反対方向にドラッグすると，縮小になります。

3-4 領域モード以外からのグラフの伸縮/移動

領域変更（グラフの伸縮や移動）は「領域モード」で行うのが基本ですが，「領域モード」以外からでも領域変更を行う方法がいくつかあります。

- マウスのホイールを利用した伸縮

  マウスのホイールを回転させる。

  画面中央を中心にグラフが拡大もしくは縮小されます。

  （領域モードでは，ズームポイントを中心に伸縮されます。）

- 右ドラッグや Shift+ドラッグによる移動

  右ボタンを押してドラッグする。

  あるいは，Shift キーを押しながらドラッグする。

- ダブルクリックによる移動

  グラフ画面上でダブルクリックする。

  その位置が画面中央に移動します。

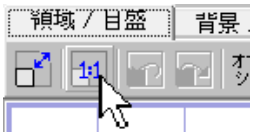
  上記の「マウスのホイールを利用した伸縮」と合わせて利用します。

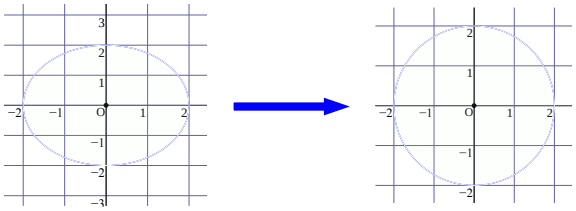
3-5 領域の変更　1 対 1 ，UNDO，REDO

- 1 対 1

1 対 1 ボタン 1:1 をクリック

グラフの縦横比が 1 対 1 になります。





- 領域変更の UNDO

領域 UNDO ボタン をクリック

表示範囲を変更を直前の状態に戻します。

UNDO 操作は 200 回まで遡ることができます。

- 領域変更の REDO

領域 REDO ボタン をクリック

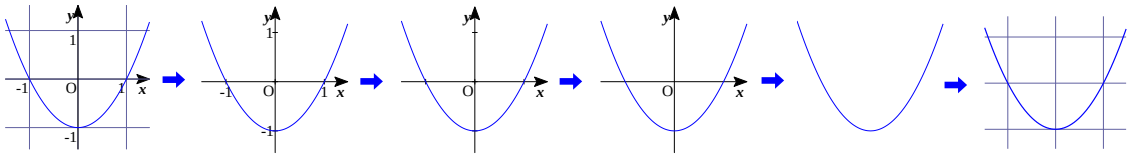
UNDO 操作を取り消すときに使います。

3-6 目盛線と軸の表示

- 目盛りや座標軸の表示切り替え

目盛/軸表示ボタン をクリック

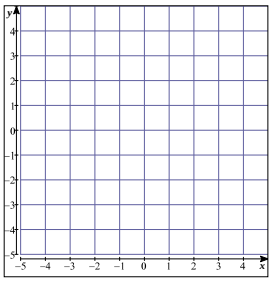
目盛りや軸の表示が切り替わります。



- 軸の領域外表示

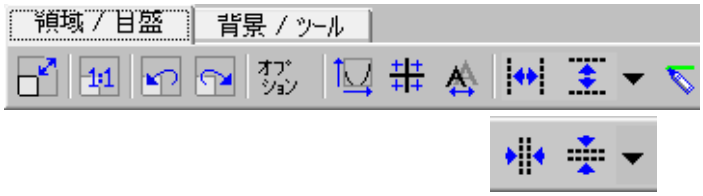
目盛パレットの領域外軸表示ボタン をクリック

もう一度クリックすると，表示は解除されます。



3-7 目盛幅を変える

- 目盛幅を変える



x 方向の目盛幅を広げるには，*x* 方向の目盛幅を広げるボタンをクリック

y 方向の目盛幅を広げるには，*y* 方向の目盛幅を広げるボタンをクリック

や の右クリックで目盛幅が縮まります。

x 方向の目盛幅を縮めるには，*x* 方向の目盛幅を縮めるボタンをクリック

y 方向の目盛幅を縮めるには，*y* 方向の目盛幅を縮めるボタンをクリック

や の右クリックで目盛幅が広がります。

と は，▼ で切り替わります。

目盛幅は，

$$1 \rightarrow \times 2 \rightarrow \times 5 \rightarrow \times 10$$

$$1 \rightarrow \times \frac{1}{2} \rightarrow \times \frac{1}{5} \rightarrow \times \frac{1}{10}$$

と変化します。

ただし，目盛幅基準値が $\pi, 2\pi, 90, 180$ などの時には，

$$1 \rightarrow \times 2 \rightarrow \times 3 \rightarrow \times 6 \rightarrow \times 9 \rightarrow \times 18 \rightarrow \times 45 \rightarrow \times 90$$

$$1 \rightarrow \times \frac{1}{2} \rightarrow \times \frac{1}{3} \rightarrow \times \frac{1}{6} \rightarrow \times \frac{1}{9} \rightarrow \times \frac{1}{18} \rightarrow \times \frac{1}{45} \rightarrow \times \frac{1}{90}$$

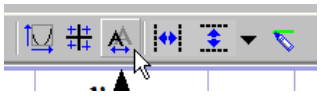
と変化します。

- 目盛り文字の間隔を変える

Ctrl キーを押しながら， をクリック

- 目盛りやラベル文字の大きさを変える

目盛パレットの目盛文字サイズ変更ボタン をクリック



3-8 目盛りと領域の標準設定

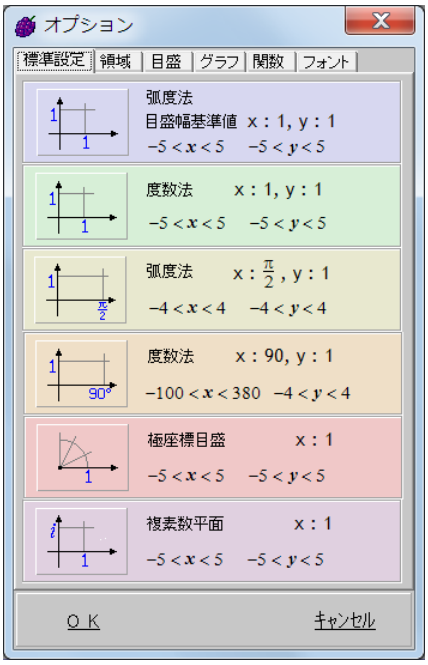
- **オプションウィンドウの表示**

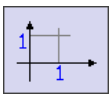
  領域パレットのオプションボタン[オプション]をクリック

  オプションウィンドウが表示されます。

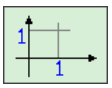
- **標準設定を使う**

  表示されている５つの中から選んでクリック

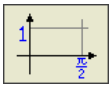




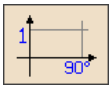
角の単位をラジアンにして，x 軸，y 軸ともに目盛幅基準値を 1 に設定します。これは GRAPES の初期状態です。



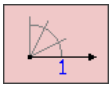
角の単位を度数法にして，x 軸，y 軸ともに目盛幅基準値を 1 に設定します。



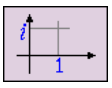
角の単位をラジアンにして，x 軸の目盛幅基準値を $\frac{\pi}{2}$ に設定します。三角関数のグラフを弧度法で描くときなどに選びます。



角の単位を度数法にして，x 軸の目盛幅基準値を 90 に設定します。また，x 軸の目盛に角の単位「° 」をつけます。三角関数のグラフを度数法で描くときに選びます。



極座標目盛りを表示します。



座標平面の複素数平面として扱います。
点を複素数として扱うことができます。
角度は弧度法に固定されます。

- **設定の終了**

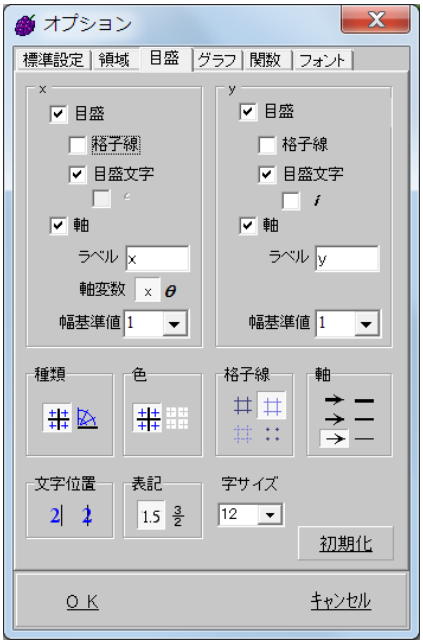
  ［OK］をクリック

3-9 目盛の詳細設定

- **目盛設定ウィンドウの表示**

1. 領域パレットのオプションボタン[オプション]をクリック
2. 次いで，［目盛］タグをクリック

- **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

- **虚数単位の表示**

［i］をチェックする。

複素数モード時に，y 軸目盛に虚数単位を付加します。

- **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

軸ラベルは，初期状態では“x”や“y”が表示されていますが，ここを書き換えることで，任意のラベルに変更することができます。

x 軸の軸変数を変えるには，[x]か[θ]をクリック

軸変数をθ にすると，θを独立変数として扱います。

$y=\sin\theta$ のようなグラフを表示するときに使います。

- **目盛幅基準値の設定**

幅基準値表示窓に基準値を入力

目盛幅は，通常この値を基準に

1 → ×2 → ×5 → ×10 と変化します。

- **目盛文字位置の設定**

表示したい文字位置を選ぶ

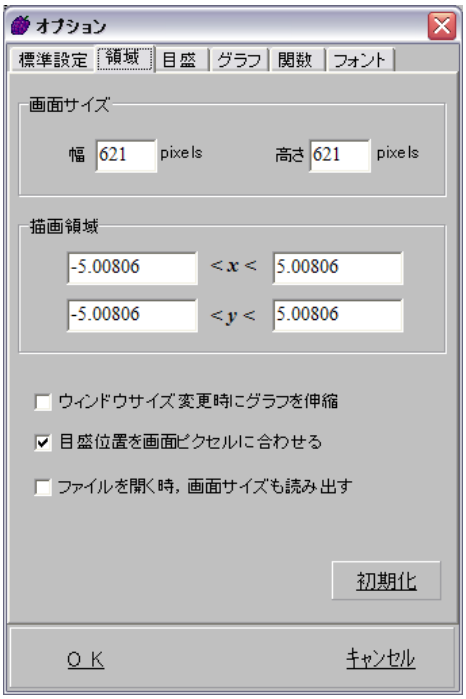
[2]を選ぶと，格子線をよけて文字が表示されます。

[1]を選ぶと，格子線の真上に文字が表示されます。

- **目盛文字の分数表記**

[$\frac{3}{2}$]を選ぶと，目盛文字が分数で表示されます。

3-10 領域の詳細設定



- **領域設定ウィンドウの表示**

  オプションウィンドウの［領域］タグをクリック

- **画面サイズの数値設定**

  幅と高さを入力

  画面サイズとは，グラフ表示領域の広さのことで，150×150 から 1600×1600 の間の整数で指定します。

- **描画領域の数値設定**

  x, y の変域を入力

- **ウィンドウサイズ変更時オプションの設定**

  初期設定では，ウィンドウサイズ変更時には，グラフの大きさは変わらず，表示領域が変化します。これをチェックすると，ウィンドウサイズを変更したとき，これに伴ってグラフの大きさも変更されます。

- **目盛位置を画面ピクセルに合わせる**

  目盛値が整数のとき，目盛格子線が画面ピクセル上に正確に位置するよう，領域を補正します。

- **ファイルを開くとき，画面サイズも読み出す**

  ファイルを開くときの画面サイズは，ファイル作成時の画面サイズではなく，縦横比だけが再現されます。これをチェックすると，ファイル作成時の画面サイズが再現されます。

- **起動時の状態に戻す**

  ［初期化］をクリック

  画面サイズ，描画領域を起動時の状態に戻します。

- **設定の終了**

  ［OK］をクリック

第４章　陰関数のグラフ

4-1 陰関数のグラフの作成・追加

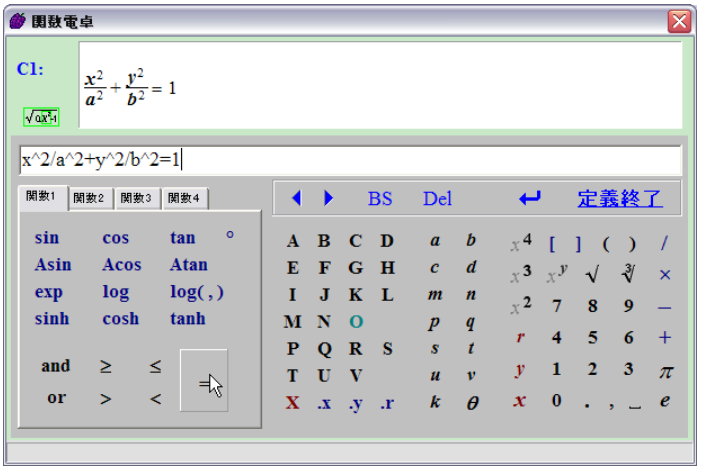
- 陰関数のグラフの作成・追加

1. 陰関数エリアの［作成］をクリック

2. 関数電卓で式入力

　キーボードもしくは関数電卓のボタンで「等式」を入力します。

　入力を終えたら，［定義終了］をクリックします。

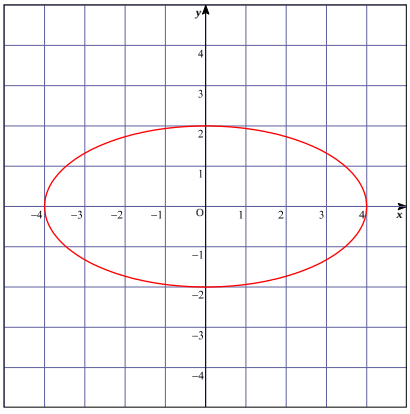


3. グラフスタイルの設定

　陰関数のプロパティウィンドウが表示されます。

　グラフ色や太さの変更方法は，「2-2 関数グラフのプロパティ」を参照してください。

設定を終えたら［OK］をクリックします。



陰関数は，９個まで描くことができます。

4-2 不等式の領域

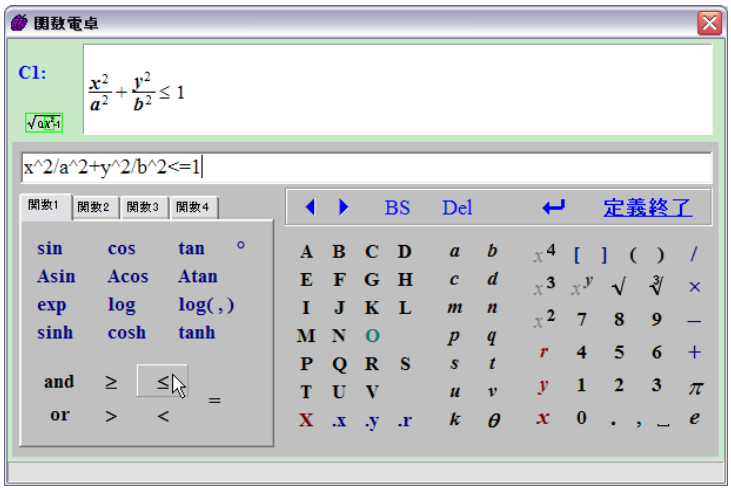
- **不等式の領域**

1. 陰関数エリアの［作成］をクリック

2. 関数電卓で式入力

　キーボードもしくは関数電卓のボタンで不等式を入力します。

　入力を終えたら，［定義終了］をクリックします。

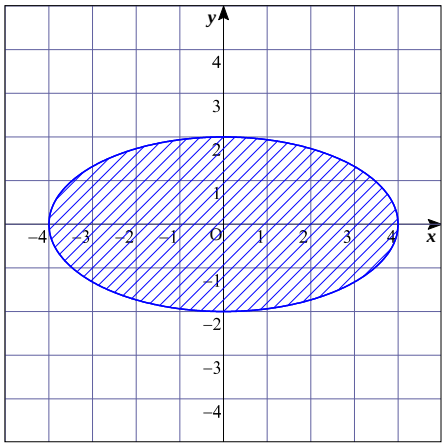


3. グラフスタイルの設定

　陰関数のプロパティウィンドウが表示されます。

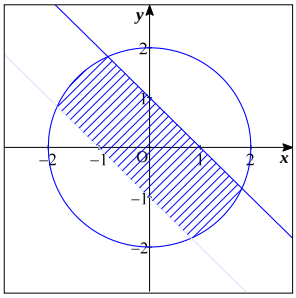
　グラフの色や太さのほかに領域のハッチングパターンを選択します。

　設定を終えたら，［OK］をクリックします。



4-3 複数の領域の共通部分と和集合

- **複数の領域の共通部分の表示**



それぞれの領域を表す不等式を“and”で結合する

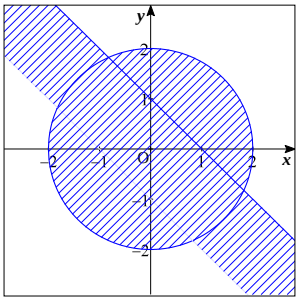
例：$(x^2+y^2 \le 2^2)$ and $(-1<x+y\le 1)$

上記は，C1： $x^2+y^2 \le 2^2$ ，C2： $-1<x+y\le 1$

C3 : C1 and C2

とすることもできます。

- **複数の領域の和集合の表示**



それぞれの領域を表す不等式を“or”で結合する

例：$(x^2+y^2 \le 2^2)$ or $(-1<x+y\le 1)$

上記は，C1： $x^2+y^2 \le 2^2$ ，C2： $-1<x+y\le 1$

C3 : C1 or C2

とすることもできます。

☆ 「and」と「or」が混在する場合，「and」が優先します。

☆ 和集合や共通集合に，括弧を用いたネストはできません。

したがって，次のような表現はできません。

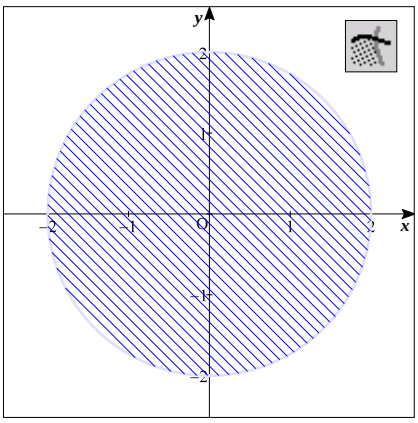
例 1：(C1 or C2) and C3

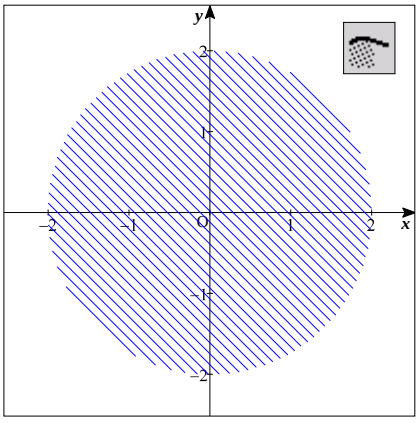
例 2：C1 or (C2 and C3)

ただし，例 2 の領域は「C1 or C2 and C3」として描くことができます。

- **境界線の扱い**

$x^2+y^2<2^2$ のように，不等式の表す領域に境界線が含まれない場合，境界線は淡色で表示されます。関数のプロパティを操作すれば淡色の境界線を表示しないようにすることができます。





4-4 陰関数はどうやって描いているの？

- **$y = f(x)$ 型の関数**

  これは，陽関数そのものですから，最も速く描くことができます。

- **２次曲線**

  ２次曲線は，２次方程式を解いて陽関数に直しています。

上記の２タイプ以外の関数については，画面上のすべての点について，その点での関数値を求め，正領域と負領域の境界線を調べて描いています。

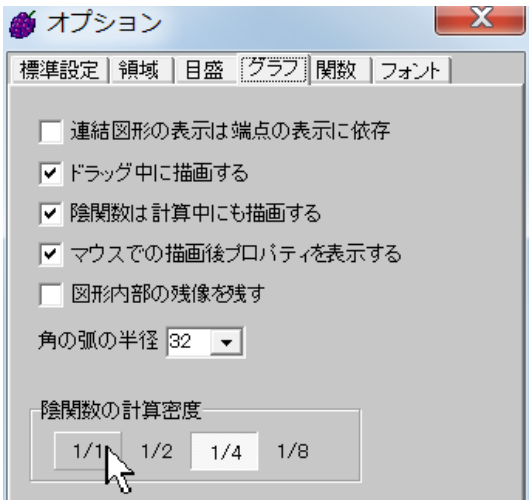
- **10 次以下の整関数**

  x , y のいずれもが 10 次以下の多項式であるとき，ラグランジュ補間を用いて計算します。表示にストレスを感じることはないでしょう。

☆　複雑な関数の場合，10 次以下の多項式であるかどうかの判断が難しくなり，正確なグラフが描かれなくなる可能性があります。この場合，陰関数のプロパティウィンドウで，［整関数の高速計算］のチェックを外します。

- **一般の関数**

  関数によってはかなりの時間を要します。そのために，画面上の点の密度を落として計算するようにしています。もし，正確なグラフが得られないようなら，グラフ設定の「陰関数の計算密度」を変えてみてください。



陰関数のグラフ描画は，連続な関数を扱うことを前提にしています。分数を含む関数では分母を払うなど，連続関数になるように工夫してください。

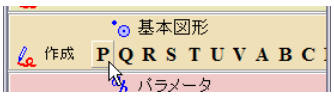
例： $x = \tan y \rightarrow x\cos y = \sin y$

第５章　点と軌跡

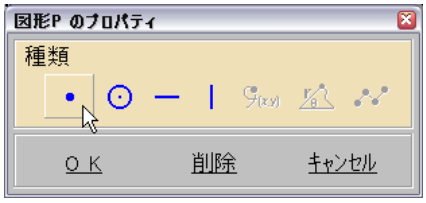
5-1 点を打つ

- 点を打つ

1. 基本図形エリアの［作成］をポイント
2. 図形名を選ぶ



3. 図形種類で点を選ぶ

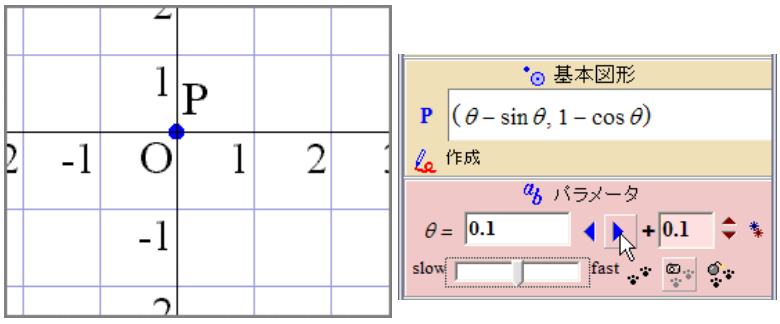


4. x に式を入力
5. y も式入力

6. グラフスタイルの設定

　点の色や大きさなどの変更方法は，次ページを参考にしてください。

7. 設定を終えたら［OK］をクリック



まだ，軌跡はできません。軌跡を描くには，残像や軌跡の太さを設定するとともに，パラメータを動かす必要があります。

- 手軽に点を打つには

　座標が式ではなく数値で与えられているとき，もっと簡単な操作で点を打つことができます。「5-7 点とドラッグ」を参照してください。

5-2 点のプロパティ

- x 座標の入力・訂正

1. x 座標の式表示窓をクリック
2. 関数電卓で訂正

- y 座標の入力・訂正

1. y 座標の式表示窓をクリック
2. 関数電卓で訂正

- 点の色や大きさの変更

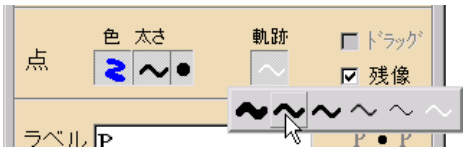
1. 点の色や大きさ表示窓をポイント
2. パレットから選ぶ

大きさ表示窓の右側をポイントすれば，白抜き点にすることができます。

グラフ色や太さの変更方法は,「2-2 関数グラフのプロパティ」を参照してください。

- 軌跡の太さの変更

1. 軌跡の太さ表示窓をポイント
2. パレットから太さを選ぶ



- 残像設定

［残像］をチェック

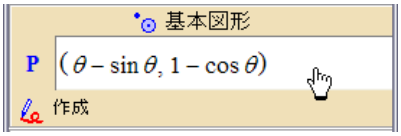
- ラベルの表示設定

ラベルを表示する位置をクリック

5-3 軌跡と残像

- 軌跡と残像

1. 基本図形エリアの式表示部分をクリック

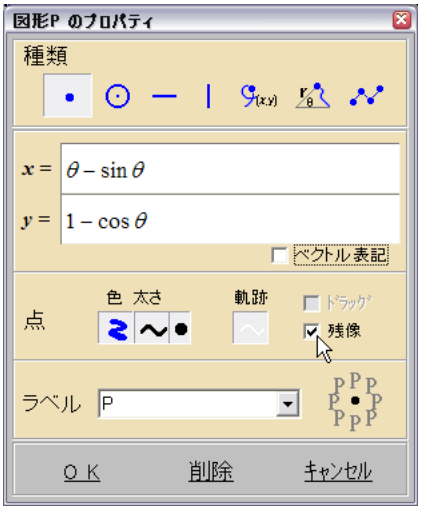


2. 軌跡の太さを選ぶ

3. ［残像］をチェック

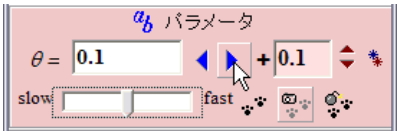
   点の残像がなくても軌跡を残すことはできます。

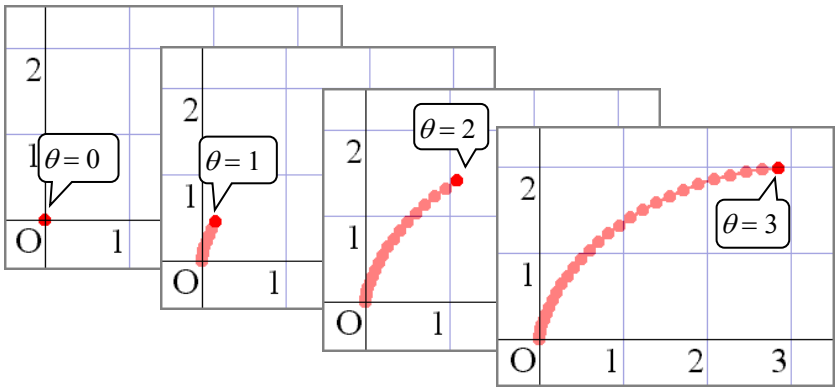
4. ［OK］をクリックして決定



5. パラメータを動かす。

   パラメータを動かすと，点の残像が残り，軌跡が描かれます。





- 残像や軌跡を消す

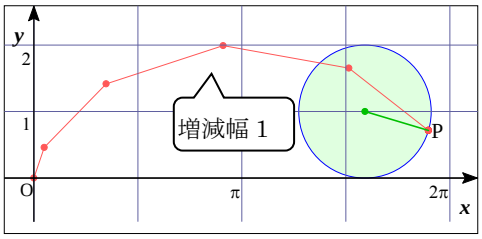
  パラメータエリアの残像消去ボタン をクリック

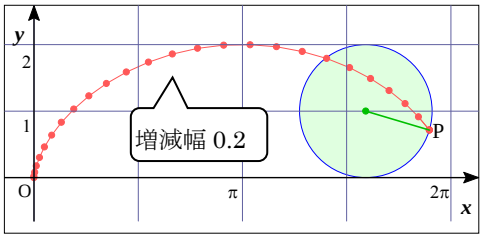
5-4 軌跡を使いこなす

軌跡は描いた点を折れ線で結んだものです。
すでに描いた軌跡に対してできる操作は，隠す，消す，濃さの変更の 3 つだけです。

- **軌跡をきれいに描く**

軌跡は描いた点を結んでいるだけなので，きれいに描くためには増減幅を小さくします。



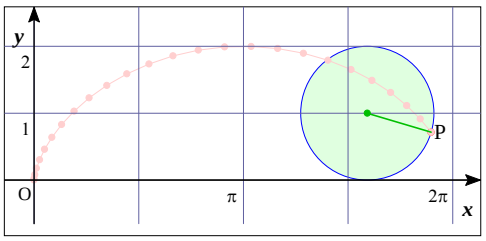


- **軌跡の濃さを変える**

1. ［領域／目盛］パレットの［オプション］をクリック
2. グラフ設定で残像の濃さを選ぶ。

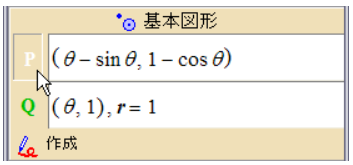
この変更は，すべての残像や軌跡に適用されます。

すでに描いた軌跡や残像について，色や太さを変える事はできません。

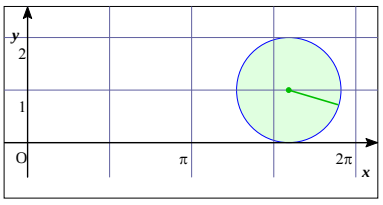


- **特定の残像や軌跡だけを隠す**

基本図形エリアの図形名をクリック

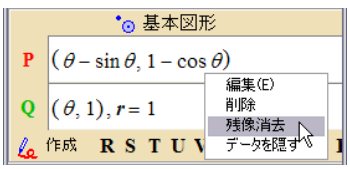


基本図形エリアの図形名はボタンになっていて，これを「表示スイッチ」といいます。表示スイッチを使えば，ここの図形について表示・非表示を瞬時に切り替えることができます。



- **特定の残像や軌跡だけを消す**

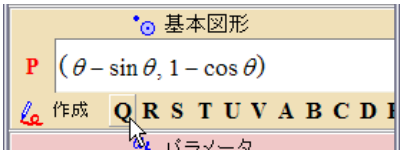
1. データパネルの関数式表示部分を右クリック
2. 残像消去を選ぶ



5-5 円を描く

- 円を描く

1. 基本図形エリアの［作成］をポイント



2. 図形名を選ぶ

3. 図形種類で円を選ぶ

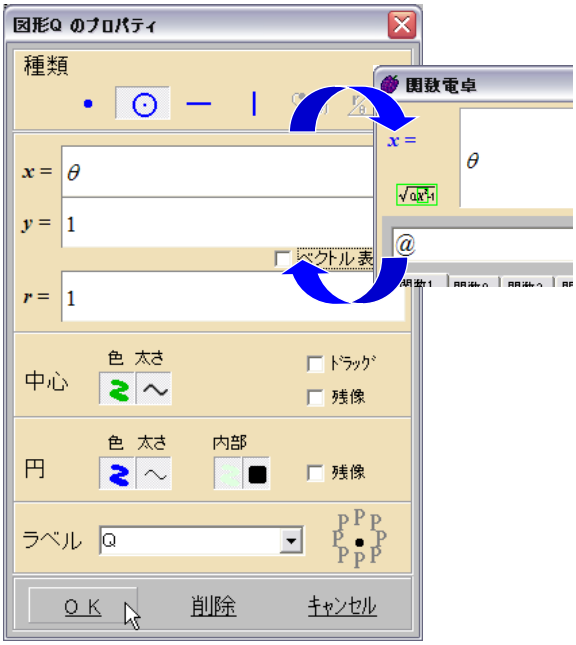
4. 中心 x, y の式を入力

5. 半径 r も式入力

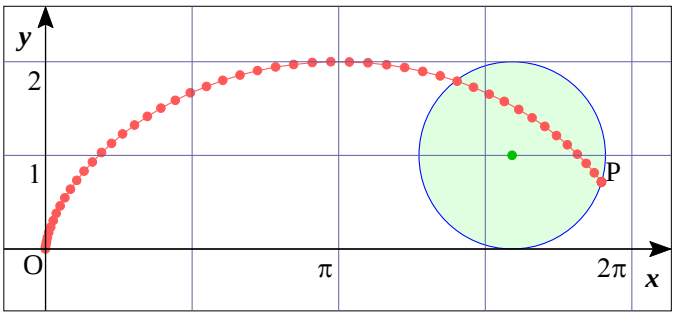
6. 中心の色，大きさなどを設定

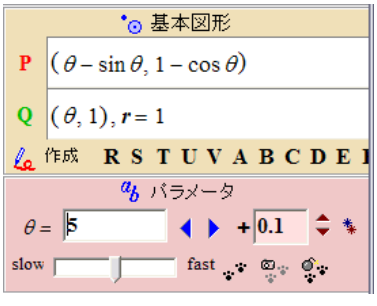
色や太さの変更方法は，「2-2 関数グラフのプロパティ」を参照してください。

7. 円周の色，太さ，内部色などを設定



8. ［OK］をクリック

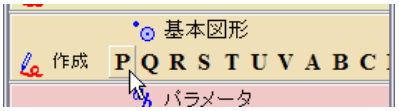




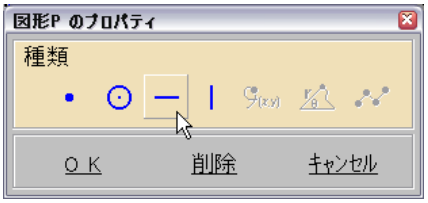
5-6 水平線，垂直線を描く

- 水平線描く

1. 基本図形エリアの［作成］をポイント



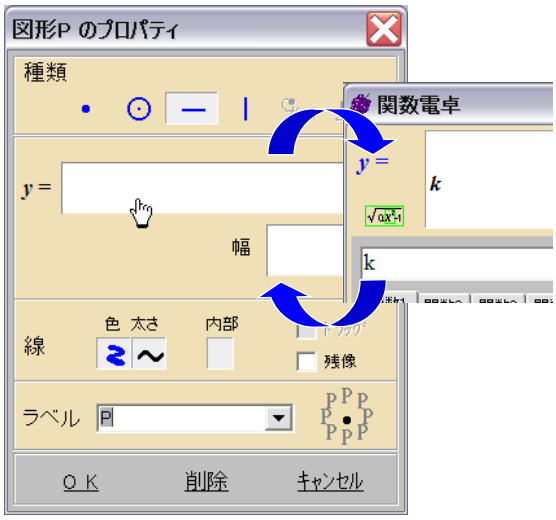
2. 図形名を選ぶ



3. 図形種類で水平線を選ぶ

4. 水平線の y 座標を入力

5. 幅があるときは，幅も入力

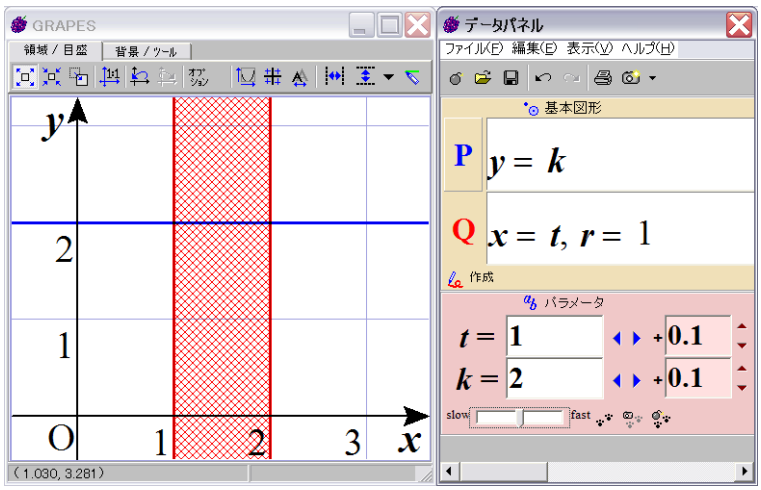


6. 線の色，大きさなどを設定

色や太さの変更方法は，「2-2 関数グラフのプロパティ」を参照してください。

7. 幅があるときは，内部色などを設定

8. ［OK］をクリック



- 垂直線を描く

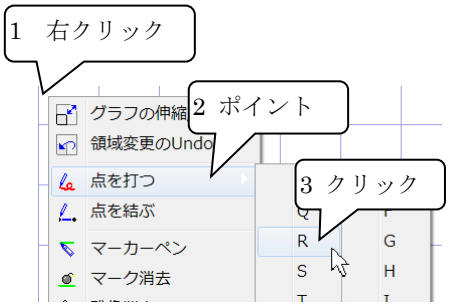
水平線と同様です。

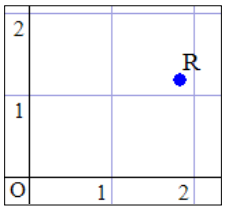
5-7 点とドラッグ

- **点を打つ**

1. 点を打ちたい位置で右クリック（Shift＋ドラッグ）
2. ［点を打つ］をポイント
3. 点の名前を選んでクリック

   点のプロパティウィンドウが表示されます。
4. 点の色や大きさなどを決める
5. ［OK］をクリック





- **点のドラッグ**

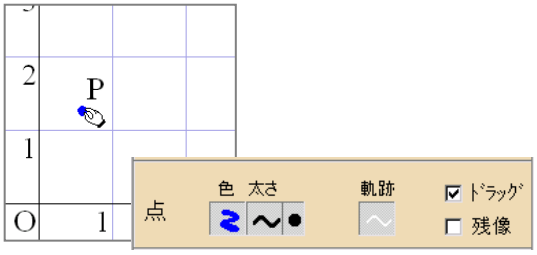
基本図形は，座標が数値のとき，ドラッグして動かすことができます。

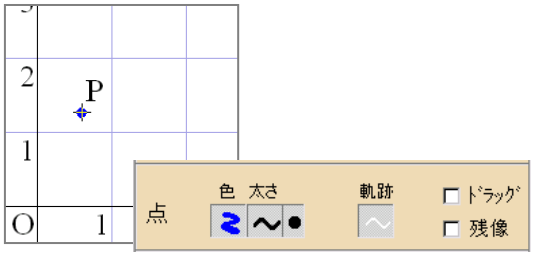
1. 点にポイントする

   ドラッグ可能な場合マウスカーソルの形状が変化します。（下左図）
2. 左ボタンでドラッグする。

a. ［ドラッグ］チェックボックスが，
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。

b. ［ドラッグ］チェックボックスが，
チェックされていないとき，
ドラッグすることはできません。





複数のドラッグできる点が同じ位置にあるとき，これらの点の中で，基本図形エリアで最も下に表示されている点だけがドラッグできます。

- **点のドラッグのオプション**
  - **格子線上をドラッグ**

    格子線の近くをドラッグすると，格子線に吸い寄せられます。
  - **格子点に固定**

    Ctrl キーを押しながらドラッグすると，格子点に固定されます。
  - **曲線上をドラッグ**

    Alt キーを押しながらドラッグすると，近くの曲線（陽関数のグラフ，陰関数のグラフ，曲線）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。

- **曲線上の点のドラッグ**

  曲線上に点をとり，その点をドラッグすることができます。「6-1 媒介変数表示の曲線 10」を参照してください。

第６章　曲線

6-1 媒介変数表示の曲線

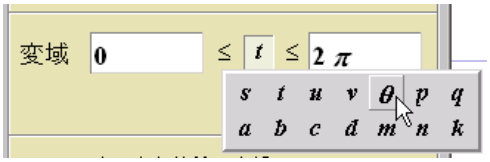
- 媒介変数表示の曲線を描く

1. 曲線エリアの［作成］をポイント
2. 図形名を選ぶ

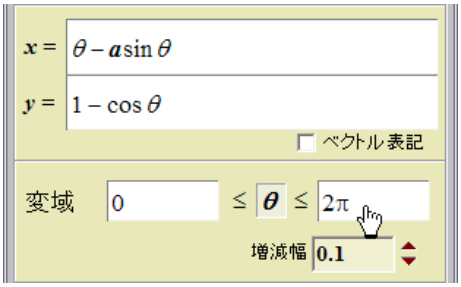
3. 図形の種類で曲線ボタン をクリック

4. x に関数式を入力
5. y も関数式を入力

6. 媒介変数の文字の設定
   a. 変域エリアの媒介変数文字表示窓をポイント
   b. 文字パレットから媒介変数文字を選ぶ

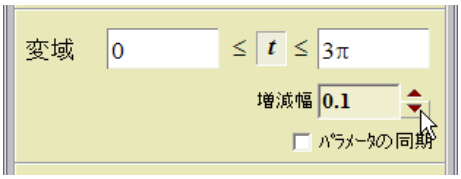


7. 媒介変数の変域の設定



変域表示窓をクリックして関数電卓で入力します。変域の下限と上限をそれぞれ設定することができます。

8. 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。ただし，GRAPES の仕様で，

（変域の上端－下端）／増減幅 ≦ 5000

である必要があります。

★［パラメータの同期］をチェックすると，データパネルのパラメータ値が，ここで定めた変域で制約を受けます。また，データパネルのパラメータの増減幅とここでの増減幅が同期します。

9. 曲線のプロパティの設定

色，太さ，線種，内部色，残像を設定します。

色や太さの変更方法は,「2-2 関数グラフのプロパティ」を参照してください。

10. 点のプロパティの設定

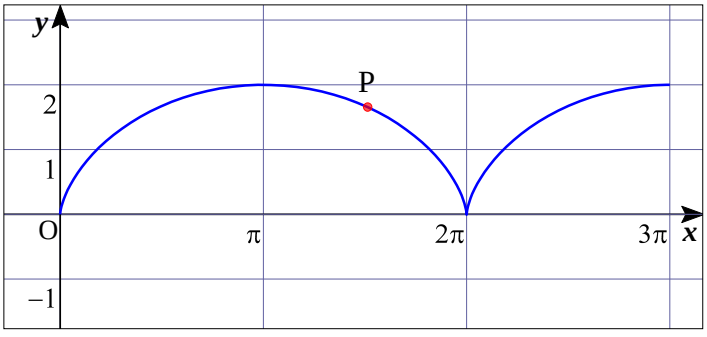
曲線上を動く点を描くことができます。

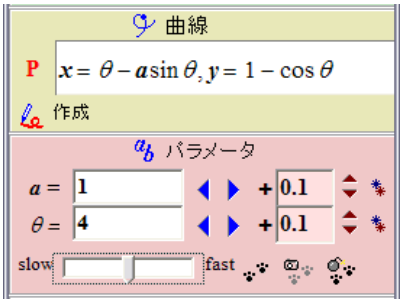
★［ドラッグ］をチェックすると，曲線上の点をドラッグすることができます。

11. ラベルの設定

ラベル文字と表示位置を設定します。

12. ［OK］をクリック





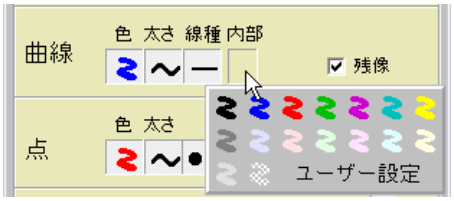
6-2 曲線内部のペイント

- 内部のペイント

  媒介変数表示の曲線や極方程式のグラフでは，曲線の内部をペイントすることができます。ただし，デフォルトでは内部は透明色になっているため，ペイントするにはこれを変更する必要があります。

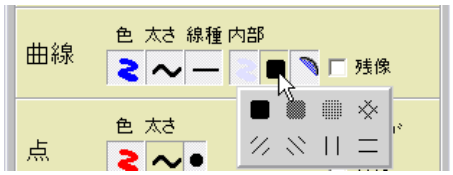
- 内部色の変更

  1. 内部色表示窓をポイント
  2. パレットから色を選ぶ



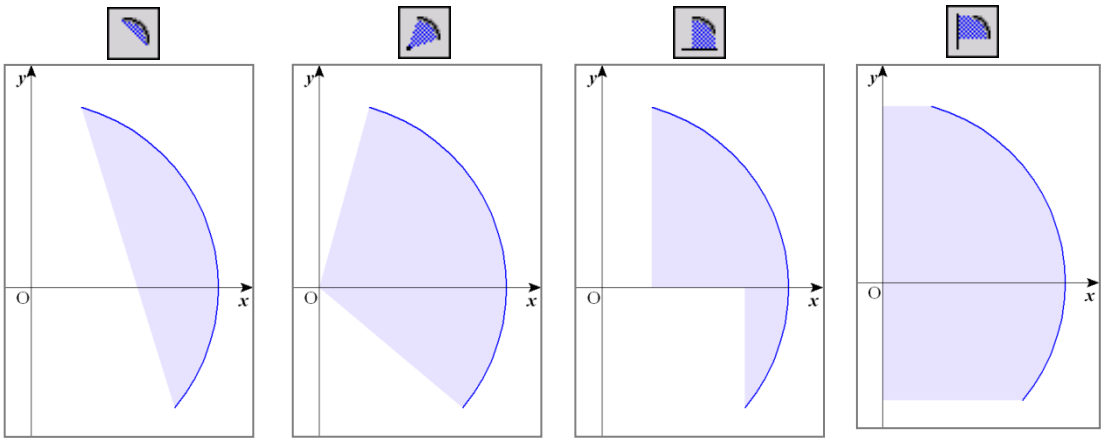
- ペイントパターンの変更

  1. パターン表示窓をポイント
  2. パレットからパターンを選ぶ



- ペイント領域の変更

  1. ペイント領域表示窓をポイント
  2. パレットからペイント領域を選ぶ



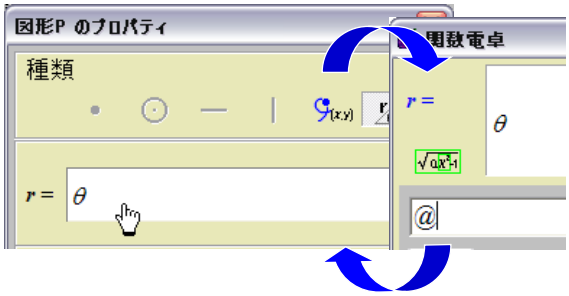
6-3 極方程式のグラフ

- 極方程式のグラフを描く

1. 曲線エリアの［作成］をポイント
2. 図形名を選ぶ

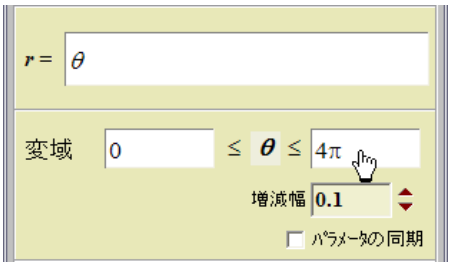
3. 図形の種類で［極方程式］ボタン をクリック

4. r の式をθを用いて入力



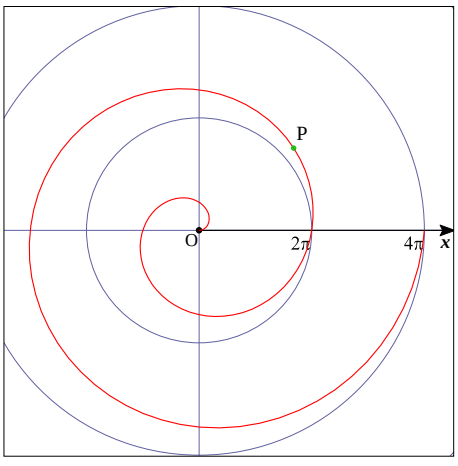
5. θの変域や増減幅の設定

変域表示窓をクリックして関数電卓で入力します。変域の下限と上限をそれぞれ設定することができます。



6. 曲線や点，ラベルのプロパティの設定

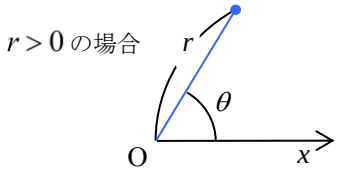
7. ［OK］をクリック

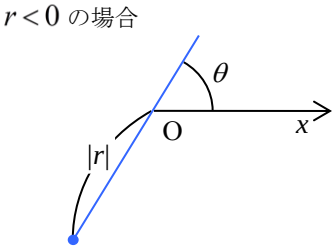


6-4 負の動径

- **負の動径とは**

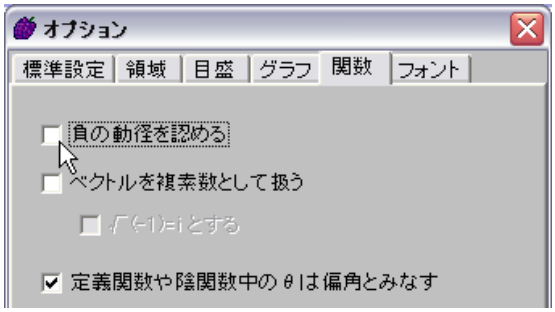
極座標では，点の位置を極（原点）からの距離－動径 r －と，始線（x 軸）からの角度－偏角θ－で表します。したがって，動径は 0 以上の値しかとりません。しかしながら，極方程式よっては負の動径を認めたほうが都合のよい場合があります，この場合，下右図のように，本来の方向とは逆の方向に点をとります。初期状態では負の動径を認める設定になっています。負の動径を認める場合，$(-r,\theta)=(r,\theta+\pi)$ です。



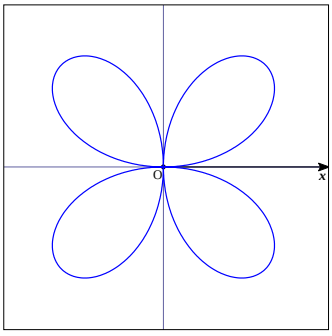


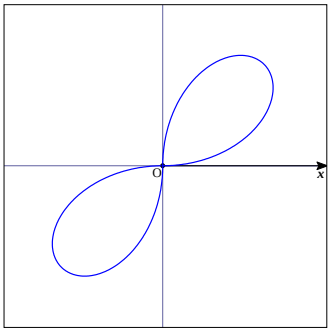
- **負の動径を認めないようにするには**

オプションウィンドウの［関数］設定で，「負の動径を認める」のチェックを外す



これ以後に描くグラフでは，動径が負の部分は描画されなくなります。



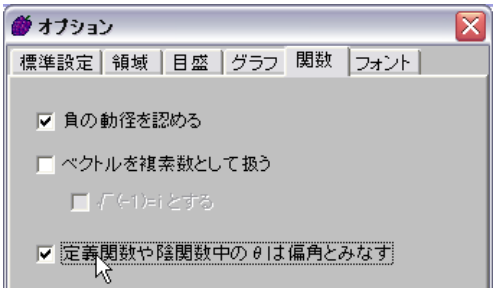


$r=\sin 2\theta$ において，負の動径を認めた場合（左図）と認めない場合（右図）

6-5 陰関数と極方程式

陰関数では，動径 r および偏角 θ 使うことができますから，これを用いて極方程式のグラフを描くことができます。

- **陰関数と極方程式**



1. **オプションウィンドウの［関数］設定で，［定義関数や陰関数中の θ は偏角とみなす］をチェック**

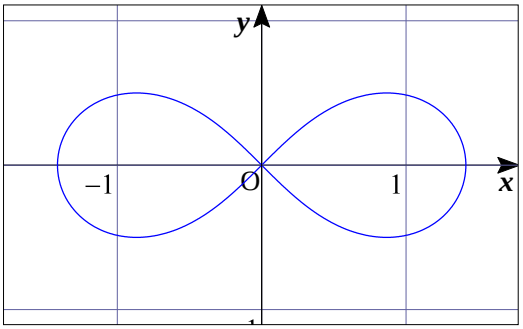
   以後，θ はパラメータではなく，偏角とみなされます。

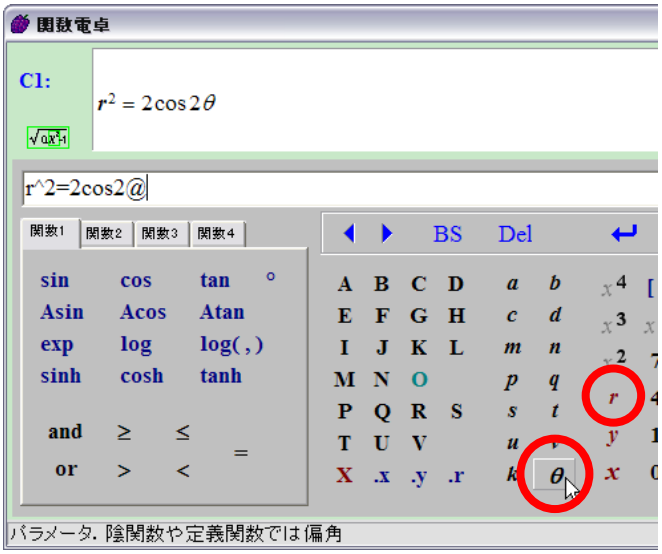
2. **陰関数エリアの［作成］をクリック**

3. **電卓で式入力**

   動径 r，偏角 θ を用いて式を入力します。

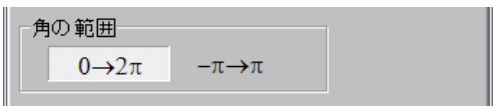
4. **以後の操作は，陰関数のグラフとまったく同じ**





- **陰関数での動径と偏角の扱い**

偏角 θ の範囲は，$0 \le \theta < 2\pi$ か $-\pi \le \theta < \pi$ です。いずれにするかは，オプションウィンドウの［関数］設定の「角の範囲」で設定します。



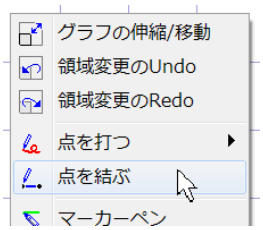
なお，陰関数の中では負の動径は認められていません。

第７章　連結図形

7-1　２点を結ぶ

- **２点を結ぶ**

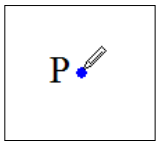
1. **グラフ表示エリアで右クリックし，ポップアップメニューから［点を結ぶ］をクリック**



　これで，「点を結ぶ」モードになります。
「点を結ぶモード」は，もう一度クリックするか，
拡大・縮小・移動ボタンのいずれかのクリックで
解除されます。

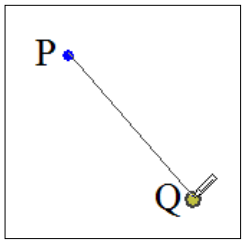
2. **一方の点にマウスをポイント**

　点にマウスを重ねると，ペン型カーソルになります。



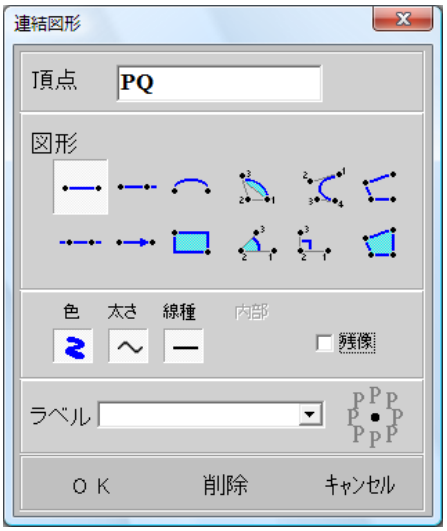
3. **線分の両端をマウスで指定**

a. 線分の一端で左ボタンを押し，
b. そのままドラッグして，
c. もう一端で離す。

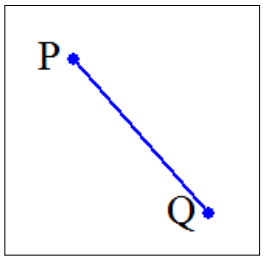


4. **線分の諸設定**

　連結図形のプロパティウィンドウで，図形の種類，
色，太さ，領域色，残像の有無を設定します。



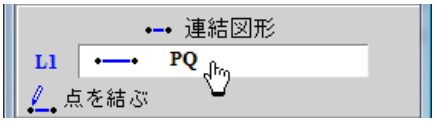
5. **［OK］をクリックして決定**



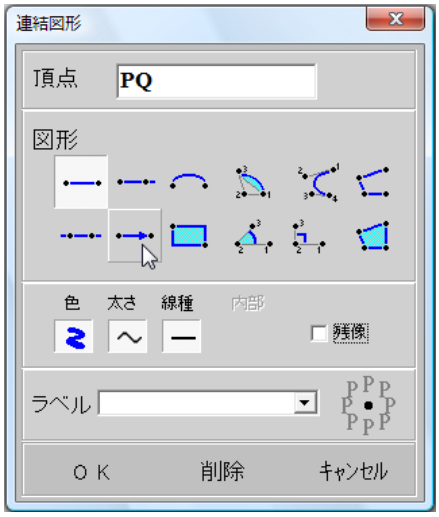
7-2 連結図形のプロパティと訂正・削除

- 連結図形の訂正・削除

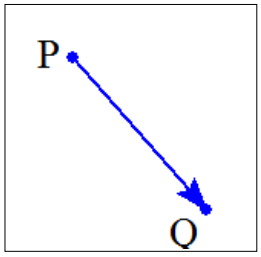
1. 連結図形エリアで，該当する図形をクリック
   または，連結図形上で右クリックし，ポップアップメニューから［連結図形のプロパティ］をクリック



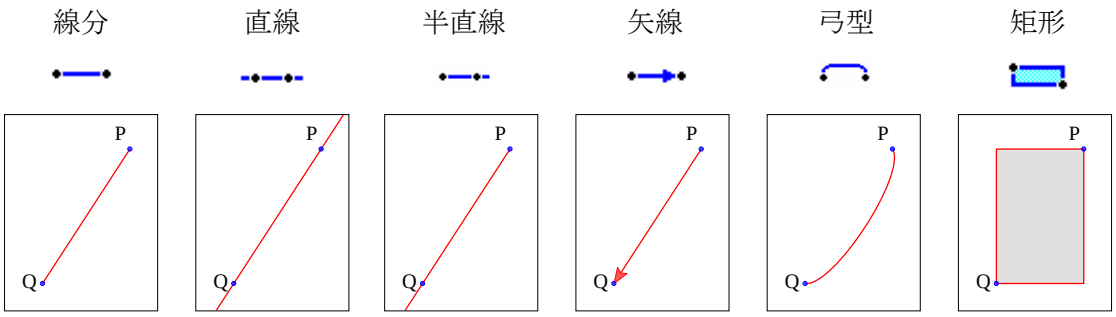
2. 図形の種類，色，太さ，領域色，残像を設定
   削除のときは，［削除］をクリック



3. ［OK］をクリック



- 図形の種類について



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

7-3 折れ線や多角形を描く

● 折れ線や多角形の表示

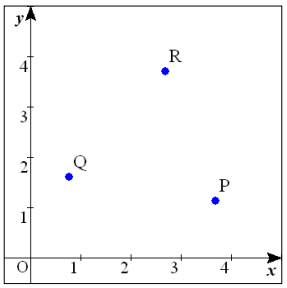
操作に先立って，3 個以上の点を描いておく必要があります。ここでは，3 点 P，Q，R がすでに描かれているとして，これをもとに三角形を描く方法の説明をします。

1. **グラフ表示エリアで右クリックし，ポップアップメニューから［点を結ぶ］をクリック**

   「線を結ぶ」モードにします。

2. **2 点 P, Q を結ぶ**

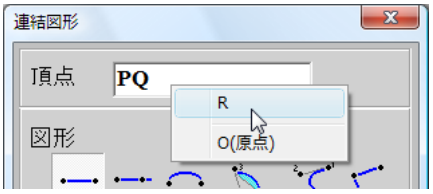
   点 P で左ボタンを押し下げ，そのまま点 Q までドラッグしてボタンを離します。



［連結図形］プロパティウィンドウが表示されます。

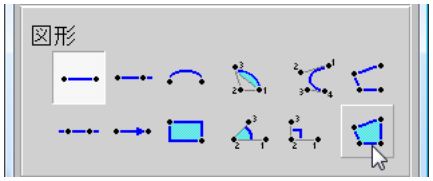
3. **頂点表示窓を右クリック**

   ポップアップウィンドウが開き，指定可能な頂点の一覧が表示されます。

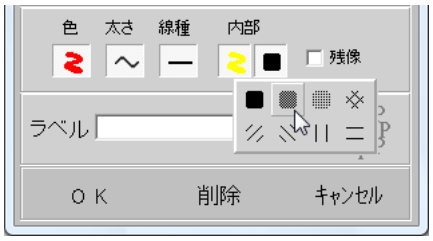


4. **ポップアップメニューの［R］をクリック**
5. **［多角形］ボタンをクリック**

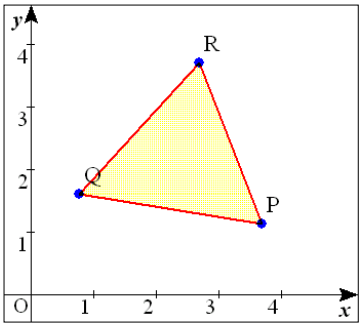
   折れ線を描くときは［折れ線］ボタンをクリックします。



6. **境界色や内部色とハッチングパターンを設定**



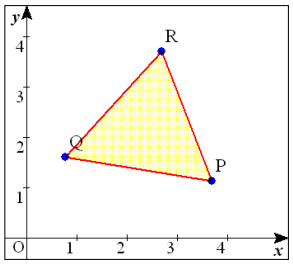
7. **［OK］をクリック**



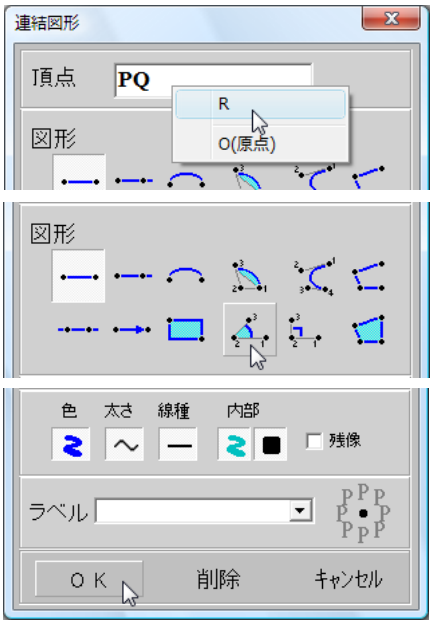
7-4 角を表示する

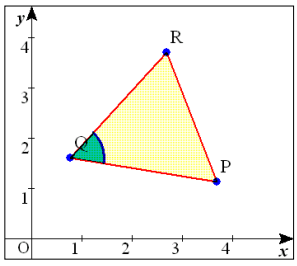
● 角の表示

操作に先立って，3 個以上の点を描いておく必要があります。ここでは，三角形 PQR がすでに描かれているとして，これをもとに∠PQR を描く方法の説明をします。

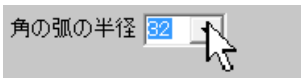


1. 「線を結ぶ」モードにする。
   「線を結ぶ」モードについては，前頁を参照してください。
2. 点 P, Q を結ぶ
3. 頂点表示窓を右クリック
4. ポップアップメニューの［R］をクリック
5. ［角］ボタン をクリック
6. 線色および内部色を設定
7. ［OK］をクリック





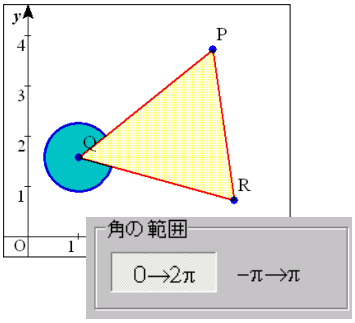
角を表す弧の半径を変えるには，オプションウィンドウの［グラフ］設定で，「角の弧の半径」を変更します。

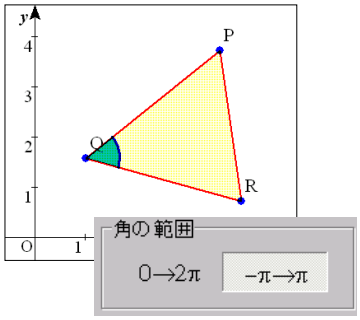


● 角の向き

角は左回りに表示されます。例えば，∠PQR は，半直線 QP から出発して左回りに半直線 QR に到達するまでの角です。

常に 180° 以下の角を表示したい場合には，オプションウィンドウでの［関数］設定で，「角の範囲」を［$-\pi \to \pi$］に設定します。

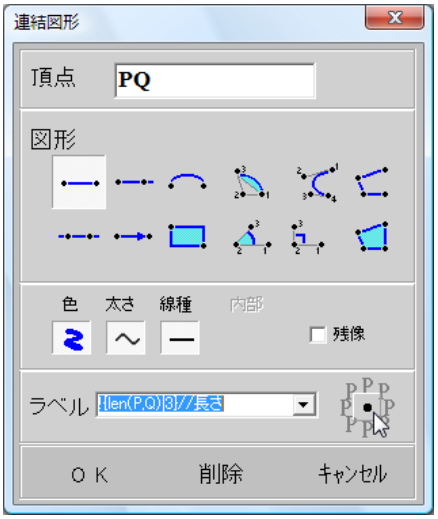


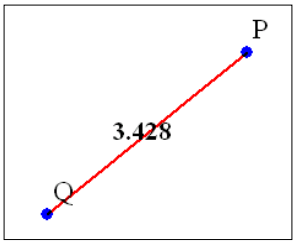


7-5 線分の長さや角の大きさを表示する

- 線分の長さを表示する

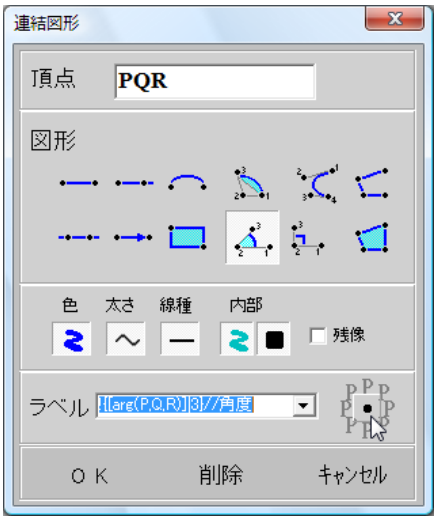
1. 線分 PQ を右クリック
2. ［連結図形のプロパティ］をクリック
3. ラベル表示窓のプルダウンメニューから
   “!{[PQ]}”を選ぶ
4. ラベル表示位置をクリック

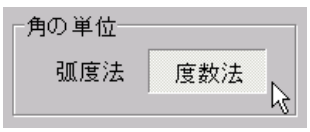


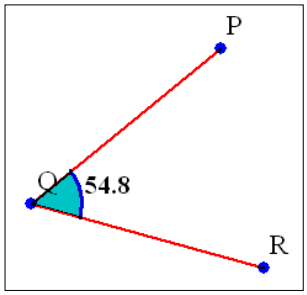


- 角の大きさを表示する

1. 角を示す弧の部分を右クリック
2. ［連結図形のプロパティ］をクリック
3. ラベル表示窓のプルダウンメニューから
   “!{[arg(P,Q,R)]|3}”を選ぶ
4. ラベル表示位置をクリック
5. オプションウィンドウの［関数］設定で「角の単位」を［度数法］にする







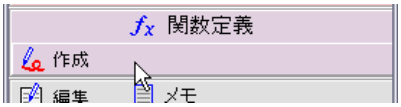
第８章　関数の相互参照

8-1 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. 関数定義エリアの［作成］をクリック

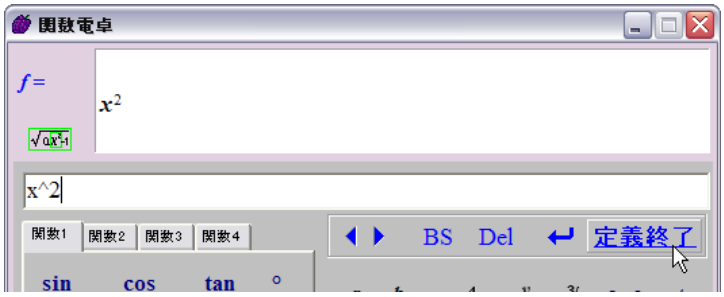


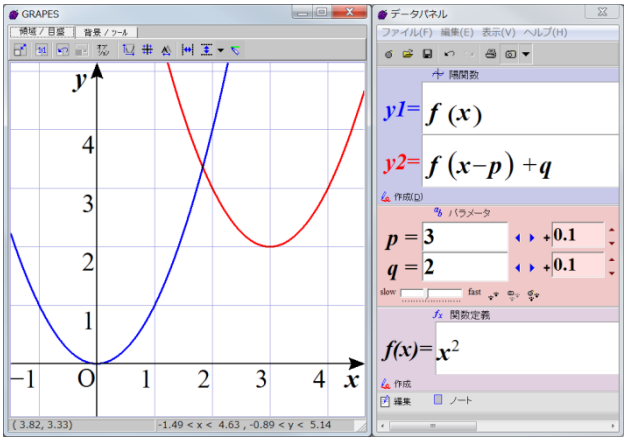
2. 関数電卓で式入力

定義関数は，

$f, g, h, f1, \cdots, f5$ の

８個あります

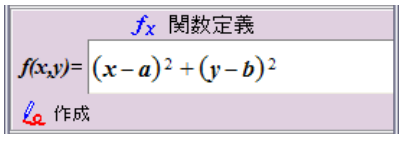




２つの関数　$y = f(x)$ ，　$y = f(x-p)+q$　のグラフ

- **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。

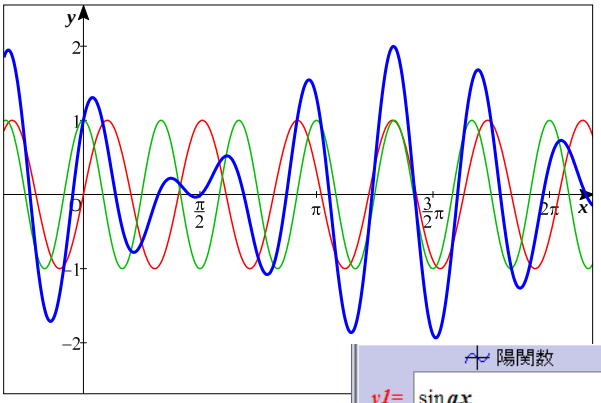


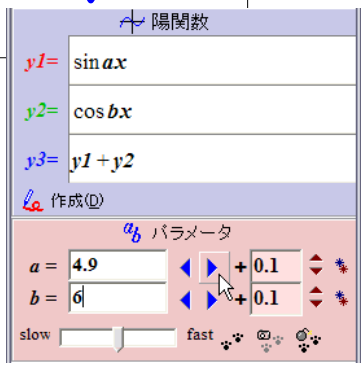
２変数関数を参照するときは，“ $f(2x, k)$ ”のように，２個の引数を与えます。なお，第２引数は省略することができ，この場合，$y = 0$ として扱われます。つまり，$f(x) = f(x, 0)$ です。

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。

8-2 陽関数を参照する

- **関数の参照**





右図は，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。

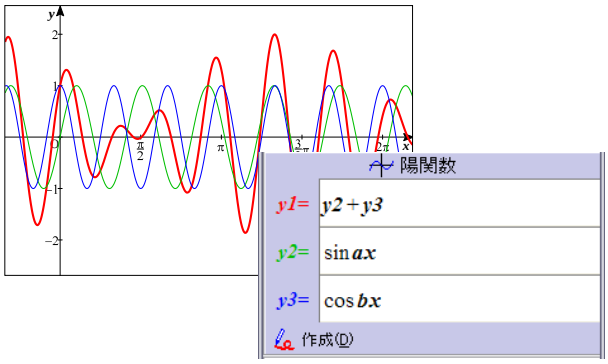
このような場合には，

$$y3=y1+y2$$

とした方が，処理が高速になります。

また，別の関数の合成を調べるときに，$y1$ や $y2$ だけを書き換えればよいので，データの再利用が容易になります。

- **参照の順序**



関数を参照するとき，その順序に注意を払う必要はありません。

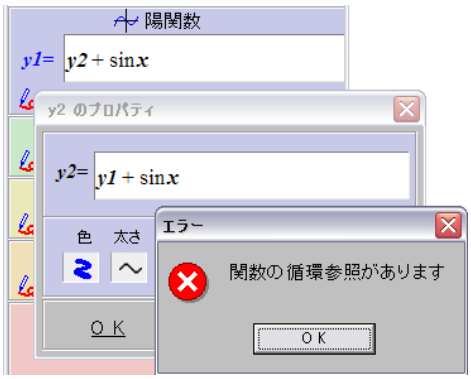
例えば，

$$y1=y2+y3$$

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

としても，同じように描かれます。

- **循環参照**



どのような参照も自由にできるわけではありません。関数の循環参照には十分気をつける必要があります。循環参照が見つかった場合には，エラーメッセージが表示され，入力はキャンセルされます。

陰関数からも陽関数を参照することができますが，この場合，速度は低下します。
また，基本図形や曲線から陽関数を参照することはできません。

8-3 グラフをドラッグする　－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。

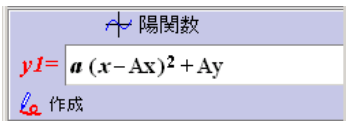
- グラフをドラッグする

1. 点 A を打つ

　点の打ち方については「5-7 点とドラッグ」を参照してください。

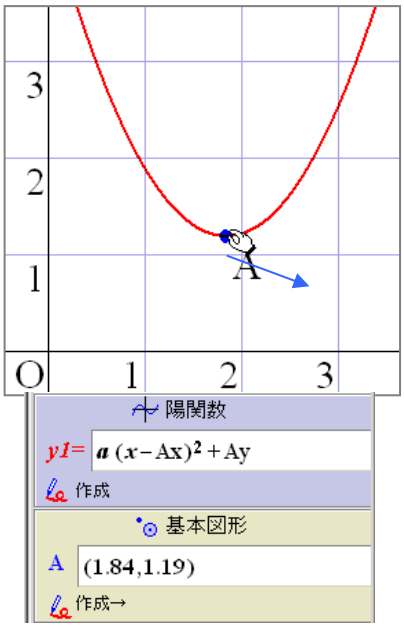
2. 点の座標を使って２次関数のグラフを描く

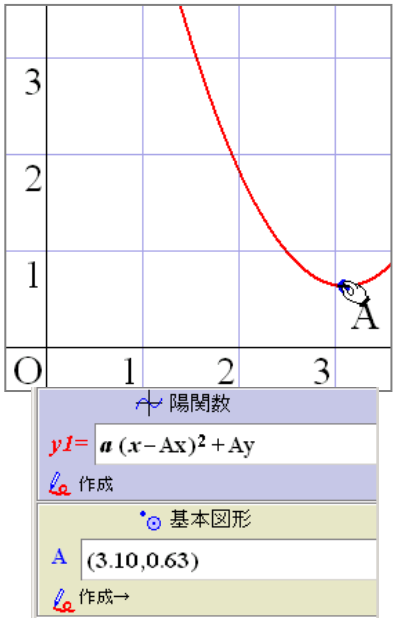
　点 A の座標は，(A.x , A.y) で表されます。
したがって, 点 A を頂点にもつ２次関数は,
$y = a(x - \mathrm{A.x})^2 + \mathrm{A.y}$ です。



3. 点 A をドラッグ

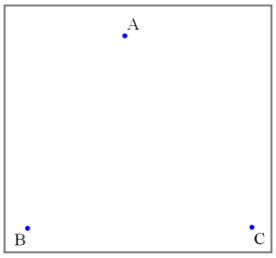
　マウスのボタンを離すとグラフが移動します。





8-4 図形を連動させる

点の座標は他の関数から参照することができます。例として，三角形とその重心を描いてみます。頂点を動かすと，重心も動きます。



- **基本図形の成分の参照**

1. **頂点 A, B, C を打つ**

   ３つの点 A, B, C を打ちます。

2. **頂点 A, B, C を結ぶ**

   辺は必要ありませんが，あった方が三角形らしいでしょう。

   三角形の描き方については，「7-3 折れ線や多角形を描く」を参照してください。

3. **重心 G を定める**

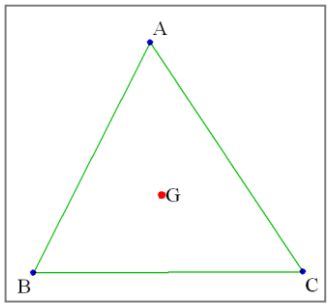
   点 A, B, C の座標は，

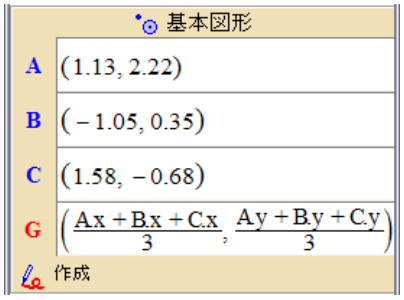
   (A.x , A.y), (B.x , B.y), (C.x , C.y)

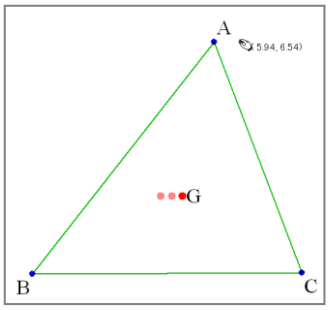
   ですから，重心 G の座標は，

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

   です。







4. **頂点を動かす**

   重心 G も動きます。

三角形 ABC の重心は，ベクトルを用いれば簡潔に表すことができます。これについては，「第 9 章ベクトルの利用」で説明します。

第９章　ベクトルの利用

9-1 点とベクトル

- **点とベクトル**

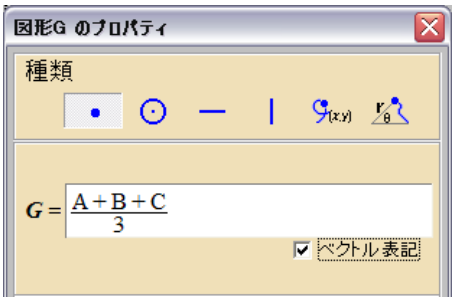
三角形 ABC の重心 G は，座標を用いて

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

と表されます。これを位置ベクトルで表現すれば，

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

ですが，同等のことが GRAPES でもできます。



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

です。GRAPES では，点と位置ベクトルは同一視されています。

- **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)

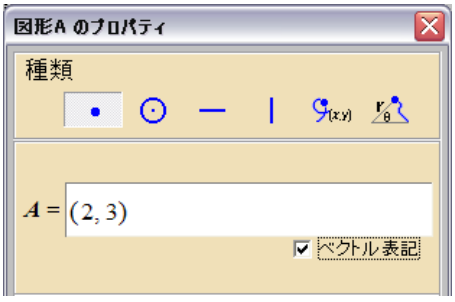
と表します。これは，座標の表記と同じです。

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

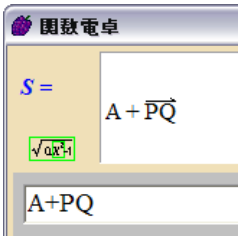
(2 , 3).y = 3

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 $\overrightarrow{PQ}$ を表します。



9-2 ベクトル表記で入力する

ベクトルを用いると，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。

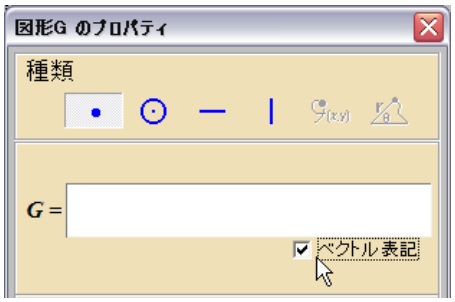
- **点の位置をベクトルを用いて入力する**

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. ［ベクトル表記］チェックボックスをチェック

   ２つあった式表示窓が１つになります。



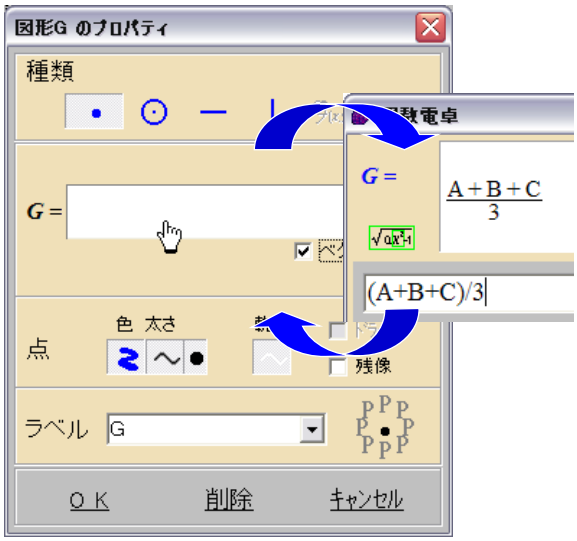
3. 式表示部分をクリック

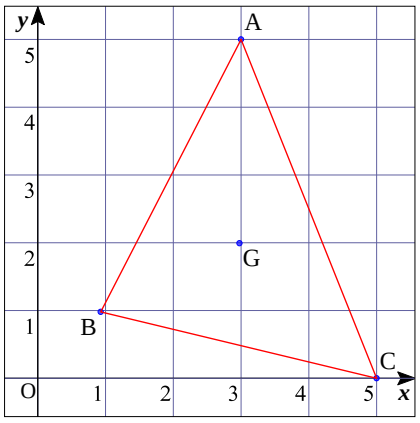
4. 関数電卓で式入力

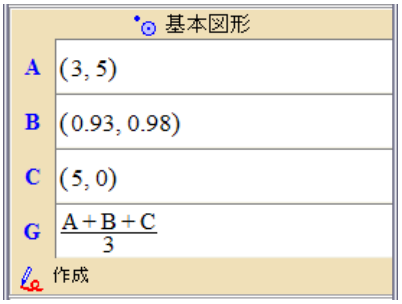
   関数電卓では，ベクトルの式を扱うことができます。右の例は，三角形 ABC の重心を入力しているところです。

   なお，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］をクリック







9-3 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**

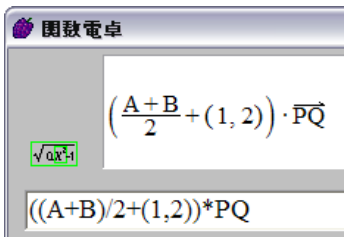
２つのベクトルの内積を求めるには，演算記号 ・ を使います。

キーボードから入力するときは，　* を使います。

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

- **混合演算**

括弧を使えば，複雑な計算にも対応できます。

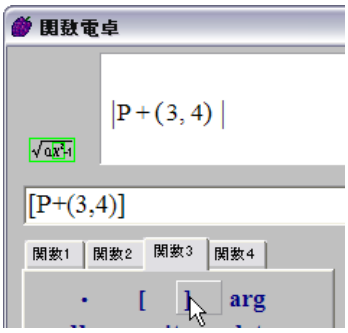


- **ベクトルの大きさ**

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。

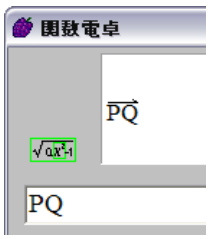
キーボードから入力するときは［］を使います。

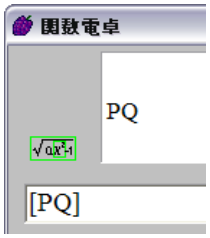
例：$|(3,4)|$　：ベクトル $(3,4)$ の大きさ



- **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。





9-4 距離と角度

- 距離を与える関数 len

len (P , Q)　　：２点 P, Q の距離

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。

len (P , (2 , 3))　：点 P と点(2 , 3) の距離

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。

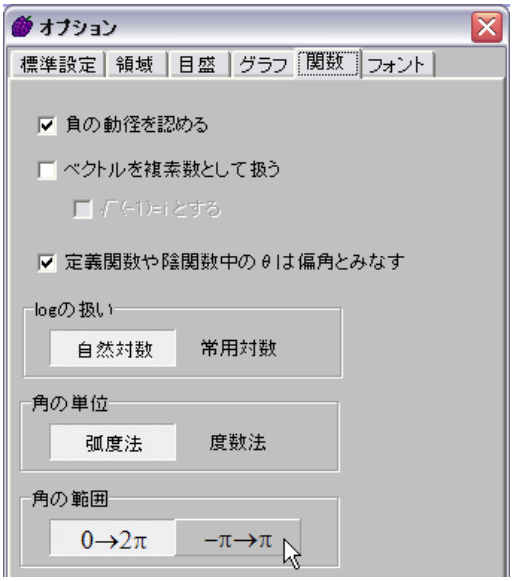
- 角度を与える関数 arg

arg (P , Q , R)：∠PQR の左回りの大きさ

arg (P , Q)　　：∠POQ

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 2π までです。角の単位や範囲は，オプションウィンドウの［関数］設定で変えることができます。



- 行列式の値

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。

9-5 位置を与える関数

- **線分の分点を与える関数“分点”**

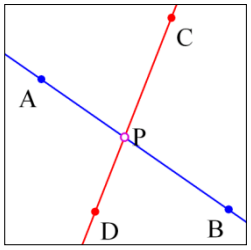
  関数“分点”は，線分の内分点や外分点を与えます。

  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

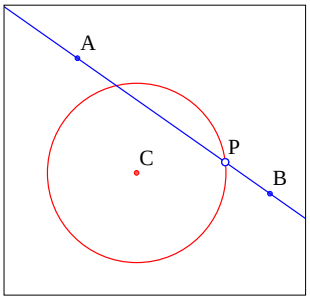
  交点(A , B , C , *r*) : 直線 AB と中心 C 半径 r の円の交点

  関数“交点”は，２直線の交点を与えます。

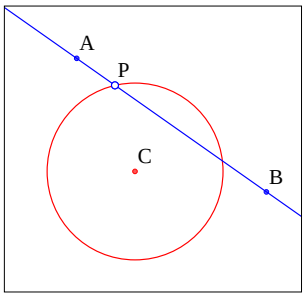
  キーボードから入力するときは intr を使います。

  ☆ 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*)



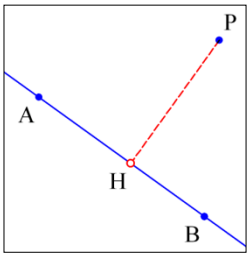
交点(B , A , C , *r*)

  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

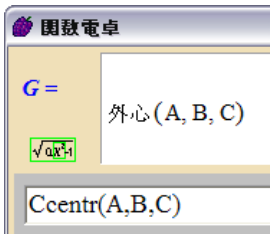
  関数“垂足”は，点から直線に下ろした垂線の足を与えます。

  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

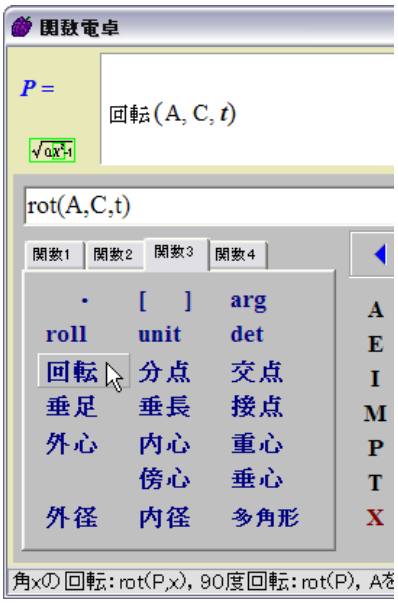
関数“重心”は三角形の重心を与えます。

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，オプションウィンドウの［関数］設定を用います。

- **単位円上の点**

roll(t)：単位円上の点 $(\cos t , \sin t)$ 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， $a\,\mathrm{roll}(\theta)$ 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$(t, \mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_n)$ ： n 角形の辺上の点。$0 \le t < n$ で多角形を一周する。（$n \le 20$）

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$(k-1, \mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_n) = \mathrm{P}_k$

非整数のとき，辺上の点を表す。

$t = t_0 + k\,n$ （k は整数）のとき，

多角形$(t, \mathrm{P}_1, \cdots, \mathrm{P}_n)$＝多角形$(t_0, \mathrm{P}_1, \cdots, \mathrm{P}_n)$

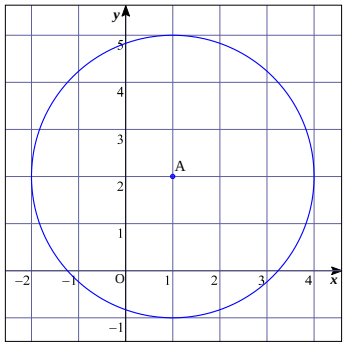
とくに，

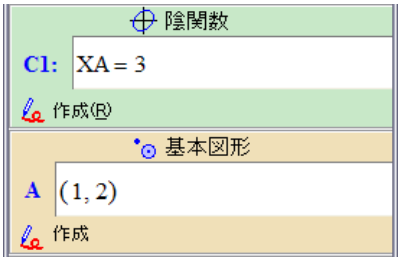
多角形$(n, \mathrm{P}_1, \cdots, \mathrm{P}_n)$＝多角形$(0, \mathrm{P}_1, \cdots, \mathrm{P}_n)$

- **X は 点(x, y) を表す**

陰関数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，陰関数で $\mathrm{XA} = 3$ とすれば，中心 A 半径 3 の円を表します。





9-6 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,

  $$f(4,5)=2\times4+3\times5$$

  ですが, $\mathrm{P}=(4,5)$ のときも，やはり,

  $$f(\mathrm{P})=2\times4+3\times5$$

  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,

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

  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。

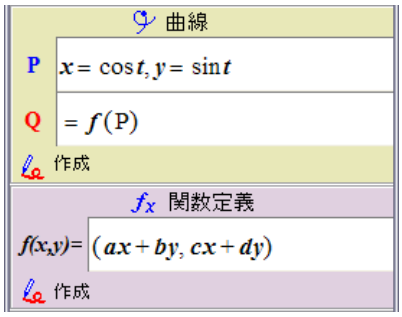
  例えば,

  $$f(x)=x\mathrm{A}+(1-x)\mathrm{B}$$

  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,

  $$f(x,y)=(a\,x+b\,y,c\,x+d\,y)$$

  は， 1 次変換です。

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

  $$\mathrm{Q}=f(\mathrm{P})$$

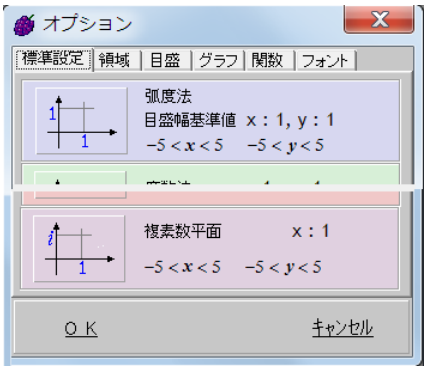
  とすれば，点 Q は点 P の像を表します。

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

9-7 ベクトルと複素数

ベクトルデータを複素数として扱かうことができます。この場合，ベクトル同士の演算や実数との間の演算が拡張されます。

- **複素数モード**



オプションウィンドウの［標準設定］ページで，複素数平面 を選択することで，ベクトルを複素数とみなして扱うことができます。この状態を「複素数モード」といいます。

複素数モードでは，実数は複素数（ベクトル）の部分集合として扱われます。

- **基本的な概念**

GRAPES での複素数は，ベクトルを拡張しているだけなので，ベクトルの演算をそのまま使うことができます。（内積を除く）

ベクトルの x 成分が複素数の実数部分に，y 成分が虚数部分に対応します。

複素数の四則や整数乗に対応するほか，指数関数や三角関数，双曲線関数が複素数を引数に持つように拡張されます。

- **複素数平面**

基本図形や曲線の「点」はベクトルなので，複素数モードではそのまま「複素数平面での点」として扱うことができます。

点 P などのベクトル変数に実数値が渡された場合は通常エラーとなりますが，複素数モードでは複素数平面上の点として意味を持ちます。

例： 通常モードでは， P = 2 はエラー

複素数モードでは， P = 2 は P = (2, 0) と同等。

9-8 複素数の演算と関数

- **四則演算と冪**

  和，差，積，商，冪は，複素数同士および複素数と実数の間で可能です。

  ☆　２つのベクトルの積（例：P * Q）は，内積を与えますが，複素数モードでは複素数の積を与えます。

  ☆　冪については，複素数の整数乗，正数の複素数乗が可能です。正数の複素数乗はオイラーの公式 $e^{i\theta}=\cos\theta+i\sin\theta$ に基づきます。角は弧度法に固定されます。

- **虚数単位**

  虚数単位 i を使うことができます。i = (0, 1) です。

  例：$a+b\,i=a+b\,(0,1)=(a,b)$

- **$\sqrt{-1}$ の扱い**

  $\sqrt{-1}$ を i として扱います（ディフォルト設定）。

  オプションウィンドウの［関数］設定で，［√(–1) = i とする］のチェックを外すと，Sqrt(負数) はエラーとなります。

- **比較演算子（等号，不等号）**

  比較演算子は実数に対してのみ利用可能です。複素数同士，あるいは複素数と実数の間で，比較演算子を使うことはできません。

- **複素数モードで拡張される関数**

  exp, log, sin , cos , tan, sinh, cosh, tanh, sol

  ☆　多価関数は扱っていません。

  ☆　ベクトルに対応している関数は，そのまま使うことができます。

  ☆　方程式の解を与える関数 sol は，虚数解も扱えるようになります。

  例：sol (x^4 = 16 , 1) = –2 , sol (x^4 = 16 , 2) = 2

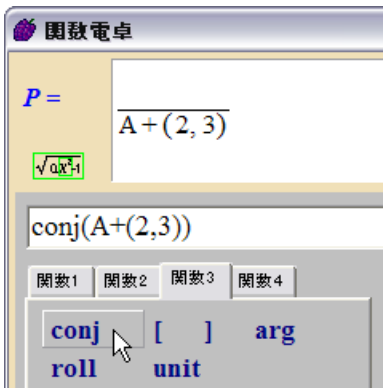
  sol (x^4 = 16 , 3) = (0, –2) , sol (x^4 = 16 , 4) = (0, 2)

- **共役複素数を与える関数**

  conj

  例：P = (1, 2) のとき conj (P) = (1, –2)

  ☆　関数 conj は通常は関数電卓にありませんが，複素数モードにすると関数電卓の［関数３］内に［conj］ボタンが表示されます。

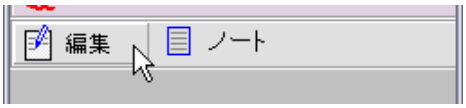


第 10 章　ステッカーとラベル

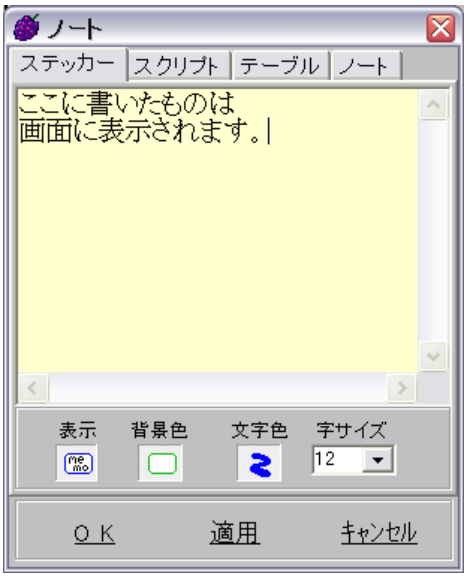
10-1 ステッカーの編集

プロジェクトに説明などをつけたいとき，ステッカーを使います。ステッカーには，テキスト文字のほか，式や式の値を表示することができます。

- **ステッカーの編集**



1. ノートエリアの［編集］をクリック



2. ノートウィンドウでステッカーを書く

書いたとおりに表示されます。

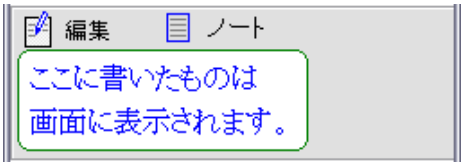
3. 文字色，背景色，文字サイズを選ぶ

4. 表示方法を選ぶ

枠なし，枠あり，背面あり，背面なし（透明），非表示から選びます。

5. ［OK］を押す

途中確認は，［適用］をクリックします。



- **ステッカーの特徴**

a. 表示できる内容

テキスト，式，式の値

b. 文字飾り

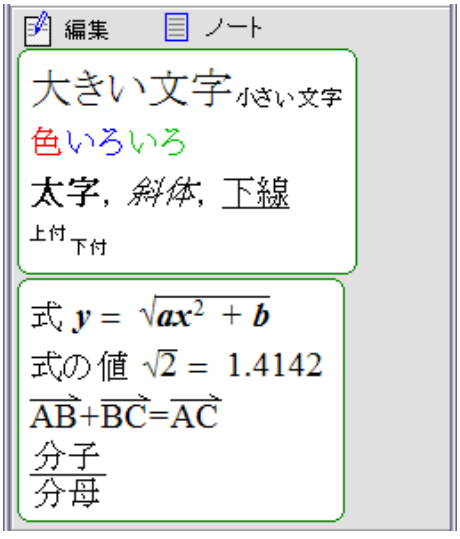
文字色，太字，斜体，下線，字サイズ，
シンボルフォント，上付文字，下付文字，
ベクトル記号，分数

c. 配置できる場所

グラフ表示エリアの任意の位置，データパネル

d. その他

最大 10 箇所に分けて表示することができる。



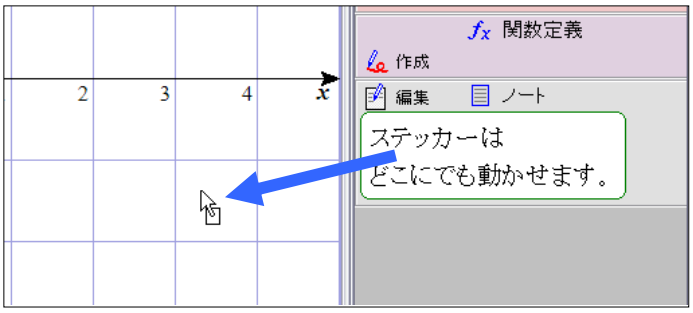
10-2 ステッカーの移動

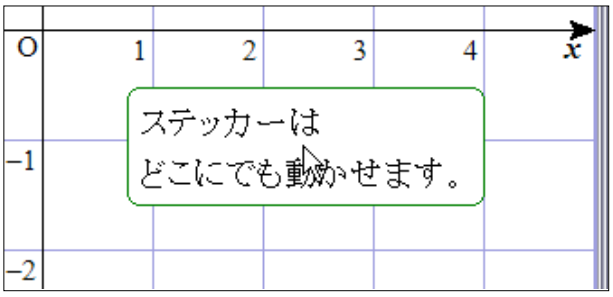
ステッカーはデータパネル上に作られますが，マウスのドラッグで，グラフ表示エリアに移動することができます。

- **グラフ表示エリアへの移動**

1. **ステッカー上で左ボタンを押す。**
2. **ドラッグして移動**
3. **左ボタンを離す。**

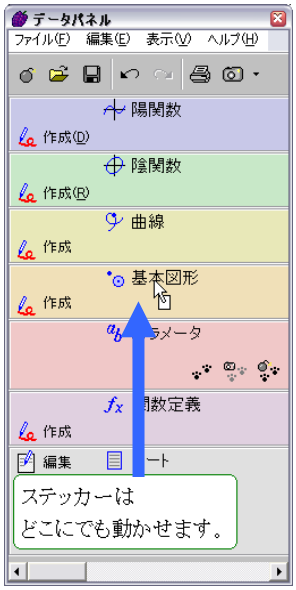
グラフ表示エリアからデータパネルへの移動も同様です。

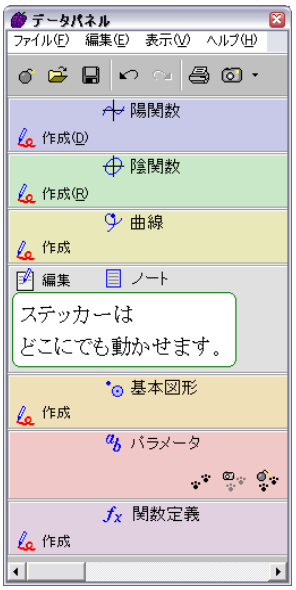




- **データパネル内の移動**

ステッカーやノートエリアのタイトル部分をドラッグすると，データパネル内の任意の位置に移動することができます。

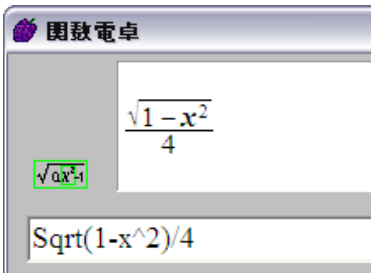




ノートエリア以外にもデータパネル内の任意のエリアは，タイトル部分をドラッグすると上下に移動することができます。

10-3 式や式の値を表示する

- **式テキストと式イメージ**



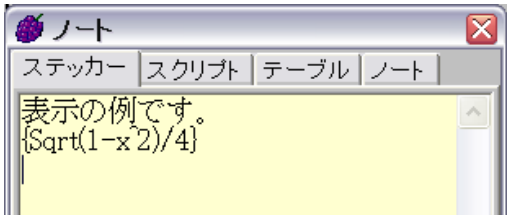
式をテキストで表現したものを式テキストといい，それの実際のイメージを式イメージといいます。

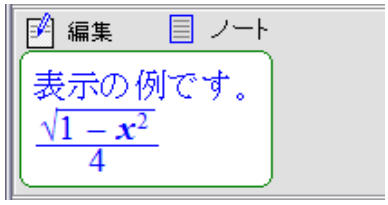
右関数電卓の例では，“Sqrt(1-x^2)/4”が式テキスト，

$\frac{\sqrt{1-x^2}}{4}$ が式イメージです。

- **式イメージの表示**

ステッカーやラベル中に式イメージを表示するには，式テキストを半角の中括弧｛｝で囲みます。





☆　ステッカー編集画面で右クリックしてポップアップメニューを表示し，関数式の挿入を利用することで，式テキストを簡単に挿入することができます。

（参照：15-1 ポップアップメニューの利用）

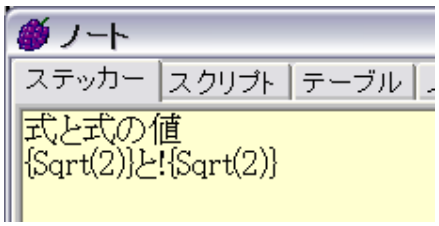
- **式の値の表示**

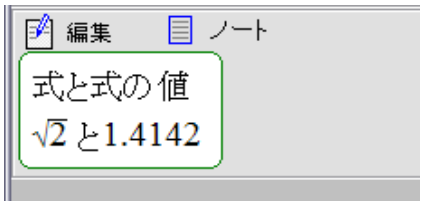
式テキストを !{｝で囲むと，式の値を表示します。

例：{Sqrt(2)} は $\sqrt{2}$ を表しますが，

!{Sqrt(2)} は $\sqrt{2}$ の値（1.4142）を表します。

☆　定数だけではなく，パラメータや点の座標を含んだ式の値を表示することもできます。





- **表示桁数**

式の値は，有効数字５桁の浮動小数点で表示されます。有効桁数を変えるには，式の後ろに“|有効桁数”を書き加えます。

例　!{Sqrt(a)|8}　：$\sqrt{a}$ の値を８桁の浮動少数点で表示します。

10-4 変化する式を表示する

- **係数が変化する式の表示**

パラメータ a , b の値が与えられているとき，これらパラメータを含んだ関数式を，ステッカーを用いて，具体的に表示することができます。例えば，$y=ax+b$ で $a=3, b=-2$ であれば，$y=3x-2$ と表示したいのです。

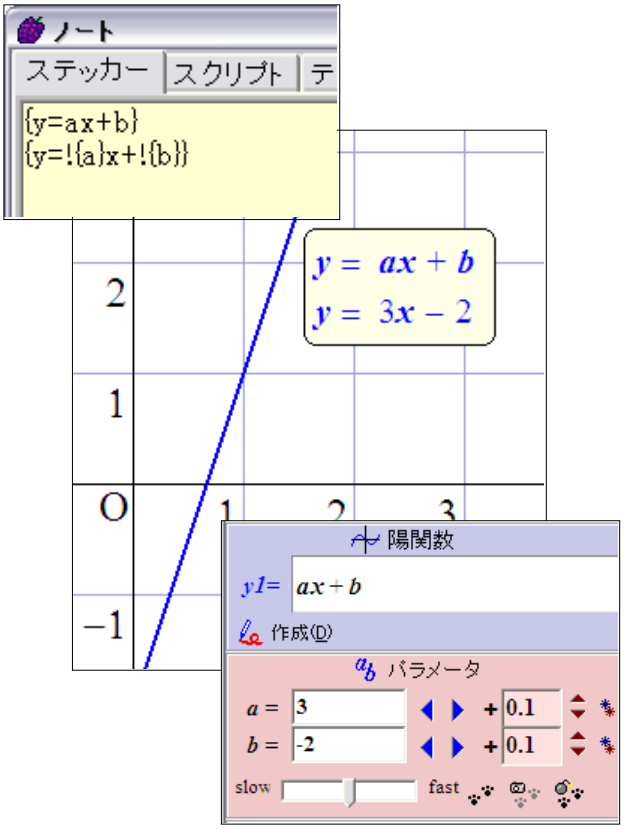
ステッカーに

{y=ax+b} と書けば，$y=ax+b$

という式が表示されます。a や b を値で表示するには，

{y=!{a}x+!{b}}

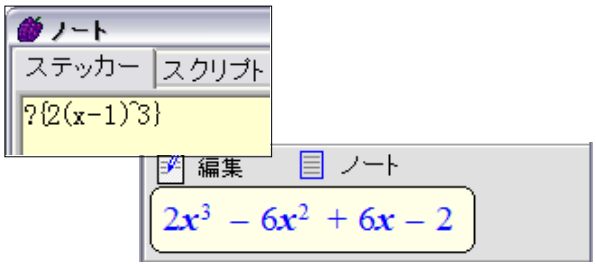
とします。なお，!{b}の前の符号“+”は，b の値によっては“－”に自動変換されて表示されます。



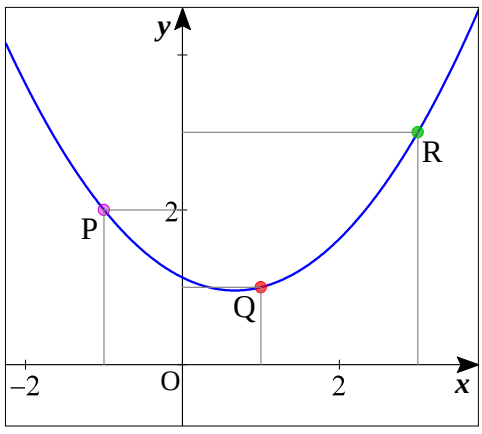
- **多項式の表示**

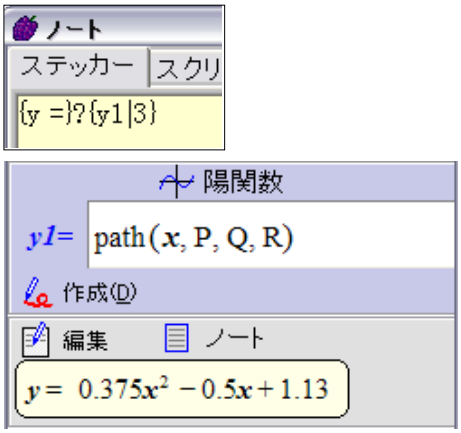
式が 6 次以下の多項式のとき，それを簡単に表示することができます。

例えば，関数 $2(x-1)^3$ の展開式を表示するには，?{$2(x-1)^3$} とします。



下図は，3 点 P, Q, R を通る 2 次関数 $y1=\mathrm{path}(x,\mathrm{P},\mathrm{Q},\mathrm{R})$ のグラフとその方程式です。

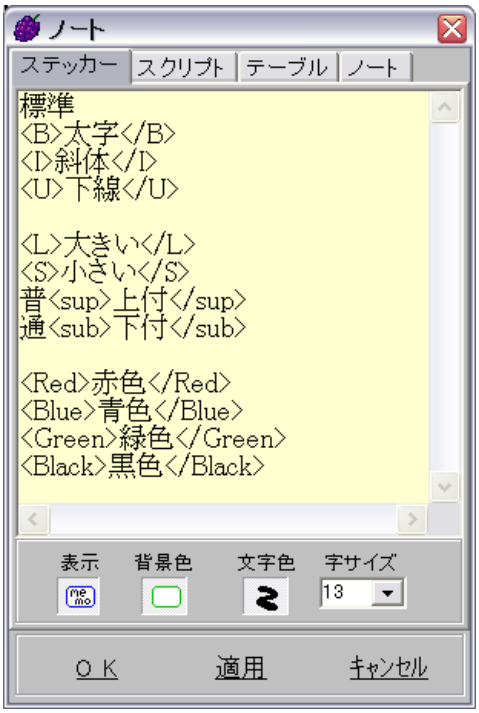


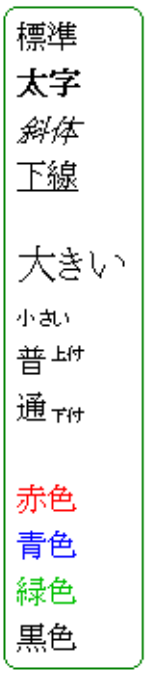


10-5 文字飾り

- **太字，斜体，下線**
  - 太字

    〈B〉と〈/B〉で挟まれた文字は，太字になります。(〈B〉は〈b〉でも可)
  - 斜体　〈I〉～〈/I〉
  - 下線　〈U〉～〈/U〉





- **文字サイズ**
  - 大きな文字にする。

    〈L〉～〈/L〉
  - 小さな文字にする。

    〈S〉～〈/S〉
  - 上付き文字にする。

    〈Sup〉～〈/Sup〉
  - 下付き文字にする。

    〈Sub〉～〈/Sub〉

- **文字色**

  文字色には，Black, Blue, Green, Red, Purple, Gray の 6 色があります。

  〈文字色〉と〈/文字色〉で挟まれた文字は，指定した文字色になります。

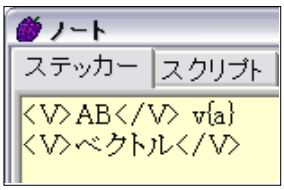
  ☆　任意色の指定には，〈color = #rrggbb〉～〈/color〉を利用します。rrggbb は，3 原色を 16 進数で指定します。例えば，赤なら FF0000，黄色なら FFFF00 です。

- **書式を元に戻す**

  〈Normal〉によって，すべての書式を標準に戻します。

- **ベクトル記号**

  〈V〉～〈/V〉　　v{ 文字列 } でも同じです。



- **シンボルフォント書式**

  〈Symbol〉～〈/Symbol〉　　ギリシャ文字などに利用します。

- **分数**

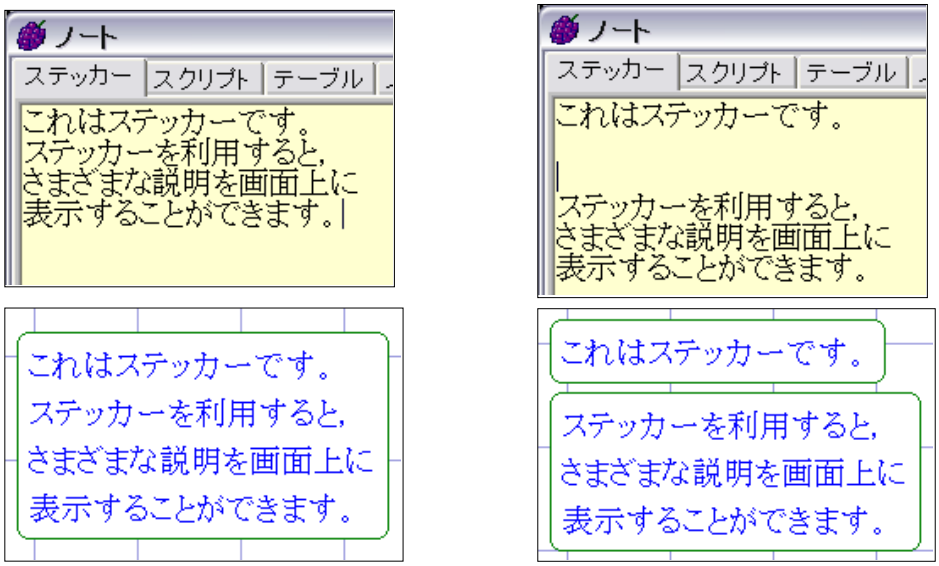
  〈Frac〉～ / ～〈/Frac〉

☆　ステッカー編集画面で右クリックして表示されるポップアップメニューを利用することで，各種書式を簡単に挿入することができます。(参照：15-1 ポップアップメニューの利用)

10-6 複数のステッカー表示

- **複数のステッカー表示**

  ステッカー文中の分割したい部分に，2 行以上の連続した空白行を挿入



☆　ステッカーの背景色や表示方法を，ステッカーごとに変える事はできません。

☆　全体の文字サイズや文字色もステッカーごとに変えることはできません。ステッカーごとに文字色などを変えたい場合には，前ページで示したような文字飾りを用いてください。

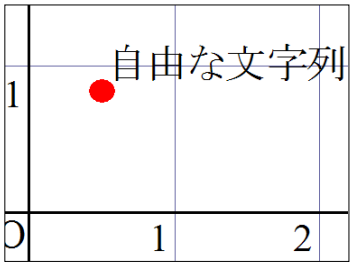
☆　最大 10 個までのステッカーを表示することができます。

10-7 ラベル

基本図形，曲線，連結図形にはラベルがありますが，これを用いれば，ステッカー同様に文字列や数式を表示することができます。

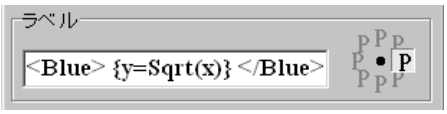
- **ラベル文字の変更**

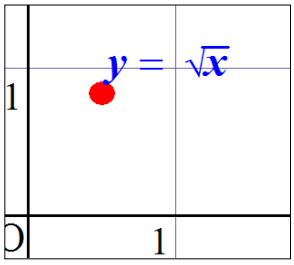
  ラベル文字を書き換える



- **文字飾りと数式**

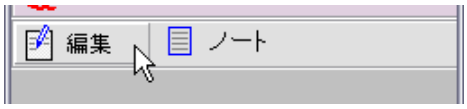
  ステッカーと同じ書式で，文字飾りや数式を書くことができます。





10-8 ノート

GRAPES のプロジェクトを作成したとき，ちょっとしたドキュメントを残したいときがあります。このようなときに［ノート］を使います。



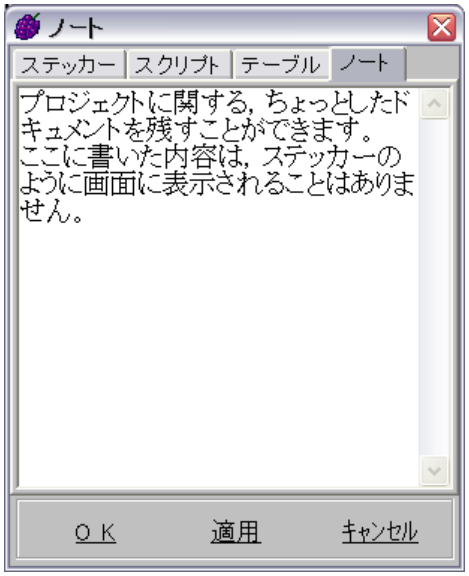
- ノートの編集

1. ノートエリアの［編集］をクリック

2. ノートウィンドウで［ノート］をクリック

3. 自由に書き込む

4. ［OK］をクリック



ノートに書いた内容は，ファイルに保存されますが，ステッカーのように画面に表示されることはありません。

10-9 フォントの設定

GRAPES6.70 以降では，言語の種類に依存することなくステッカーやスクリプトを扱うことができます。しかし，わずかの種類の unicode フォントを除いて，ほとんどのフォントでは，対応している言語は限られています。そのために，利用者が表示するフォントを選択できるようにしています。

- フォント設定ウィンドウの表示

領域パレットのオプションボタン[オプション]をクリックして，オプションウィンドウを表示し，［フォント］タグをクリック

- 編集用フォントの設定

  ステッカー，ノート，スクリプトを編集するときのフォントを指定します。

- ステッカーフォントの設定

  ステッカーは，英数文字とそれ他の文字で異なるフォントを指定することができます。

- フォントの初期化

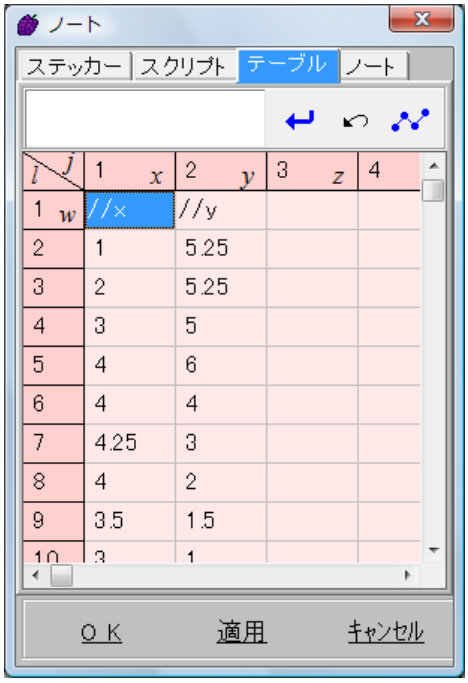
  ［初期化］をクリックすると，編集用フォントにはシステムの標準フォントが，ステッカーの英数文字（コード$1FFF 以下の文字）には Times New Roman が，ステッカーのその他の文字には MS P ゴシックがセットされます。

ここで設定したフォントは，GRAPES 終了時に各ユーザーのレジストリに記録されるので，次回の利用時にも同じ設定で使うことができます。ただし，個々のプロジェクトファイルには記録されません。

第 11 章　テーブル

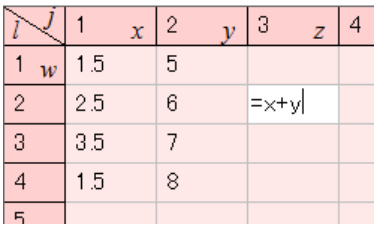
11-1 データ入力

GRAPES には 50 行×10 列～1000 行×500 列のセルを持つテーブルがあり，テーブル外の関数式からセルの値を参照することができます。



- **データ入力**

1. ノートエリアの［編集］をクリック
2. ノートウィンドウで［テーブル］をクリック
3. セルにデータを入力
   - 入力できるのは，数値だけです。
   - 関数式を入力した場合，計算結果がセルの値になります。この場合の関数式は，GRAPES で認識できるものに限られます。（参照：「式テキストと式イメージ」）
     計算結果が 2 次元データの時には，右隣のセルに y座標が入力されます。
   - ダブルスラッシュ「//」やダブルクォーテーション「”」で始まる文字列は，注釈文として，そのまま表示されます。
   - データの削除は，Del キーで行います。
   - 編集状態でダブルクリックすると，関数電卓を利用してセルの値を入力することができます。
   - 関数式や関数電卓を利用するとき，l は行番号，j は列番号，x は編集セルと同じ行の 1 列目のセルの値，y は 2 列目のセルの値，z は 3 列目のセルの値を表します。
     例：右図で，編集セルが 2 行 3 列の位置にあるとき，
     $l=2, j=3, x=2.5, y=6$ （$z=8.5$）



4. ［OK］を押す

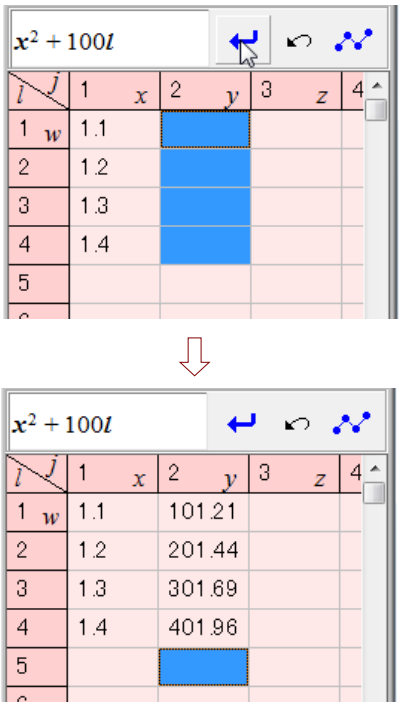
途中確認は，［適用］をクリックします。

☆　［適用］または［OK］が押されるまで，テーブルデータは確定しません。すでに描かれているグラフがテーブルデータを参照している場合，これらボタンが押されてデータの入力・変更が確定するまで，グラフは更新されません。

- **一括計算**

  選択領域を対象に一括計算を行います。

1. **テーブル上部の式入力窓をクリックして関数式を入力**
   - 式の中で，l はブロック中の各セルの行番号を，j は列番号を表します。
   - x は各セルと同じ行の 1 列目のセルの値，y は同じ行の 2 列目のセルの値，z は同じ行の 3 列目のセルの値を表します。
   - w は同じ列の 1 行目のセルの値を表します。
   - 他のセルを参照するには，Cells(行,列)を利用します。
2. **計算ボタン** **をクリック**
   - 選択領域の左上から右下に向けて，行方向に計算を実行します。
   - 計算結果が 2 次元データの時には，右隣のセルに y 座標が入力されます。
   - 計算後，選択領域のすぐ下のセルが，編集セル（選択領域）になります。



［一括計算例］

- **編集セルの移動**
  - カーソルキーで移動します。
  - Tab キーで右のセルに，Shift+Tab キーで左のセルに移動します。
  - Enter キーで下のセルに移動します。
  - マウスのクリックで任意のセルに移動します

11-2 テーブルの編集

- 行や列の選択

  行番号や列番号をクリック

  テーブル全体を選択するときは，テーブル左上のセルをクリック

- ブロック選択

  マウスのドラッグや Shift+カーソルキーの移動

  ☆　セルが編集状態の時にはマウスを使ったブロック選択ができません。この場合，ESC キーを押して編集状態を解除してください。

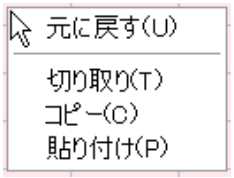
- 行データや列データの移動

  行番号や列番号をドラッグ

- コピー&ペースト

  右クリックのポップアップメニューから選ぶ

  - ショートカットキー（Ctrl+C でコピー, Ctrl+X でカット，Ctrl+V でペースト）も利用できます。
  - エクセルのシートとのコピー&ペーストもできます。



- 元に戻す

  Undo ボタン をクリック

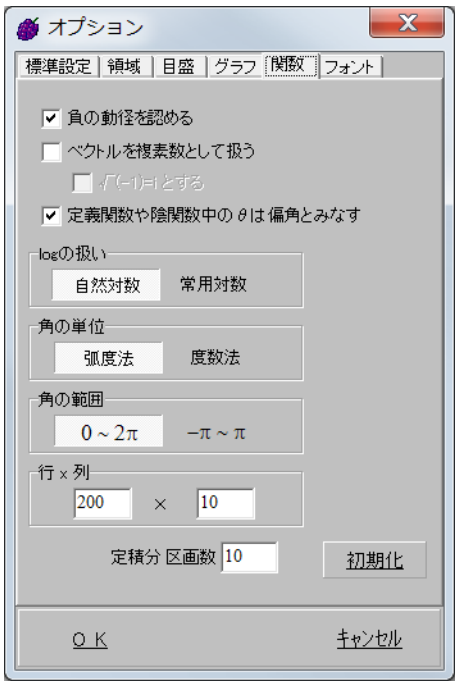
- 列幅の変更

  最上部のセルについて，セルの境界をドラッグ

11-3 テーブルサイズの変更

- テーブルサイズの初期値

  初期状態では，200 行×10 列　です。



- テーブルサイズの変更

  50 行×10 列から 1000 行×500 列までのサイズに変更することができます。

  1. オプションウィンドウの［関数］タグをクリック
  2. 行数と列数を入力

     行数は 50～1000，列数は 10～500 の整数です。
  3. ［OK］をクリック

- プロジェクトの保存や読み出しとテーブルサイズ

  プロジェクトをファイルに保存する際，テーブルサイズも保存されます。このファイルを読みだすと，（それまでのテーブルサイズにかかわらず）ファイルで設定されたテーブルサイズになります。

11-4 テーブルデータの利用

- **データの読み出し**
  - Cells(行，列)　：　**指定したセルの値**

    行や列には定数だけではなく，式を与えることもできます。

    無効なセル位置や値のないセルを指定したとき，エラー値を返します。

    （セルは，実数値とエラー値をとります。）
  - CellsP(行，列)　：　**指定したセルとその右隣のセルの値を成分とするベクトル**

- **統計関数**
  - CountC(*j*) : *j* 列データの個数
  - SumC(*j*) : *j* 列データの総和
  - AverageC(*j*) : *j* 列データの平均
  - StdvC(*j*) : *j* 列データの標準偏差
  - MinC(*j*) : *j* 列データの最小値
  - MaxC(*j*) : *j* 列データの最大値
  - QrtlC(*j*, *n*) : *j* 列データの第 *n* 四分位数
  - CorrelC(*i* , *j*) : *i* 列データと *j* 列データの相関係数

- **散布図の作成**

テーブルデータの 2 つの列をそれぞれ *x* 座標と *y* 座標とする散布図を描きます。

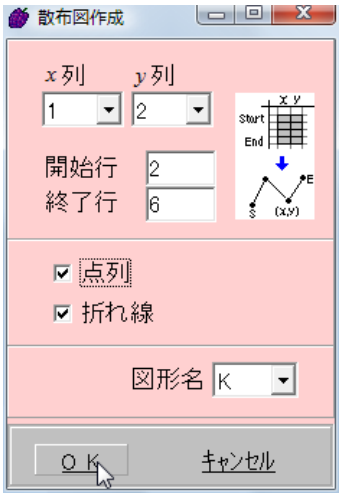
1. **グラフ化するセル範囲を選択**
2. **グラフ作成ボタン** **をクリック**

   散布図作成ウィンドウが表示されます。
3. **点列と折れ線のそれぞれについて，不要な場合にはチェックを外す。**

   どちらも選択しない場合，グラフは描かれません。

   また，上記 1 で選択したデータの範囲をここで変更することもできます。
4. **図形名を指定**

   図形は「曲線」の中の「点列」を利用して描かれます。



5. ［OK］をクリック

曲線のプロパティ画面が表示されます。

6. **点や線の色・太さなどを設定して［OK］をクリック**

☆　このようにして作成されたグラフは，テーブルを参照しますが，編集中のテーブルデータはグラフに反映されません。編集中のデータをグラフに反映させるには，ノートウィンドウ下部の［適用］をクリックします。

11-5 スクリプトとテーブルデータ

スクリプトを用いると，数列などの一連の計算結果をセルに代入することができます。

- **セルへの代入**

  Cells(行, 列) := 式　　（行や列は，定数以外に式を入れることもできます）

  例：　Cells(1,2) := 4

  ☆ 右辺の値が 2 次元データの場合，指定したセルとその右隣のセルに値が代入されます。
  例えば，Cells(1,2) := (4,5) は，Cells(1,2) := 4 , Cells(1,3) := 5 と同じ結果になります。
  右辺の値が 2 次元データの場合，CellsP(行, 列) := 式　を使うこともできます。

- **全セルの初期化**

  ClrAllCells

  テーブルの全セルの値を消去します。

- **列データに関する操作**

  ClrColumn(*i*)　　：*i* 列の全セルの値を消去します。

  TrimColumn(*i*)　　：*i* 列の無効データを削除し，セルを上に詰めます。

  TrimColumns(*i* , *j*)　：*i* 列と *j* のデータをセットトみて，一方でも無効データが含まれているときは，同じ行の他方のデータも削除し，上にセルを詰める。

  SortColumn(*i*)　　：*i* 列のデータを昇順にソートする。

  SortDesColumn(*i*)　：*i* 列のデータを降順にソートする。

- **サンプル**

フィボナッチ数列の 20 項目までをセルに代入します。

```
Cells( 1 , 1 )  :=  1
Cells( 2 , 1 )  :=  1
For  n := 3  to  20
        Cells( n , 1 )  :=  Cells( n−1 , 1 ) + Cells( n−2 , 1 )
Next
```

Cells(行，列) では列番号を省略することができます。この場合，列番号は 1 となります。

```
Cells(行) = Cells(行，1)
```

これを利用すれば，上記のスクリプトは，

```
Cells( 1 )  :=  1
```

```
Cells( 2 )  :=  1
For  n := 3  to  20
        Cells( n )  :=  Cells( n−1 ) + Cells( n−2 )
Next
```

となります。

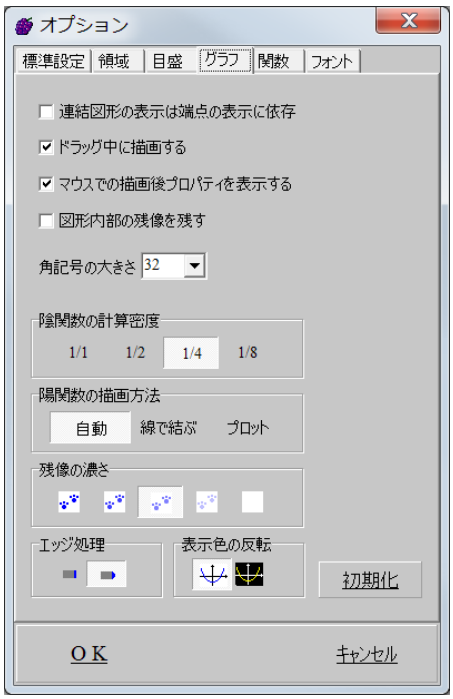
スクリプトの詳細は，「第 14 章　スクリプト」をお読みください。

第 12 章　関数を使いこなす

12-1 グラフの詳細設定

- グラフ設定ウィンドウの表示

領域パレットのオプションボタン[オプション]をクリックして，オプションウィンドウを表示し，［グラフ］タグをクリック



- **連結図形の表示**

　これをチェックすると，連結図形を構成している点のうちひとつでも非表示のものがあると，その連結図形は表示されなくなります。

- **ドラッグ中の描画**

　点をドラッグした場合，ドラッグしているうちからその点を参照しているグラフが描画されます。このチェックをは外すと，ドラッグが終了してから描画されます。

- **陰関数のグラフは計算中にも描画する**

　陰関数は描画に時間がかかるため，計算しながら描いています。これのチェックを外すと，計算終了後に一気に描くようになり，ちらつきがなくなります。

- **マウスでの描画後のプロパティ表示**

　点や線分をマウスのドラッグで描いたとき，強制的にプロパティ画面が開くようになっていますが，このチェックを外すと，表示されなくなります。

- **図形内部の残像を残す**

　円や曲線などで図形内部のペイントを行っても，それを残像として残すことはできませんが，これをチェックすることで，内部のペイントも残像として残ります。

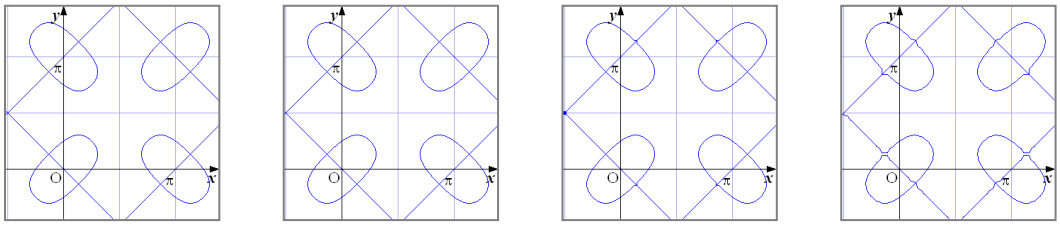
- **角の弧の半径**

　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。

- **陰関数の計算密度**

一般の陰関数では，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。（参照：「4-4 陰関数はどうやって描いてるの？」）

1/1　1/2　1/4　1/8

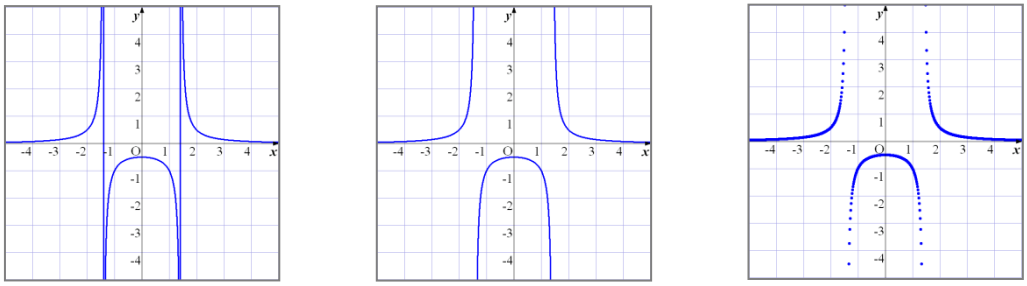


▲ $\sin x+\sin y=\sin 3x+\sin 3y$ のグラフ

- **陽関数の描画方法**

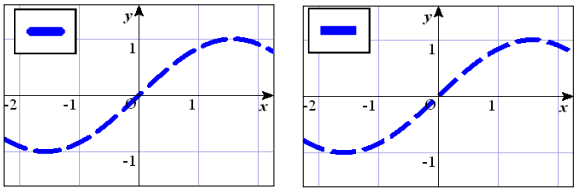
陽関数のグラフでは，x に対する y の値を計算してグラフを描いています。このとき，これらの点をすべて線で結んで連続したグラフにするか，そのまま点をプロットするか，一定の条件のもとでのみこれらを結ぶようにするかを選択することができます。

線で結ぶ　自動　プロット



- **線のエッジ処理**

点線や破線などでグラフを描くときの端点の形状を指定することができます。



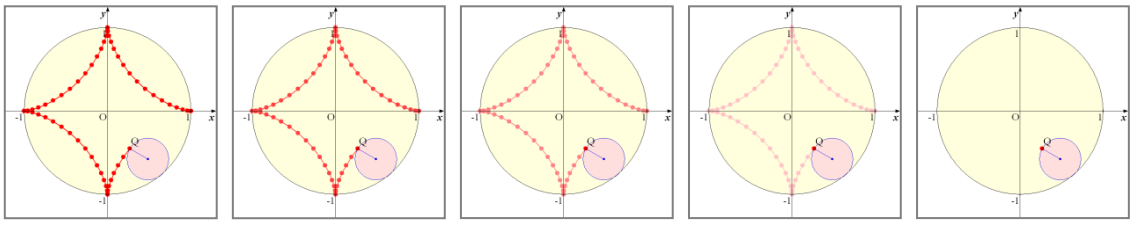
- **表示色の反転**

グラフや目盛の色を反転して表示します。
黒板への直接投影などの際に利用します。背景画像とマーカー色は反転しません。

- **残像の濃さ**

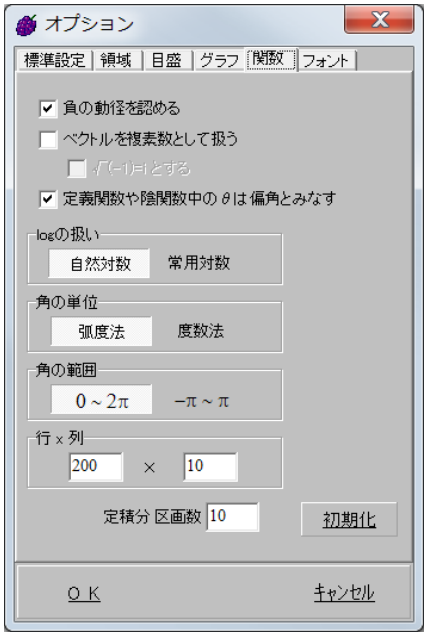
残像や軌跡の色は，元になるグラフ色を薄くしたものですが，濃さを変えることができます。

100%　75%　50%　25%　0%



12-2 関数の詳細設定

- **関数設定ウィンドウの表示**



領域パレットのオプションボタン[オプション]をクリックして，オプションウィンドウを表示し，［関数］タグをクリック

- **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**

これをチェックすると，ベクトルを複素数平面上の点として扱えるようになります。詳細は，「9-7 ベクトルと複素数」を参照してください。

- **定義関数や陰関数中のθは偏角とみなす**

陰関数や定義関数中では，θ は点 (x, y) についての偏角を表します。しかし，θは陽関数や基本図形ではパラメータとして働きます。もし，陰関数や定義関数でもθ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$0 \le \theta < 2\pi$ 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$-\pi \le \theta < \pi$のとき，「角」は常に劣弧（小さい方の角）を表示します。

- **定積分関数 ∫(積分変数，下限，上限，式) の区画分割数**

定積分は，4 次の近似式を用いています。この近似式は 9 次以下の多項式において理論的な誤差を生じませんが，積分区間を細かく分割して個々の小区間にこの近似式を適用することで，一般の関数に対しても非常によい精度で計算することができます。

12-3 GRAPES の関数

- **左関数**

通常の関数のことで，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
- $\mathrm{A}\sin x, \mathrm{A}\cos x, \mathrm{A}\tan x$ ：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ウィンドウの［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。これを変更はオプションウィンドウの［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）
- $\mathrm{int}(x)$ ：整数部分（ガウス記号と同等）
- round(x)：四捨五入
- frc(x) ：小数部分　$\mathrm{frc(x)} = \mathrm{x} - \mathrm{int(x)}$
- $|x|$もしくは abs(x)：絶対値　（| | のキー入力は [ ] ）
- $\mathrm{sign}(x)$ ：符号
- rnd(x) ：0 以上 x 未満の整数の乱数

  ただし， $x=1$の場合は，0 から 1 の間の実数の乱数を与えます。
- $\Gamma(x), \mathrm{B}(x,y)$ ：ガンマ関数とベータ関数
- $f(x)$, $g(x)$, $h(x)$, $f1(x)$～$f5(x)$ ：定義関数
- $f'(x)$ ～$f5'(x)$ ：定義関数の導関数
- $f''(x)$ ～$f5''(x)$ ：定義関数の２次導関数
- $F(x)$～$F5(x)$ ：定義関数の不定積分。　$F(x)=\int_0^x f(t)\,dt$

  ただし， $f(x)$は 8 次以下の多項式に限ります。
- nCr(x,y) ：二項係数
- gcd(x,y)， 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, $f1$～ $f5$ も同様）

- $\mathrm{Max}(x_1, \cdots, x_n), \mathrm{Min}(x_1, \cdots, x_n)$：**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。

- Cells(*m*, *n*)　：**テーブルデータの読み出し。*m* 行 *n* 列のセルの値**

  無効なセルか，セルのデータが数値でない場合，エラー値を返します。

  列番号を省略した場合，1 列目の値を返します。Cells(*m*) = Cells(*m*,1)

- rgb(*r*, *g*, *b*)：**色を表す 24 ビットの整数値を与える。**

  r, *g*, *b* は Red, Green, Blue の強さを表す 0～1 の実数です。

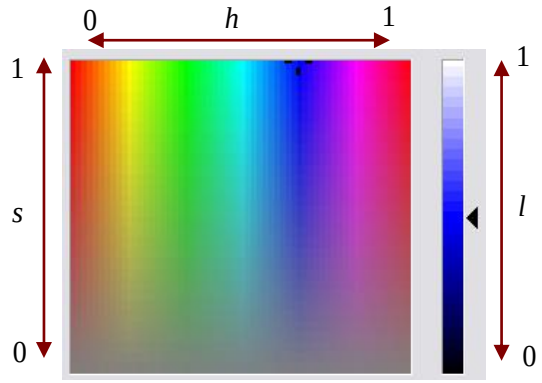
  例：rgb(1,0,0) は赤，rgb(1,1,0) は黄，rgb(0.5,0.5,0.5) はグレー

- hsl(*h*,*s*,*l*)：**色を表す 24 ビットの整数値を与える。**

  h, *s*, *l* は色相，彩度，明度を表す 0～1 の実数です。

  h については周期関数となります。hsl(*h*+1 , *s* , *l*) = hsl(*h* , *s* , *l*)

  例：hsl(0,1,0.5) は赤，hsl(0,1,0.8) は薄い赤，hsl(0,0,0.5) はグレー



- area(*x*)：**表示領域**

  表示領域の左下 (area(0) , area(1)) ，右上 (area(2) , area(3))

- areasize(*x*)：**表示領域のピクセル単位の大きさ**

  幅 areasize(0) ，高さ areasize(1)

- Pressed(Virtual key code)：**指定したキーが押されているかどうかを調べる。**

  仮想キーコードで与えたキーが押されていれば true（1）を，押されていなければ false（エラー値）を返す。

  仮想キーコードとは，windows システムで定義されている整数です。

キー	仮想キーコード
マウス左ボタン	1
Shift キー	16
Ctrl キー	17
左カーソルキー	37

12-4 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°：度数を弧度に変換（“° ”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

  - $y1$～$y20$：陽関数の値（陽関数や陰関数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 , θ ： $r = \mathrm{len}(x, y)$, $\theta = \arg(x, y)$ （陰関数や定義関数の中でのみ使用可）
  - mousePos：マウス位置の座標 （x 座標は mousePos.x，y 座標は mousePos.y）

- 関数やパラメータ文字を引数に持つ関数

  - $\mathrm{sol}(f(x) = g(x) , n)$：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。
  - $\mathrm{cof}(n, f(x))$： $f(x)$が 6 次以下の整式のとき，n 次の係数を与える。

    例： $\mathrm{cof}(2, (x-2)^3) = -6$
  - Σ(パラメータ，初期値，終了値，式)：級数の和 （“Σ”のキー入力は Sum）

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

    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。
  - Π(パラメータ，初期値，終了値，式)：級数の積 （“Π”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。
  - ∫(積分変数，下限，上限，式)：定積分 （“∫”のキー入力は Igr）

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

    積分変数にはパラメータや x を使います。

    なお，下限や上限に変数 x を含んだ式を代入することはできません。
  - stepsize(パラメータ)：パラメータの増減幅

12-5 ベクトル（点）に関する関数

- 実数値関数
  - path(x , P_1 , ⋯, P_n) ：与えられた n 個の点を通る(n–1) 次関数
  - | P | もしくは len(P) ：ベクトルの大きさ（“| |”のキー入力は [ ] ）
  - len(P_1 , P_2) ： ２点間の距離
  - arg(P) ：偏角　（角の範囲は［関数設定］で設定（$0 \le \arg < 2\pi$, $-\pi \le \arg < \pi$ ）します）
  - arg(P_1 , P_2)　　：角　$\angle P_1OP_2$

    arg(P_1 , P_2 , P_3) ： ３点の作る角　$\angle P_1P_2P_3$
  - det(a, b, c, d)　：行列式の値　$ad - bc$

    det(P , Q)　：行列式の値　P.x·Q.y − P.y·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 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 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 Irad）
  - 面積 (P_1 , ⋯, P_n) ：多角形の符号付き面積（“面積”のキー入力は areaS）

- ベクトル値関数
  - 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, ⋯, V, A, ⋯,N に対して，1, 2, 3, ⋯, 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。
  - 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 Gcentr）
  - 外心(A, B, C)　：三角形 ABC の外心（“外心”のキー入力は Ccentr）
  - 垂心(A, B, C)　：三角形 ABC の垂心（“垂心”のキー入力は Hcentr）
  - 内心(A, B, C)　：三角形 ABC の内心（“内心”のキー入力は Icentr）
  - 傍心(A, B, C)　：三角形 ABC の傍心（“傍心”のキー入力は 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 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 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° 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 , $\cdots$, P_n) ：*n* 角形の辺上の点。$0 \le t < n$ で多角形を一周する。（$n \le 20$ ）

  （“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。
- mousePos : マウス位置の座標
- centerPos : グラフ表示領域の中央の座標
- 定義関数とベクトルの関連については，「9-6 定義関数とベクトル」を参照してください。

12-6 テーブルに関する関数

- セルデータを読み出す関数
  - Cells(i , j) : i 行 j 列のデータ
  - CellsP(i , j) : i 行 j 列のデータとその右隣のデータを座標とした座標データ

    CellsP(i , j) = (CellsP(i , j), CellsP(i , j+1))

- 統計関数
  - CountC(j) : j 列データの個数
  - SumC(j) : j 列データの総和
  - AverageC(j) : j 列データの平均
  - StdvC(j) : j 列データの標準偏差
  - MinC(j) : j 列データの最小値
  - MaxC(j) : j 列データの最大値
  - QrtlC(j, n) : j 列データの第 n 四分位数
  - CorrelC(i , j) : i 列データと j 列データの相関係数

12-7 演算子と変数

- 演算子
  - ＋，－，×，／：四則演算
  - ・：実数や複素数の積，ベクトルの内積
  - ^：実数乗

    $x \wedge y = x^y$ 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
  - $a, b, c, d, k, m, n, p, q, s, t, u, v, \theta$

    θ は，陰関数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
  - θ：極方程式の偏角（陰関数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
  - π：円周率

12-8 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。

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

  これら関数の引数の範囲は次のように定めれています。
  - **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。

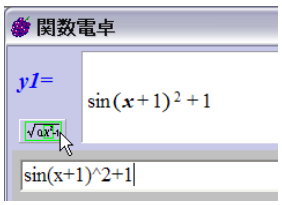
- **関数の引数範囲表示**

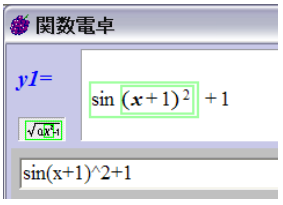
  上記の規則のほとんどは私たちが日常使っているものですが，曖昧です。

  GRAPES がどのように判断しているかを関数電卓上で確認することができます。

  関数電卓の引数範囲表示ボタン $\sqrt{\square}$ をクリック

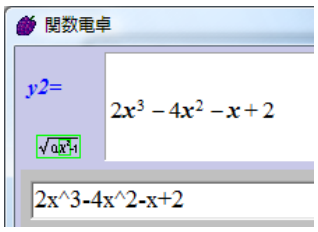
  引数の範囲が緑色の枠で表示されます。



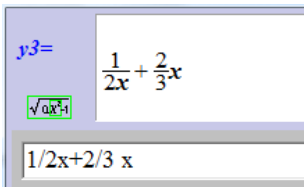


12-9 関数の入力例

(1) $y=2x^3-4x^2-x+2$

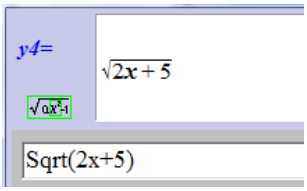


(2) $y=\frac{1}{2x}+\frac{2}{3}x$

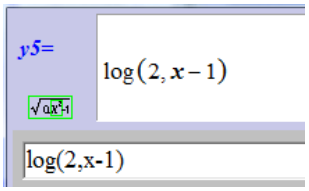


← 2 / 3 と x の間にスペースを入れます。

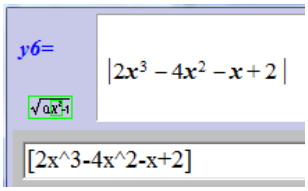
(3) $y=\sqrt{2x+5}$



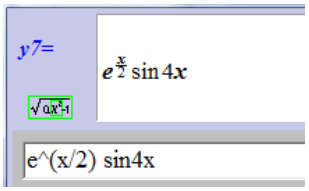
(4) $y=\log_2(x-1)$



(5) $y=\left|2x^3-4x^2-x+2\right|$



(6) $y=e^{\frac{x}{2}}\sin 4x$

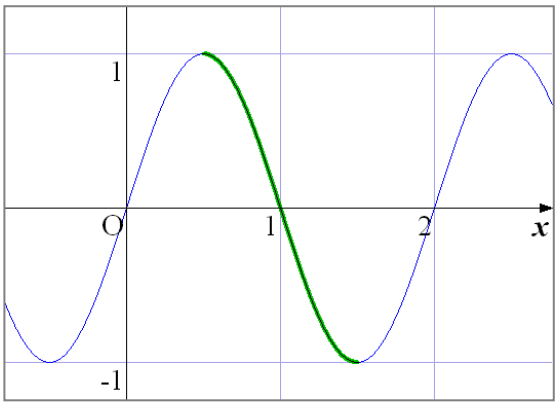


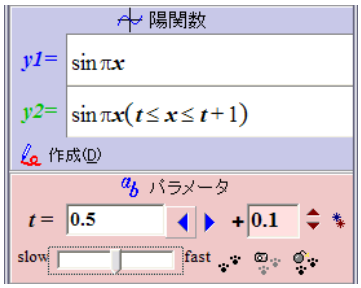
12-10 定義域の制限

- **定義域の制限**

関数の定義域を制限するには，関数式の後に定義域を付け加えます。

例： $y=x^2\ (x\geq 0)$ ， $y=\sin\pi x\ (t\leq x\leq t+1)$



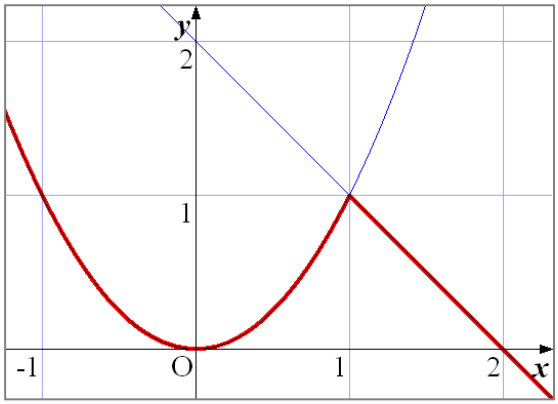


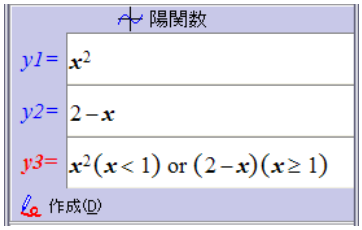
$t\leq x\leq t-1$ は，$(t\leq x)\ \text{and}\ (x\leq t+1)$ と書くこともできます。

- **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，

$y=x^2\ (x<1)\ \text{or}\ (2-x)(x\geq 1)$ のように論理演算子“or”を用いて結合します。





上の例では，$y3=y1(x<1)\ \text{or}\ y2(x\geq 1)$ と書くこともできます。

12-10 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**

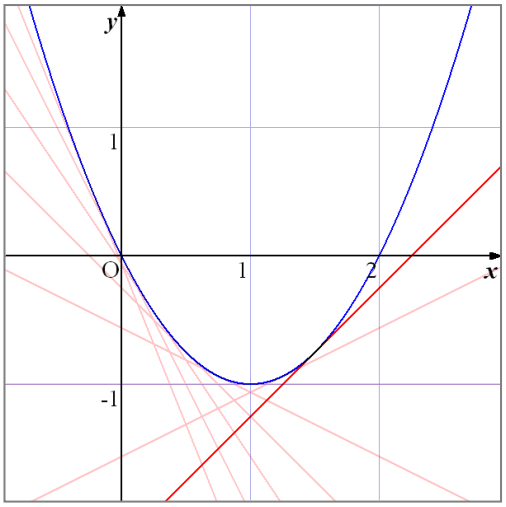
微分して使いたい関数を定義関数に登録します。

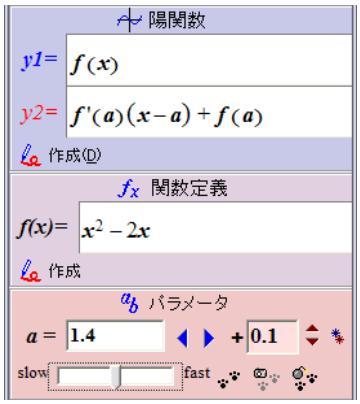
例えば，$f(x)=x^2-2x$ などとします。

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**

積分が可能な関数は，8 次以下の整関数に限られます。それ以外の関数の積分については，定積分ウィンドウや定積分関数を利用してください。

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。

第 13 章　関数を調べよう

13-1 座標と関数の表示

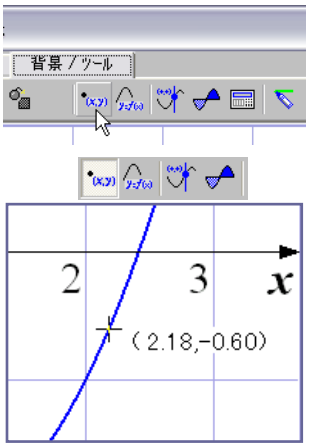
- **グラフや軌跡の座標表示**

1. ［ツール］パレットの座標ボタン をクリック

   座標表示ボタンを押し下げます。

2. マウスカーソルをグラフに重ねる

   グラフにマウスカーソルを重ねると，その点の座標が表示されます



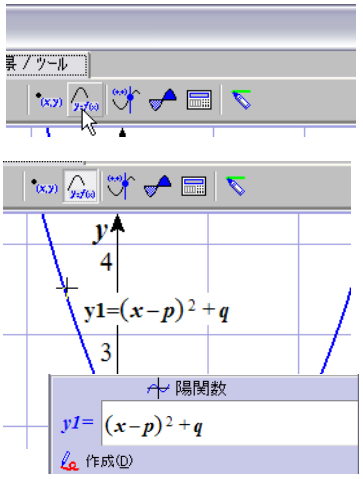
- **関数式表示**

1. ツールパレットの関数式ボタン をクリック

   関数式ボタンを押し下げます。

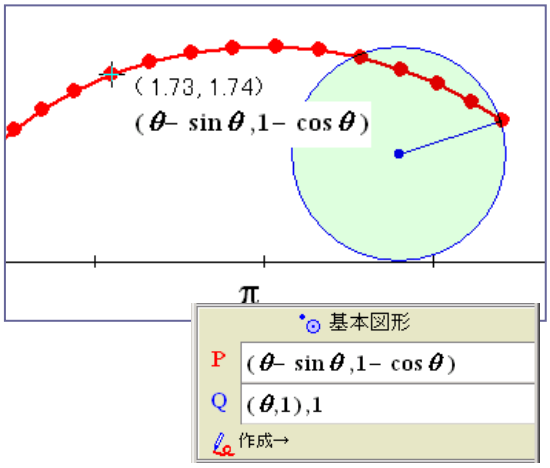
2. マウスカーソルをグラフに重ねる

   グラフにマウスカーソルを重ねると，そのグラフの方程式が表示されます。



- **軌跡の座標・式表示**

基本図形の軌跡や曲線の場合，残像や曲線にマウスカーソルを重ねても座標や式を表示します。

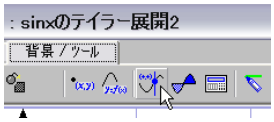


13-2 関数値ウィンドウ

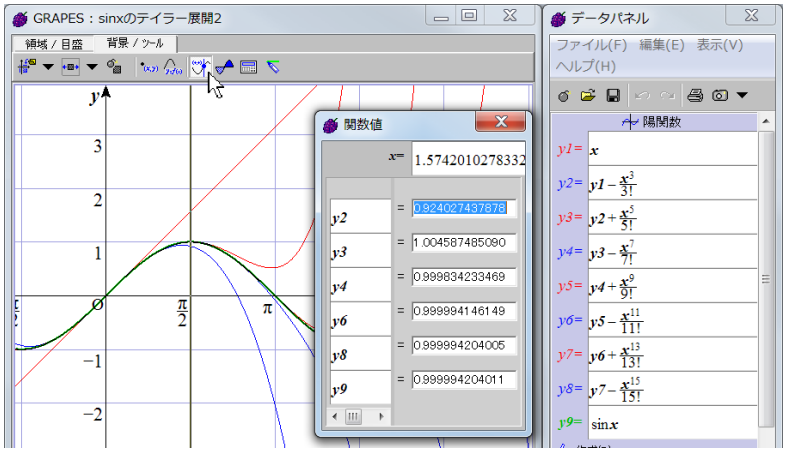
関数値ウィンドウでは，陽関数や任意の関数式について，x の値に対する関数値が表示され，複数の関数の値をグラフを見ながら比較することができます。

- **関数値ウィンドウを表示する**

［ツール］パレットの関数値ボタン をクリック

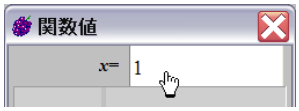


関数値ウィンドウが表示されます。また，グラフウィンドウには，x の値を示す垂直線が表示されます。



- **x の値を変える（電卓入力）**

1. x の値表示窓をクリック
2. 関数電卓で入力

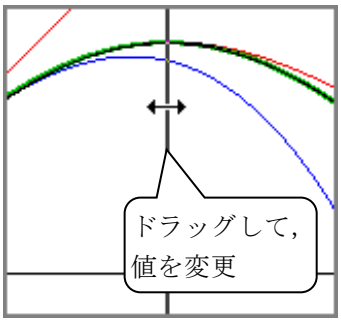


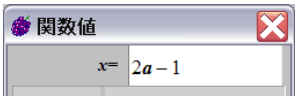
☆ x が数値で与えられているときは，下記のようにマウスのドラッグを用いて x の値を変えることができます。

- **x の値を変える（マウスでドラッグ）**

グラフウィンドウの垂直線をドラッグ

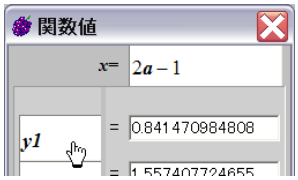
☆ 下図のように，x が式で与えられているときは，垂直線をドラッグで動かすことはできません。

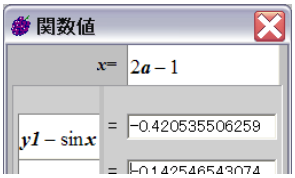




- **関数式を変更する**

1. 関数式表示部分をクリック
2. 関数電卓で式入力



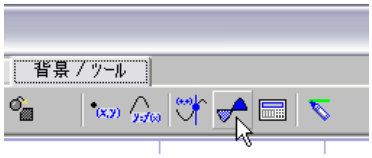


13-3 定積分ウィンドウ

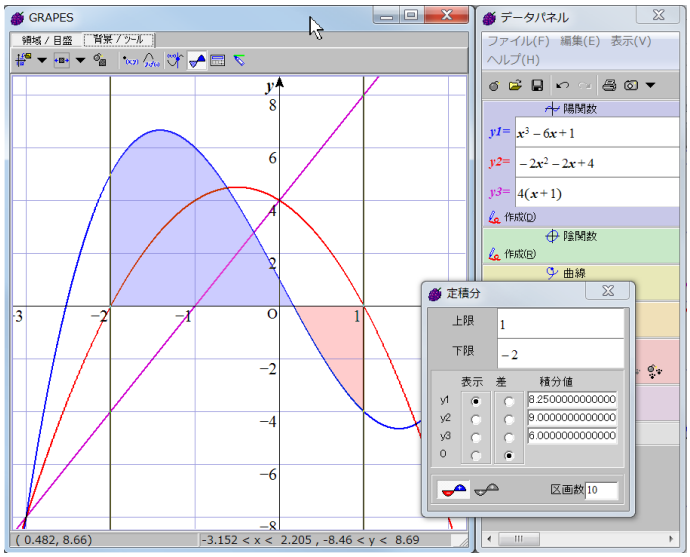
定積分ウィンドウには，陽関数について，定積分の値が表示されます。また，積分領域がグラフに表示されます。

- **関数値ウィンドウを表示する**

ツールパレットの定積分ボタン をクリック



定積分ウィンドウが表示されます。また，グラフウィンドウには，積分の範囲を示す 2 本の垂直線が表示されます。



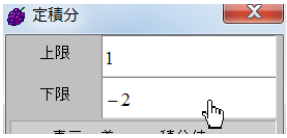
- **定積分ウィンドウ**

定積分ウィンドウには，定積分の範囲，「表示」ボタンと「差」ボタン，そして定積分の値が表示されています。また，下部には，積分/面積切り替えボタンと，区画数ボックスがあります。

積分領域は，正領域が青，負領域が赤で表示されます。

なお，定積分ウィンドウは，表示したままで，パラメータの増減など各種操作を行なうことができます。

- **下限や上限の値を変える（電卓入力）**



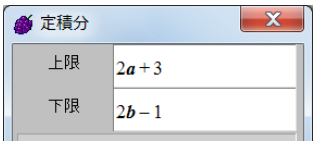
1. **下限や上限の値が表示されているボックスをクリック**
2. **関数電卓で入力**

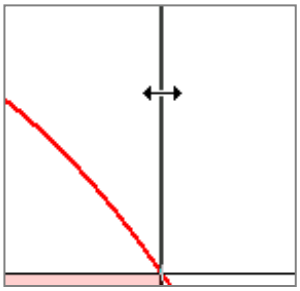
☆　下限や上限が数値で与えられているときは，下記のようにマウスのドラッグを用いて値を変えることができます。

- **下限や上限の値を変える（マウスでドラッグ）**

グラフウィンドウの垂直線をドラッグ

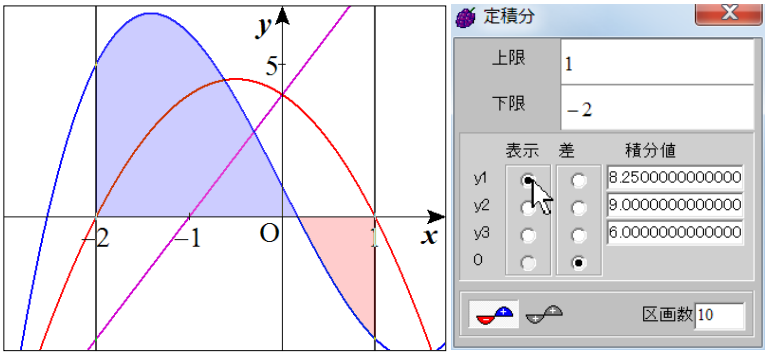
☆　下のように，下限や上限が式で与えられているときは，垂直線をドラッグで動かすことはできません。





- **積分領域の表示**

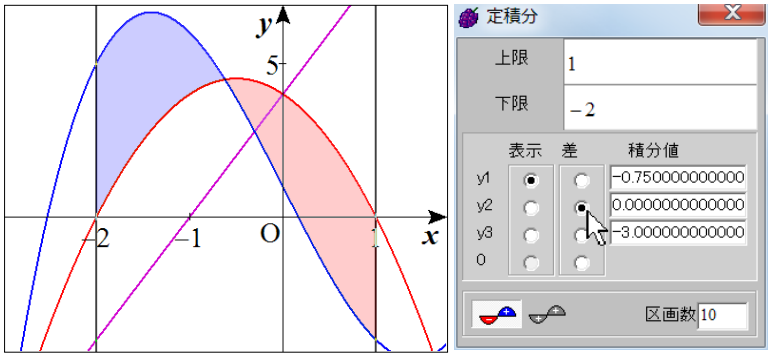
表示したい関数について，「表示」枠の中のボタンをクリック



- **差の表示**

差をとる関数について，「差」の枠で囲まれたボタンをクリック

すべての関数について，ここで選んだ関数との差の積分が表示されます。差を表示しないときには，最も下の「差」ボタンをチェックします。



- **面積と定積分**

積分ボタン のクリックで，定積分の値を求めます。

面積ボタン のクリックで，関数値に絶対値をつけて積分します。

- **近似計算の区画数**

定積分は特殊な 4 次近似式で計算しており，9 次以下の多項式では誤差を生じません。それ以外のときには，積分区間を細かく区切って計算します。区画数は 200 以下の自然数です。

13-4 関数電卓と式の計算，式の入力

- 関数電卓の表示

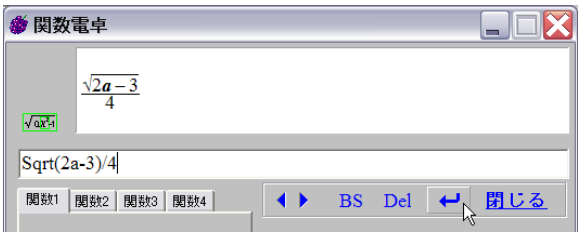
［ツール］パレットの関数電卓ボタン をクリック

- 式の値を求める

式の計算ボタン をクリック

これで，値が表示されます。

式はベクトルでも構いません。

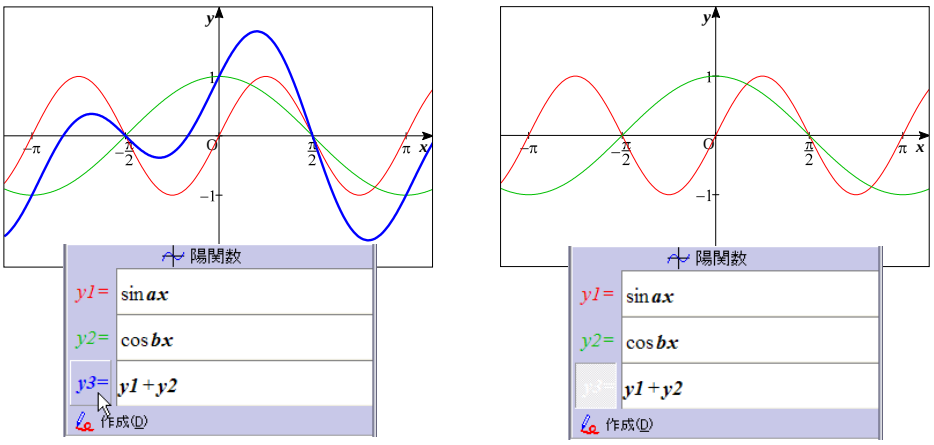


13-5 表示スイッチ

任意のグラフについて，データパネルのラベルをクリックするだけで，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クリック



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。また軌跡も表示されなくなります。

13-6 背景の利用

グラフ表示エリア背景に壁紙のように画像を貼り付けることができます。

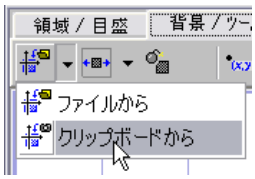
- **背景の読み込み**

読み込みボタン をクリック

画像ファイルを選びます。bmp, jpg, gif, png, emf, wmf に対応しています。

画像ファイルをグラフウィンドウにドロップしても背景にセットされます。

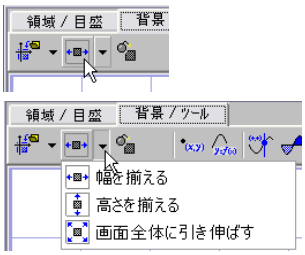
- **背景の貼り付け**



▼をクリックして，貼り付けボタン をクリック

クリップボードの画像が背景にセットされます。これを使うときは，あらかじめ画像をクリップボードにコピーしておく必要があります。

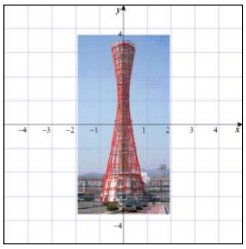
- **背景の引き伸ばし**

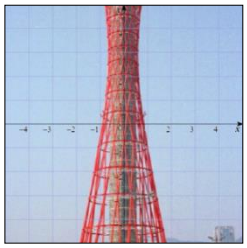


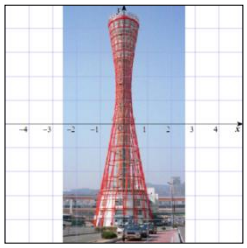
引き伸ばしボタン をクリック

もう一度クリックすると，元に戻ります。

▼をクリックすることで 3 種類の引き伸ばし方法を選ぶことができます。


そのまま


幅を揃える


高さを揃える

全体に引き伸ばす

- **背景の消去**

背景消去ボタン をクリック

- **背景の保存**

GRAPES のデータを保存する際，背景については，読み込んだファイル名だけが保存されます。したがって，GRAPES の保存ファイルと背景ファイルが同じフォルダにあるときに限って，読み出し時に背景がセットされます。なお，クリップボードから貼り付けた背景は保存されません。

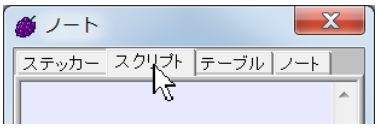
第 14 章　スクリプト

14-1 スクリプトとは

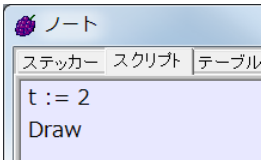
スクリプトとは，GRAPES 内で動く小さなプログラムです。

- スクリプトの特徴
  - 繰り返しやサブスクリプトの呼び出しを含んだ制御構造
  - パラメータのコントロール
  - 図形の座標への代入
  - テーブルセルへの代入
  - あらゆる式の計算
  - 描画，表示，残像のコントロール
  - 関数のグラフを波形に持つ音の出力
  - 自動実行条件の設定

- スクリプトの編集とスクリプトボタン



1. ノートエリアの［編集］をクリック
2. ［スクリプト］タグをクリック
3. スクリプトを記述する
4. ［OK］をクリック



ノートエリアに，スクリプト 1 行目の内容のボタンができます。
このボタンをスクリプトボタンといいます。

- スクリプト名の表示
  - 最初の 1 行が，スクリプトボタンとして表示される
  - 最初の行に注釈文（//）があるときは，注釈文の内容がボタン名になる
  - スクリプトの 1 行目に HideScript 命令があるとき，このスクリプトは表示されない

- スクリプトの書き方についての詳細
  - １行には１命令しか書くことができない
  - 語と語の間には，半角空白を入れる
  - 注釈は，ダブルスラッシュ“//”の後に記述する
  - スクリプトにだけ使える語（Draw など）は，大文字でも小文字でもかまわない
  - パラメータは小文字で書く。また，関数（sin, Sqrt, Pi など）はその通りに書く
  - 複数のスクリプトは空白行で区切る。（最大 20 個）

14-2 スクリプトの構成要素

- **変数**

  パラメータや基本図形の座標のほかに，ローカル変数として x , y , z , w があります。

- **代入**

  パラメータ，基本図形，テーブルセルに，式の値を代入することができます。

- **条件文と制御構造**

  ある条件を満たすときだけ実行する if ~ then ~ あるいは if ~ then ~ else ~，繰り返しを行なう for ~ next，ある条件を満たす間だけ繰り返す while ~ wend があります。

- **グラフ描画のコントロール**

  計算とグラフ描画を行なう Draw，OverDraw，座標の計算のみを行なう Calc，表示やステッカー表示の更新を行なう Refresh，RefreshSticker があります。

- **描画色のコントロール**

  点の色や関数のグラフ色を変更する SetColor，線の色を変更する SetLineColor，ペイント色を変更する SetInnerColor があります。

- **残像のコントロール**

  残像を消去する ClrAImg と ClrAImgData，残像化を行なう SetAImg，残像をコントロールする AImgOn, AImgOff があります。

- **表示のコントロール**

  グラフを表示する ShowObj，グラフを非表示にする HideObj，表示/非表示を切り替える InvShowObj があります。

- **表示範囲のコントロール**

  表示範囲を変更する SetArea ，グラフ表示範囲のピクセル数を変更する SetAreaSize，グラフ表示領域の中央の点の座標を変更する SetCenterSize があります。

- **テーブルのコントロール**

  テーブルデータの削除やソートなどが行えます。

- **音の出力**

  関数のグラフを波形の持つ音を出力する Play ，PlayAfter，PlaySync，サンプリング周波数を変更する SetSampleRate があります。

- **自動実行条件の設定**

  指定した関数が書き換えられたり，点やパラメータの値に変化があったりしたとき，スクリプトを自動実行する on ~ change があります。また，マウスの左ボタンが離されたときにスクリプトを自動実行する OnMouseUp があります。

- **実行のコントロール**

  時間待ちを行なう SetTme ～ Wait，停止して入力を待つ Pause，実行を中断する end があります。

- **別スクリプトの実行**

  Call で別のスクリプトを実行することができます。このとき，呼び出し先のスクリプトに対して，４つまでの引数を渡すことができます。

- **スクリプトボタンの非表示**

  スクリプトの先頭行に“HideScript”があるとき，スクリプトボタンは表示されません。

- **スクリプト編集のヒント**

  画面で右クリックで表示されるポップアップメニューを利用することで，スクリプトの全命令およびスクリプトでよく使用する関数を挿入することができます。

  （参照：15-1 ポップアップメニューの利用）

14-3 スクリプトの実行と停止

- スクリプトボタンによる実行

  スクリプトボタンのクリックでスクリプトが実行されます。

  ☆　文法ミスがある場合，ノートウィンドウを閉じるときにエラーメッセージが出されます。しかし，文法ミスに対しては，かなり不十分なチェックしか行なっていません。したがって，ミスがあるにもかかわらず，エラーメッセージが出ないこともあります。

  ☆　実行時エラーがある場合，エラー行は実行されず，そのまま次の行を実行します。

- On change 命令による実行

  スクリプトの 3 行目までに On Change 命令があるとき，指定した要素の変更で，スクリプトが実行されます。

  例：　On a, b Change：**パラメータ *a*, *b* のいずれかの変更で実行される。**

- onMouseUp 命令による実行

  スクリプトの 3 行目までに onMouseUp 命令があるとき，マウスの左ボタンが離されるとスクリプトが実行されます。

- スクリプトの終了

  スクリプト最終行の実行後，スクリプトは終了します。

- end 命令によるスクリプトの終了

  スクリプト途中行で end 命令があるとき，そのスクリプトは終了します。

- スクリプトの中断

  F1 キーを押し続ける

  実行に時間のかかるスクリプトを途中で停止させるときに用います。

14-4 変数と代入

- **グローバル変数**

  パラメータ，点の座標，テーブルセル，スクリプトの中で変数として使うことができます。

- **ローカル変数**

  4 つのローカル変数 x , y , z , w を使うことができます。これは，実行中のスクリプト内だけで有効です。例えば，スクリプト A の中の x とスクリプト B の中の x は，まったく無関係な値をとります。

  ローカル変数は，実数もしくはエラー値を代入することができます。ローカル変数で 2 次元ベクトル値を扱うときには，X , Y を用います。X は変数の組 (x , y) のことで，X = (x , y) です。同様に，Y = (z , w) です。

  例： X = (1 , 2)

  と

  x := 1

  y := 2

  は同じ結果を与えます。

  ローカル変数の初期値はエラー値で，実行後に値は破棄されます。また，実行中のスクリプトが他のスクリプトから CALL コマンドで呼び出されているときには，CALL 文の引数がこれら変数に代入されます。

  例： //Sum

  a := x+y

  は，C 言語風に書けば，次のようになります。

  void Sum(x , y , z , w);

  { a = x + y }

- **代入**

  パラメータ，点の座標，テーブルセル，およびローカル変数 x , y , z , w に対して，式の値を代入することができます。

  代入対象 := 式

  とします。代入対象と「:=」の間は空白を空けます。「:=」と式の間も同様です。

  パラメータや点（の座標成分）への代入の場合，式が値をもたないとき（$\frac{1}{0}$ など），代入文は実行されません。

- **パラメータ := 式**

  例：　a := sin t

- **点（の座標成分）:= 式**

  例：　P := (2 , 3) ，P.x := 2

  ☆　点への代入はそれ以外に比べて，速度的に劣ります。

  例えば,

  　　P := (2 , 3)

  を実行するより,

  P(s , t) において,

  　　s := 2

  　　t := 3

  を実行する方が，はるかに高速です。

- **P（点の番号）:= 式**

  点の番号とは，P, Q, R, …, V, A, B, …,N に対して，1, 2, 3, …, 7 , 8, 9, …, 21

  例：　P(2) := (2 , 3)　　（Q := (2 , 3) と結果は同じ）

  　　P(n) := (2 , 3)　　（この場合，*n* には 1 から 21 までの整数を与えます）

- **ローカル変数 := 式**

  例:　x := tan t

  ☆　ローカル変数は，実数値のほかにエラー値もとることができます。したがって，代入文の右辺が値を持たないときでも代入文は実行され，エラー値が代入されます。

- **Cells(行，列) := 式　（行や列は，定数以外に式を入れることもできます）**

  例：　Cells(1,2) := 4 / a

  ☆　右辺の値が 2 次元データの場合, 指定したセルとその右隣のセルに値が代入される。

  例：Cells(1,2) := (4,5) は，Cells(1,2) := 4 , Cells(1,3) := 5 と同じ結果になります。

☆　　列番号を省略した場合，1 列目のセルを表します。

- **CellsP(行，列) := 式　（行や列は，定数以外に式を入れることもできます）**

  右辺がベクトルデータのとき，指定したセルとその右隣のセルに代入されます。

  右辺がベクトルデータでないときには，代入は実行されません。

- Inc と Dec

  Inc（Dec）はパラメータやローカル変数の値を，指定した値だけ増加（減少）させます。

  Inc(パラメータ，増加幅)

  Dec(パラメータ，減少幅)

  増減幅が 1 の場合，“Inc(a)”のように増減幅を省略することができます。

- SetStepSize

  SetStepSize はデータパネルに表示されたパラメータの増減幅に，数値を代入します。

  SetStepSize(パラメータ，式)

  例：SetStepSize(t , 0.2)　　（パラメータ *t* の増減幅を 0.2 にセットします。）

  ☆　このコマンドで増減幅を変更した場合，データパネルで増減幅を変更した時と同様の結果を生じます。もし，そのパラメータを利用した曲線があり，「パラメータの同期」が ON になっている場合，その曲線も再描画されます。

14-5 制御構造と条件文

- for ～ next

  for～next 文は，for と next の間に挟まれた文を繰り返します。

  ```
  for パラメータやローカル変数 := 初期値 to 終値 step 刻み幅
      実行文（複数行でも可）
  next
  ```

  「step 刻み幅」を省略した場合，１ずつ増加します。

- while ～ wend

  while と wend の間に挟まれた文を，条件が満たされている間だけ繰り返します。

  ```
  while 条件式
      実行文（複数行でも可）
  wend
  ```

- if ～ then ～

  条件が満たされれば，命令を実行します。

  ```
  if 条件文 then 実行文
  ```

  命令が複数行あるときには，次のようにします。

  ```
  if 条件文 then
      実行文（複数行でも可）
  endif
  ```

- if ～ then ～ else ～

  条件が満たされたときと，満たされないときで，異なる命令を実行します。

  ```
  if 条件文 then 実行文1 else 実行文2
  ```

  あるいは

  ```
  if 条件文 then
      実行文（複数行でも可）
  else
      実行文（複数行でも可）
  endif
  ```

14-6 描画と計算

- Draw

  現在のパラメータ値に従って座標を計算し，グラフを描きます。

  軌跡を残さないときには，コロンをつけた Draw:を用います。

ステッカーを除く画面上のすべての表示が更新されます。

☆スクリプトでは変数に値を代入しても，それに伴うグラフの再計算は行なわれません。再計算するには Draw や Calc を実行します。

- OverDraw

  Draw とほとんど同じですが，画面の消去を行わず，ただ描き加えていきます。Draw よりも高速です。

- Calc

  現在のパラメータ値に従って座標を計算し軌跡や残像を内部メモリーに残します。ただし，画面の更新は行いません。画面の更新には，Draw もしくは後述する Refresh を使います。

軌跡を残さないときには，コロンをつけた Calc:を用います。

グラフの描画を伴わないので Draw や OverDraw よりも高速です。

- Refresh

  Calc は，残像や軌跡の計算結果を内部メモリーに残しますが，画面の更新は行いません。Refresh は Calc によって残された残像や軌跡をグラフウィンドウに表示します。

  Refresh は残像や軌跡の表示を行いますが，現在のグラフを表示することはできません。現在のグラフを表示するには，Draw を使います。Draw は残像の表示も行います。

- RefreshSticker

  ステッカーの表示を更新します。

  ステッカーに式の値が表示されている場合，最新の結果に更新されます。

14-7 グラフの色の変更

- SetColor(図形名，色)

  図形の表示色を変更します。基本図形や曲線では点の色，関数のグラフではグラフ色を，連結図形では線の色を変更します。

  - 図形は，一度にはひとつしか指定できません。
  - 色は，RGBを表す 24 ビット整数で与えますが，GRAPESは 16 進数を扱うことはできないので，rgb関数や，hsl関数を利用して色を指定します。

  例： SetColor(y1 , rgb(0.8, 0, 1))

  関数 y1 のグラフが紫色になります。

- SetLineColor(図形名，色)

  図形の表示色を変更します。基本図形や曲線では線の色，関数のグラフではグラフ色を，連結図形では線の色を変更します。

- SetInsideColor(図形名，色)

  基本図形，曲線，連結図形のペイント色を変更します。

14-8 残像消去と残像化

- ClrAImg(図形名)，ClrAImg

  図形の残像や軌跡を消去します。

  （“ClrAImg”は“Clear After Images”を省略したものです）

  - ClrAImg(図形名）とすれば，その図形の残像だけを消去します。
  - 図形名には，陽関数，陰関数，曲線，基本図形のほか連結図形を指定することができます。また，指定する図形が複数ある場合には，カンマで区切って記述します。

  例：ClrAImg(y1,C2,P,L2）：陽関数 y1，陰関数 C1，図形 P，連結図形 L2 の残像を消去します。

  - すべての図形の残像を消去するには，図形名の指定なしに，ClrAImg とします。

- ClrAImgData，あるいは ClrAImgData(図形)

  □ 残像のデータ消去するが，画面の更新は行わない。

  画面更新は，Draw 命令や Refresh 命令実行時に行われる。

- SetAImg

  現在描かれているすべてのグラフを残像として残します。

14-9 表示コントロール

- AImgOn と AImgOff

  AImgOff 命令によって，パラメータエリア上の［残像 OFF］ボタン が押し下げられ，図形の設定にかかわらず残像は記録されなくなります。AImgOn 命令によって元の状態に戻ります。

- ShowObj(図形名)，HideObj(図形名)，ShowToggleObj(図形名)

  - HideObj(図形名)

    図形の表示スイッチが押し下げられ，グラフが表示されなくなります。

    図形名には，陽関数，陰関数，曲線，基本図形のほか連結図形を指定することができます。また，指定する図形が複数ある場合には，カンマで区切って記述します。

    例：HideObj(y1,C2,P,L2)

    陽関数 y1，陰関数 C1，図形 P，連結図形 L2 を表示しないようにします。

  - ShowObj(図形名)

    図形の表示スイッチの押下げ状態は解除され，図形は表示されるようになります。

  - ShowToggleObj(図形名)

    図形の表示，非表示の状態はトグルで変化します。

- ShowAxes，HideAxes，ShowToggleAxes

  座標軸の表示，非表示，トグルで切り替え

- ShowMesh，HideMesh，ShowToggleMesh

  メッシュの表示，非表示，トグルで切り替え

- SetArea(x1,y1,x2,y2)

  2 点（x1,y1），(x2,y2）を対角線とする領域を，画面全体に表示します。この命令によって，すべてのグラフは再描画されます。

- SetAreasize(x , y)

  グラフ表示エリアの大きさ（ピクセル数）を 幅×高さ=x×y にします。この命令によって，すべてのグラフは再描画されます。

- SetCenterPos(x, y)　，SetCenterPos(P)

  指定した座標位置が画面中央にくるように，画面全体が平行移動します。

- Zoom(x)

  指定した値だけ，グラフを拡大する。拡大の中心は，グラフ画面中央。

14-10 テーブルのコントロール

- セルへの代入

  「14-4 変数と代入」を参照してください。

- ClrAllCells

  テーブルの全セルの値を消去します。

- ClrColumn(列番号)

  列のデータを削除します。

  例：ClrColumn(2)

  第２列のデータをすべて削除します。

- TrimColumn(列番号)

  列の無効データを削除し，セルを上に詰めます。

- TrimColumns (列番号１，列番号２)

  ２つの列のデータをセットとみて，一方でも無効データが含まれているときは，同じ行の他方のデータも削除し，上にセルを詰める。

列１	列２
1	2
3	
	4
5	6

⇒

列１	列２
1	2
5	6

- SortColumn(列番号)

  列のデータを昇順にソートする。

- SortDesColumn(列番号)

  列のデータを降順にソートする。

14-11 音の出力

- Play(パラメータ，開始秒，終了秒，波形関数)
  - 量子化 16 ビット，サンプリングレート 22,050Hz（変更可能）の音を出す

    例 ： Play(t , 0 , 2 , sin880Pit) （補足：左記の "Pit" はπと t の積です）

    振動数 440Hz の正弦波を 2 秒間鳴らします。
  - ステレオ出力時にはベクトルデータを与える

    例 ： Play(t , 0 , 2 , (sin 880Pit , sin 882Pit))

    左チャンネルから 440Hz，右チャンネルから 441Hz の音を出します。

  パラメータは，GRAPES で使えるすべてのパラメータと変数 x が使用できます。

  演奏が開始されるのは，すべてのデータの計算終了後です。また，波形関数には陽関数を利用することもできます。

- Play

  引数なしの Play コマンドは，作業用メモリーに残されたデータを演奏します。
  - 下記 PlayAfter コマンドで作成した Wave データを演奏する
  - 最後に演奏した Wave データを再演奏する

- PlayAfter(パラメータ，開始秒，終了秒，波形関数)
  - PlayAfter コマンドで作成された Wave データは, 作業用メモリー上に配置されたまま，演奏されません。
  - 作業用メモリー上に Wave データが残っている状態で次の PlayAfter コマンドが実行された場合，これら Wave データは結合されます。
  - 作業用メモリー上に作成された Wave データを演奏するには，Play コマンドを実行します。
  - 複数の Play()命令を連続して実行した場合，高性能なパソコンでは演奏順序が乱れることがあります。このような場合には，PlayAfter コマンドでひと繋がりのデータを作成してから，最後に Play コマンドで演奏するようにします。

  例 ： PlayAfter(t , 0 , 1 , sin400Pit)

  PlayAfter(t , 0 , 2 , sin600Pit)

  Play

  200Hz の正弦波を 1 秒間，300Hz の正弦波を 2 秒間，出力します。

- PlaySync(パラメータ，開始秒，終了秒，波形関数)

  Play コマンドと異なるのは次の 1 点だけです。

  - Play コマンドでの演奏はバックグラウンドで行われるのに対して，PlaySync コマンドでは演奏を終えるまで次の命令を実行できません。

- SetSampleRate(サンプリングレート)

  サンプリングレートを変更します。

  例 ： 48,000Hz に設定するには

  SetSampleRate(48,000)

  変更した結果は GRAPES が終了されるまで保たれます。

  - ひとつの wave データで演奏できる最大時間は，480,000÷サンプリングレート（秒）です。
  - 量子化ビット数は 16 ビット固定です。GRAPES では –10 ～ 10 の間を –32768 ～ 32767 に分割しています。したがって，波形関数の値がこの範囲を超えると，音は歪みます。

- 音の保存

  最後に出した音を，wave ファイルとして保存することができます。

  ［ファイル］メニューの［音の保存］をクリック

14-12 実行のコントロール

- SetTime(ミリ秒) ～ Wait

SetTime はタイマーをセットします。セットした時間が経過するまで Wait の位置で待ちます。セットした時間を過ぎているときは何もせずに次の命令に移ります。単位はミリ秒です。

例：SetTime(n)

```
    a := 3
    Draw
  Wait          //SetTime を実行してから n ミリ秒経過するまで待つ
```

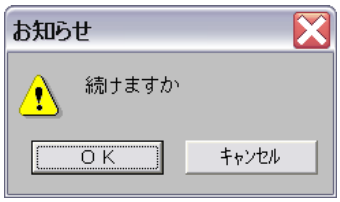
- Wait(ミリ秒)

指定した時間だけ停止します。単位はミリ秒です。

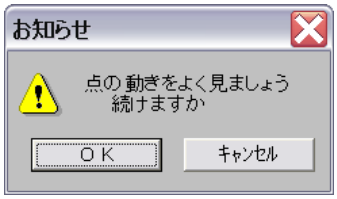
- Pause，Pause(メッセージ)

スクリプトの実行を停止し，入力を待ちます。

Pause(メッセージ)とした場合，メッセージを表示して停止します。



pause



pause(点の動きをよくみましょう)

［OK］をクリックすると実行を続けますが，［キャンセル］では実行を中断します。

- Call(サブスクリプト名 ，引数１，引数２，引数３，引数４)

同じプロジェクト内の別のスクリプトを実行します。実行後は次の行に制御が移ります。

Call コマンドの 4 つの引数は，その値がサブスクリプトのローカル変数 x , y , z , w に代入され，それらの初期値となります。これら引数はすべて，あるいは途中から後ろを，省略することができます。省略した引数にはエラー値が代入されます。

引数は実数あるいはエラー値でなければなりません。2 次元ベクトル値を引数に与えた場合，2 つの実数値を与えたものとみなされます。例えば，

Call(test , (1,2),(3,4))，Call(test , (1,2),3,4) ，Call(test , 1,2,(3,4))，Call(test , 1,2,3,4) はすべて同じ結果を与えます。

ただし，Call(test , 1,(2,3),4) のような与え方は許されません。

- **_サブスクリプト名 (引数１, 引数２, 引数３, 引数４)**

「Call(サブスクリプト名 , 引数１, 引数２, 引数３, 引数４)」と同じです。

例：_test(1,2,3,4) と Call(test,1,2,3,4) は同じ動作をします。

- End

実行中のスクリプトを終了します。もし，他のスクリプトから Call 命令によって呼ばれているときには，元のスクリプトに制御が戻ります。

- on 要素名 change

指定した要素が変更されるとスクリプトが自動実行されます。

- 指定できる要素は，パラメータ，基本図形や曲線（P, Q など），定義関数，陽関数，陰関数です。連結図形を指定することはできません。複数の要素があるときは，カンマで区切って書きます。
- この命令は必ずスクリプトの３行目までに書きます。

例：on a,b,f change

（パラメータ a,b あるいは関数 f のいずれかが変更されると実行される）

次の例では，パラメータ a の値を変更すると，パラメータ *t* を初期化して，残像や軌跡を消去します。

```
//初期化
on a change
t := 0
calc
ClrAImg
```

- onMouseUp

グラフ画面上でマウスの左ボタンが離されるとスクリプトが自動実行されます。

- この命令は必ずスクリプトの３行目までに書きます。
- 実行時のマウスの位置を取得するには，Pos 関数を利用します。

例：OnMouseUp

次の例では，画面をマウスでクリックした際，点 A からの座標距離が 1 以内のとき，点 A を青く，それ以外のときは赤くします。

```
OnMouseUp
if [A - mousePos] < 1 then SetColor(A,rgb(0,0,1)) else SetColor(A,rgb(1,0,0))
draw
```

- **残像の消去とスクリプト**

  残像の消去は，パラメータの値を初期化してから行います。これは，残像というのは，新しいグラフの描画が行われる直前に記録されるものであり，最初に残像を消去しても，次の描画時の直前に作られる今のグラフの残像までは消すことはできないからです。

- **実行のタイミング**

  on ～ change 文による自動実行は, データパネルの操作によって対象要素が直接変更されたときだけ行われます。データパネルの操作によって要素が変更された場合，座標が再計算されグラフが描画されますが，自動実行のスクリプトは，データパネルの変更の直後に実行され，その後で座標の再計算と描画が行われます。

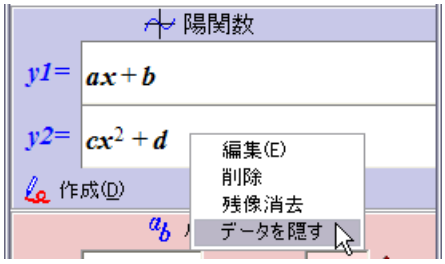
第 15 章　TIPS

15-1 ポップアップメニューの利用

- **データパネルのポップアップメニュー**

データパネルの関数式表示部分で右クリック

ポップアップメニューが表示され，その関数について，訂正，削除，残像消去を行なうことができます。



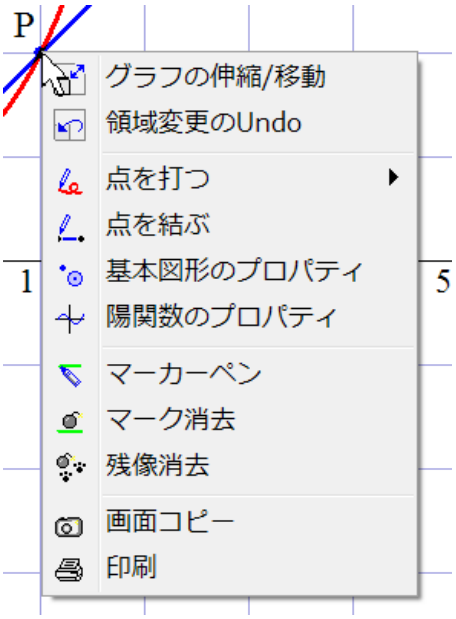
- **グラフ表示エリアのポップアップメニュー**

グラフ表示エリアで右クリック（Shift +クリック）

ポップアップメニューが表示され，グラフに関するさまざまな操作を行なうことができます。

- **領域モードにする**
- **領域変更の Undo，Redo（取り消しとやり直し）**
- **点を打つ**
- **点を結ぶ**
- **グラフのプロパティ表示**

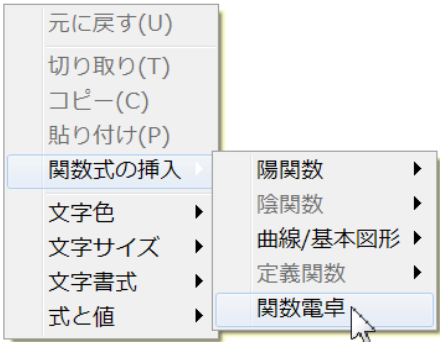
  クリックした位置にあるグラフや図形のプロパティを表示し編集することができます。
- **マーカーペンの表示と消去**
- **残像消去**
- **画面コピー，画面印刷**



- **ステッカー編集画面のポップアップメニュー**

ステッカー編集画面で右クリック

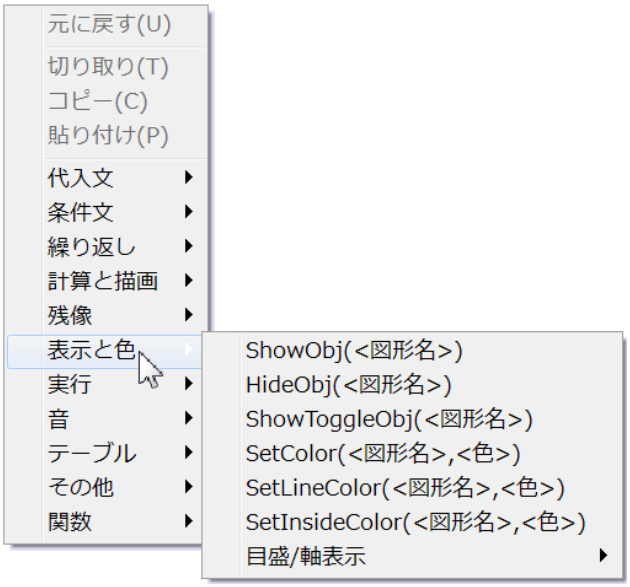
ポップアップメニューが表示され，必要な書式や関数式の挿入を行うことができます。



- **スクリプト編集画面のポップアップメニュー**

スクリプト編集画面で右クリック

ポップアップメニューが表示され, スクリプトの全コマンドおよびスクリプトでよく使用する関数を挿入することができます。



15-2 画面の印刷と貼り付け

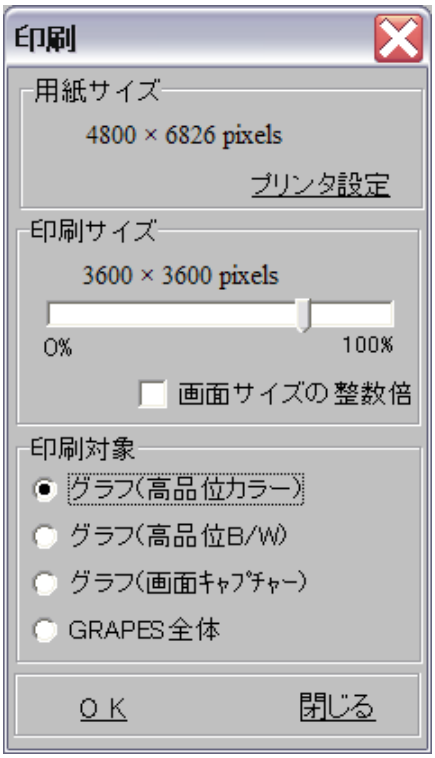
- 画面印刷

1. ツールバーの画面印刷ボタン[🖶]をクリック

   画面印刷の設定ダイアログが表示されます。

2. 画面サイズを調整する。

   プリンタの設定を変えるときは，「用紙サイズ」の［プリンタ設定］をクリックします。



3. 印刷対象の選択

   ［グラフ(高品位カラー)］や［グラフ(高品位 B/W)］を選ぶと，ベクトルデータを用いた滑らかなグラフを印刷することができます。

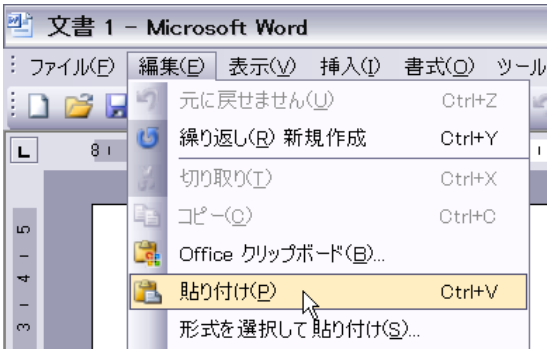
   ［グラフ（画面キャプチャー)］を選ぶとグラフを画面そのままに印刷します。また，［GRAPES 全体］を選ぶと，グラフウィンドウとデータパネルの両方を印刷します。このとき［印刷サイズ］の［画面サイズの整数倍］チェックボックスをチェックすることで，きれいな印刷結果を得ることができます。

4. ［OK］をクリック

- 画面コピーとワープロへの貼り付け

1. データパネルの画面コピーボタン[📷]をクリック

   グラフ表示エリア全体がクリップボードにコピーされます。データ形式やコピー対象はコピーボタン右横の[▼]をクリックして選びます。



2. 貼り付け先のワープロで，［編集］メニューの［貼り付け］をクリック

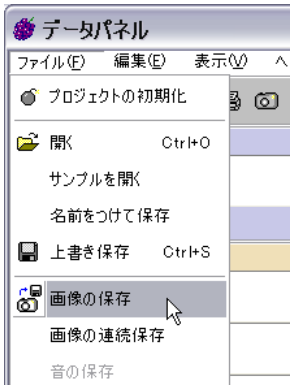
   これで，クリップボードの画像データが，ワープロの文書に貼り付けられます

   ここから後は，ワープロのマニュアルをご覧ください。

☆　画面コピーに際して，貼り付け先のソフトによっては，［グラフ（高品位カラー)］や［グラフ（高品位 B/W)］を選んでいるにも拘らず，十分な解像度が得られないことがあります。そのときは，Shift キーを押しながら，コピーキーをクリックしてください。

15-3 画像や音の保存

- **グラフ画面の保存**



［ファイル］メニューの［画像の保存］をクリック

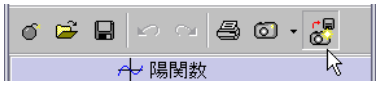
保存形式は，Windows Bitmap（*.bmp），Jpeg（*.jpg），Gif (*.gif)，Png（*.png)，Enhanced Metafile（*.emf)，Windows Metafile（*.wmf）に対応しています。このうち，Enhanced Metafile と Windows Metafile では，ぎざぎざのない高品質なグラフを保存することができます。

- **グラフ画面の連続保存**

1. ［ファイル］メニューの［画像の連続保存］をクリック

   ツールバーに［連続保存］アイコン が表示され，以後，画面の更新があるごとに画像が保存されていきます。

2. ［連続保存］アイコン をクリックして終了



下記は，ファイル名を“Image”として連続保存したときの様子です。パラメータを動かすごとに画像が保存されていきます。画像の下の文字列は各画像の保存ファイル名です。

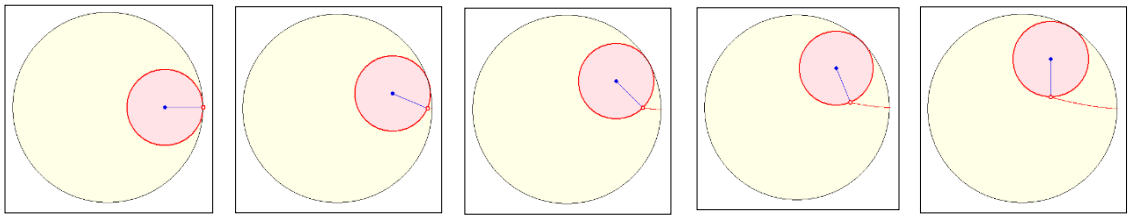
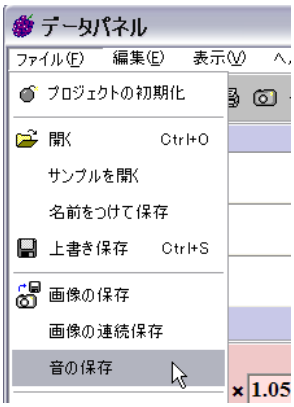


Image000　Image001　Image002　Image003　Image004

- **音の保存**



スクリプトの Play コマンドで最後に演奏した音を保存します。

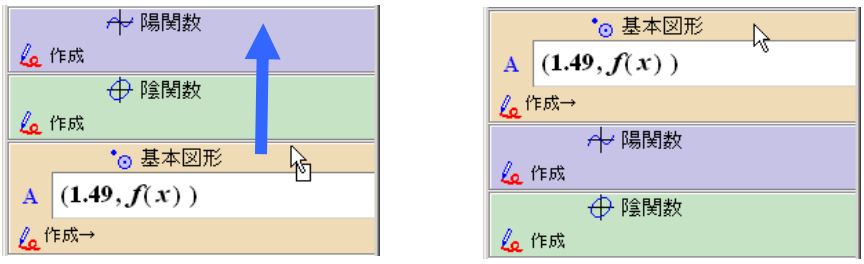
［ファイル］メニューの［音の保存］をクリック

15-4 データパネルのカスタマイズ

データパネルの各エリアや個々のデータは，表示順序を入れ替えたり，表示しないようにしたりすることができます。

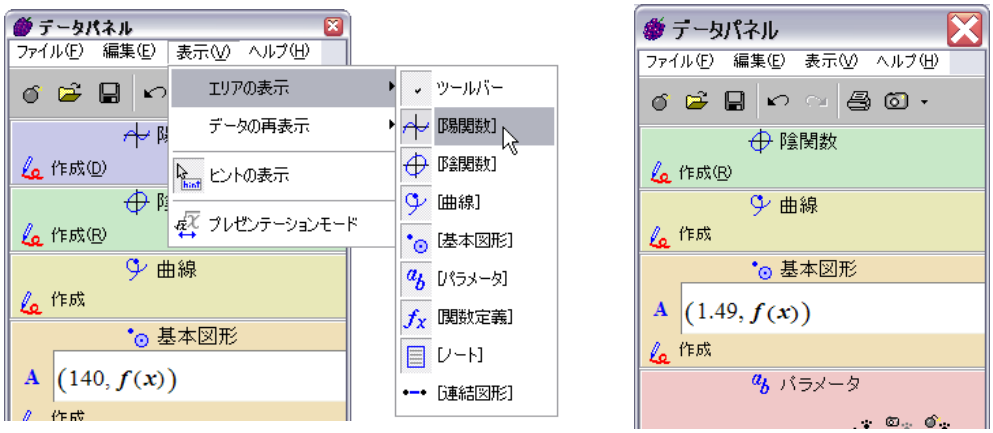
- 各エリアの移動

エリアのタイトル部分を他のエリアにドラッグ



- 各エリアを隠す

［表示］メニューの［エリアの表示］を選び，表示/非表示を切り替えるエリアをクリック

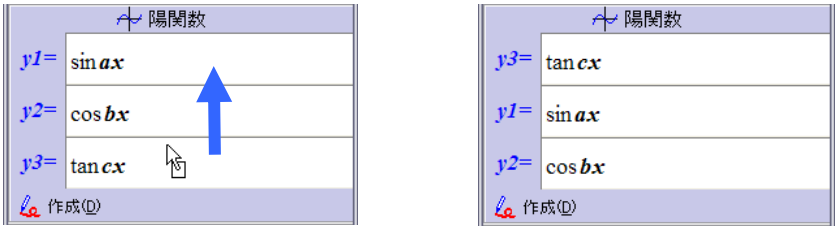


- 個々の式やデータの移動

関数式の式表示窓やパラメータの値表示窓を上下にドラッグ

同じエリア内でデータの表示順序を変更することができます。

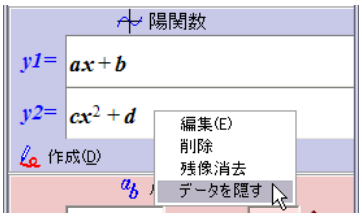
連結図形やステッカーの表示順序を変更することはできません。



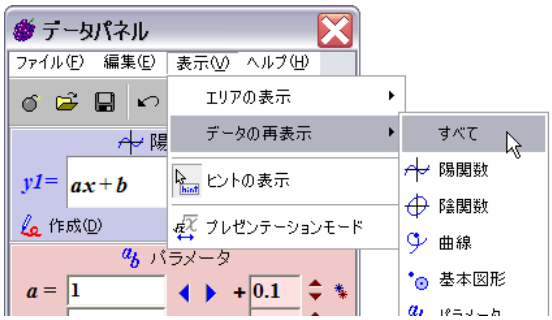
- 個々の式やデータを隠す

  隠したいデータの上で右クリックし，［データを隠す］を選ぶ

  関数や図形，パラメータが対象です。



  隠したデータを再表示するには，［表示］メニューの［データの再表示］を選ぶ

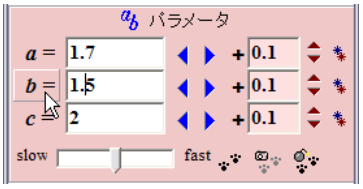


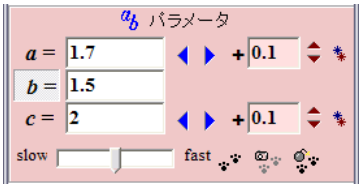
- パラメータの値を固定する

  パラメータ文字をクリック

  ラベルが押し下げられ，パラメータの値を変更できなくなります。

  もう一度クリックすると，値の固定が解除されます。





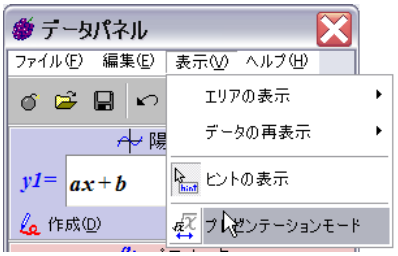
15-5 プレゼンテーションに

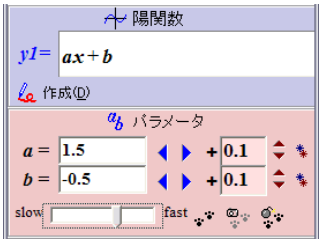
- **プレゼンテーションモード**

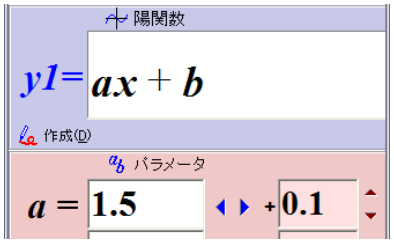
式イメージやパラメータの値を 2 倍の大きさのフォントでを表示します。

［表示］メニューの［プレゼンテーションモード］をクリック

再度クリックすると，元に戻ります。





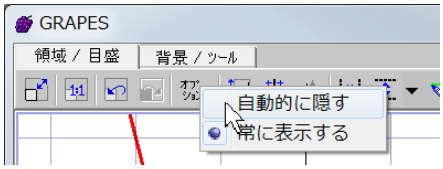


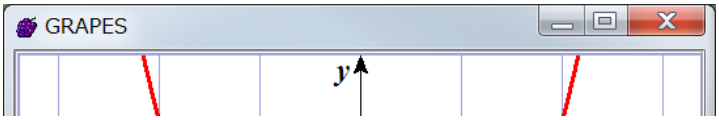
［通常モード］　　［プレゼンテーションモード］

- **グラフ表示エリアの拡大**

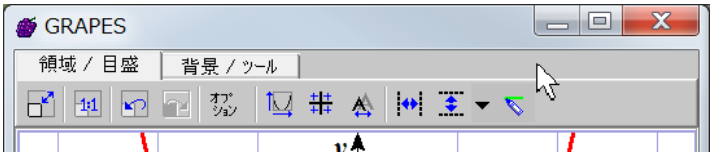
コントロールパレットを隠して，グラフ表示エリアを広げます。

コントロールパレットで右クリックし，［自動的に隠す］をクリック





グラフ表示エリア最上部にマウスを移動すると，コントロールパレットは表示されます。



- **マーカーを使う**

1. **［目盛］パレットのマーカーボタン をクリック**

   または，Alt + M を押す。

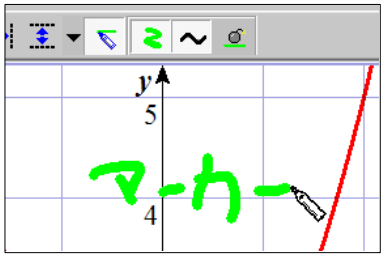
   マーカーパレットが表示され，カーソルがマーカーの形になります。

2. **左ドラッグで描く**

   マウスの左ボタンを押している間，描き続けます。

3. **右ドラッグ（Shift + ドラッグ）でマークを消す**

4. **マークの全消去ボタン をクリックして，すべてのマー**



クを消す

15-6 windows8 への対応について

- **高 DPI 対応について**

96DPI を基準として， DPI 値に応じて，コントロール類やグラフの表示サイズが変更されます。ただし，ボタンのアイコンは拡大されません。

例：120DPI の場合，120/96 = 125% の大きさに拡大されます。

- **タッチタブレットでの使い方**

グラフウィンドウでの操作

- **領域移動は右ドラッグ（つまりロングタップのあとドラッグ）**
- **拡大縮小はピンチによって，画面中央を中心に拡大縮小**
- **任意の位置を画面中央に移動するには，その位置をダブルタップ**
- **縦方向や横方向にだけ拡大縮小するには，伸縮／移動モードで操作**

任意の位置から横もしくは縦方向に（１本指で）ドラッグで拡大

上記作業の途中で反対側に大きく動かす（U ターンする）と縮小

データパネルでの操作

- **データパネル上のステッカーを移動するには，横方向にドラッグ**

（グラフウィンドウのステッカーはどのようにでもドラッグできる）

- **データパネル内の各エリアのパネルや各関数（や点など）の表示位置を変更するには，まず横方向にドラッグしてから上下に移動**
- **最初に縦方向にドラッグすると，スクロールになる**

横方向にスクロールするには，まず縦方向にドラッグしてから横に動かす。

対応できていない操作

- **パラメータの連続増減（パラメータ増減ボタンをマウスで押しっぱなしにする操作です）**

増減ボタンをタッチのあと凹んだ状態でもう一度タッチすればよいのだが・・・

タイミングが難しい。

- **点のドラッグも気難しい**

点を指で押さえたあと素早くドラッグすればよいが，もたもたしていると右ドラッグ（領域移動）になってしまう。