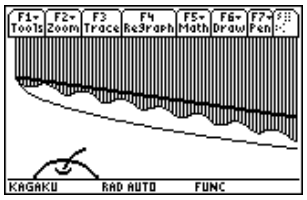
第11章　グラフアートの作成

数ナビのグラフ機能を利用すると, いろいろな関数のグラフを繋げて絵 (アート) を作成することができます。それを作成する過程は, 関数グラフの総復習になるばかりではなく, 今まで気がつかなかったいろいろな性質への気付きも多数得られます。作成に多少の苦労はあっても, それを上回る大きな達成感が得られ, まさに楽しみながら数学の理解が深められる題材になっています。

この章では, その作り方について解説します。

11.1　関数 $y = f(x)$ によるグラフアート

最初に, 数ナビを使って作成したグラフアートを紹介しておきましょう。直線や放物線などを組み合わせるだけで, 右側のようなグラフが作成可能です。この作品は, 平成 14 年度の物質化学工学科 1 年生が冬期休業の課題として作成したものです。「リンゴ切り」というタイトルがつけられています。描画範囲は, [F2] 4 で指定される範囲 ($-7.9 \leqq x \leqq 7.9,\ -3.8 \leqq y \leqq 3.8$) です。



11.1.1　関数 $y = f(x)$ を利用したグラフアートの作り方 (1)

作成の流れ

グラフアートを作成するには, まず, 自分の描きたい内容を紙に下書きしてみることが必要です。作成するものは, 動植物, 風景, 物, イラスト, 抽象画など何でもかまいません。ただし, 「アート」としての独自性の観点から, 何かのキャラクターやロゴマークのようなものは避けてください。

作成しようとする絵のイメージができたら, その輪郭を表す曲線はどのような関数のグラフで実現できるかを考える必要があります。数ナビで一度に指定できる関数の数は 99 個です。個々のグラフには, その描画範囲 (後述) や太線, 点線等の描画の仕方 (後述) を仕方できます。どのような関数を利用すればよいか分からないときは, いろいろなグラフを表示させて個々のグラフの特徴を把握することが必要と思います。

そして, 一つ一つの曲線について, 関数の式と描画範囲を根気強く指定していく作業が必要となります。かなりの試行錯誤を伴いながら作成していくことになるでしょう。

指定する関数の数が増えてくると, 全体を描画するには時間がかかるようになります。そのようなときは, 調整に必要なグラフだけを表示させるとよいでしょう。 [♦] [F1] の関数定義画面で, すでに調整済みの関数の $\surd$ 印は [F4] を利用して外しておくと描画が早くなります。

ある程度, グラフが確定したら, 必要がなければ座標軸を消してファイルを保存してください (後述)。

描画範囲の指定 [|]

個々の関数には, そのグラフの描画範囲を指定することができます。$y1 = x^2$ の $-1 < x < 2$ の範囲だけを描画させたいのであれば, 「$y1(x) = x^2 \mid -1 < x < 2$」とします。つまり, 関数の後に縦線 [|] を引き, その後に描画する範囲を指定するだけです。 [|] は [=] の下のキーです。不等号 $<$ $>$ は, [2nd] 0, [2nd] . です。

関数定義の際の留意事項

個々の関数には描画範囲を指定できますが, 「$y = 2 \mid -1 < x < 2$」のように x 軸に平行な直線を定義すると, 数ナビは指定した範囲ではなく画面一杯のグラフを描画してきます。この場合は, 直線の式に変数 x を含む形で定義してください。例えば, 「$y = 2 + 0 \cdot x \mid -1 < x < 2$」として定義してください。

一方, $x = 1$ のような y 軸に平行な直線は, $y = \cdots$ の形では定義することはできません。それは傾きの非常に大きな直線として定義すれば実現できます。例えば, 点 $(1, 0)$ を通る縦の直線は $y = 100(x-1)$ などとすれば表示されますが, この定義ではグラフの一部を描画させることは難しいです。関数の式の工夫次第では描画可能かもしれないので, 各自工夫してみてください。なお, 次の節で述べる媒介変数を利用すると, このような縦線の一部も簡単に描画させることができます。

絶対値関数 (absolute function)$y = |x|$ は, 「$y = \mathrm{abs}(x)$」として定義できます。abs は [alpha] を利用してください。または, [CATALOG] を利用してもかまいません。

いろいろな関数に絶対値記号をつけたグラフを表示させてみてください。簡単な式で複雑なグラフが表示される場合があります。

中心 (a, b), 半径 r の円の式は, 1 年の後期に学ぶように $(x-a)^2 + (y-b)^2 = r^2$ という方程式で表されます。これを y について解くと $y = \pm\sqrt{r^2 - (x-a)^2} + b$ となるので, 円の上半分は $y = \sqrt{r^2 - (x-a)^2} + b$ となります。ただし, 半円の端点部分は綺麗には描画されません。次節で説明する媒介変数表示を利用すると綺麗な円が描画可能です。

グラフの保存

作成した図を保存するには, [F1] 2(Save Copy As $\cdots$) とし, 「Type」の箇所を「GDB」とし, [▼] で「Variable」の箇所にいって適当な名前を指定してください。「GDB」形式のファイルで保存すると, 定義した関数の式や, 個々の関数に指定した描画方法などの全てが保存されます。

「Type」を「Picture」にして保存すると, 描画した絵だけが画像ファイルとして保存されます。関数の式は保存されません。

保存した画像を呼び出すには, [F1] 1(Open) として Variable に自分が保存した名前を入れて [ENTER] を押します。ただし, 「GDB」形式のファイルをオープンすると, 現在 [♦] [F1] の画面で定義されている関数はすべて削除され, 新にオープンする画像の定義関数で置き換られてしまうので注意してください。それに対して「PIC」形式のファイルをオープンすると, 現在表示されている画像に上に, 新たにオープンしたファイルの画像が重ね書きされます。この機能を利用すると, 版画の多色塗りの作業のように, いろいろな絵を重ね合わせて一つの絵を作成することができます。

11.1.2　関数 $y = f(x)$ を利用したグラフアートの作り方 (2)

前節までで, 作成しようとしている絵の輪郭ができあがるはずです。その輪郭を構成している個々の関数のグラフについて, それを太線や点線で表示させたりすることができます。以下では, その方法について説明します。

グラフの描画スタイルの設定

数ナビは, 関数のグラフを描画するとき, x の値を少しずつ変えながら対応する y の値を計算し, 隣り合う 2 点を線分 (Line) で結んでいくことで曲線を表示していきます。その 2 点を点 (Dot) の連続で表示させたり, 太い (Thick) 線で表示させたり, カーソルが動く (Animate) だけで曲線が残らないようにしたり, 曲線の上を斜線で塗りつぶしたりすることができます。

(1) 関数の定義画面 (♦ F1) に関数が定義されているとします。
(2) ▲, ▼ で表示方法を指定したい関数に合わせ, 黒く反転した関数の表示方法を変更できます。
(3) F6 (2nd F1) を押すとグラフの描画スタイルが指定できます。
(4) 特に変更がなければ, 「1:Line」にチェック印がついています。
(5) 線 (Line) ではなく点 (Dot) の連続で表示させたいときは「2：Dot」, 点 • の代わりに四角■を使いたいときは「3:Square」, 太線 (Thick) で表示させたいときは「4：Thick」に, カーソルが動くだけで曲線が表示されないようにするには「5：Animate」に合わせて ENTER を押すことで変更ができます。「7:Above」は曲線の上側に斜線を引き, 「8:Below」は下側に斜線を引く。
(6) 再度 2nd F1 を押せば, チェック印の位置が変更されているはずです。
(7) 以上のことを, 個々の関数ごとに指定してください。
(8) 必要な変更が終わったら, ♦ F3 を押すことで指定された表示方法でグラフが描画されます。

グラフ画面の描画形式の設定

関数のグラフは, 定義した関数の順番に描画されますが, 複数のグラフを同時に描画させることもできます。また, 座標軸を消して描画させることもできます。

ここでは, その方法について説明します。

(1) ♦ F1, ♦ F2, ♦ F3 のいずれかの画面において, ♦ | (または F1 9) を押します。
(2) 「Coordinate」の RECT は, 変更しないでください。これは, 直交座標 (rectangular coordinates) を利用することを示しています。
(3) 「Graph Order」を SIMUL にすると, 複数のグラフが同時に (simultaneously) 描画されます。SEQ では, 定義した順番に (in sequence) 描画されます。
(4) 「Grid」を ON にすると, 画面全体に格子縞が現れ, OFF にすると消えます。試しに, ON にしてみてください。
(5) 「Axes」の箇所を OFF にすると座標軸が消え, ON にすると座標軸が表示されます。
(6) 「Leading Cursor」を ON にすると, グラフ描画の際の先端に十字カーソルが現れます。その分, 描画スピードが遅くなりますが, 多数のグラフが混みいっているとき, この「Leading Cursor」を ON にしておくと, 今どの部分を書いているのかが明確になります。必要なときに「ON」にしてください。
(7) 「Label」を ON にすると, 座標軸の名前の x, y が表示されます。これは, 必要がなければ OFF

のままでよいと思います。

幾つかの Tips

(1) 一部を黒く塗りつぶすには, 関数を太線で描画して, その関数を少しずつずらしながら塗りつぶすことで実現できます。太線にするには, 太くしたいグラフを黒く反転させてから, [2nd] [F1] で「4:Thick」を選択します。

(2) すべての関数を一気に削除するには, [F1] 8 を押します。実際にそれを押すと, 本当に削除してよいかどうかの確認が求められます。

(3) グラフを描画させる関数は, [F4] でチェック印のつけられた関数です。関数が増えてくると, 個々の関数にチェックをつけたり取ったりするのは面倒です。すべての関数にチェック印をつけたり, 全ての関数のチェック印を外したりするには [F5] を利用します。 [F5] 1 で全ての関数のチェックをはずし, [F5] 2 で全ての関数にチェックが付けられます。

参考例の関数

この節の最初で例示した図は, 以下の関数で描画されています。自分でこれらの関数を定義し, 太線や斜線による塗りつぶしなどの指定を行って同じような図になるように調整してみてください。

$y1 = -\sqrt{x+7.5} + 1.7$

$y2 = -1/7 \cdot x + 0.7 \mid x > -7.7$

$y3 = \dfrac{-x}{5} + 1/5 \cdot \sin(3 \cdot x) \mid x > -7.6$

$y4 = \sqrt{2^2 - (x+4.8)^2} - 4.8$

$y5 = (x+4.9)^2 - 3.5 \mid -5.5 < x < -4.4$

$y6 = \sqrt{x+4.9} - 3.3 \mid x < -3.5$

11.2　媒介変数表示された関数によるグラフアート

この節では, 媒介変数表示された関数を利用したグラフアートの作成方法について解説します。この節を理解するには, 媒介変数表示された関数について学習中か, 学習済みであることが前提となります。媒介変数表示された関数は, 2 年か 3 年の微分積分で学びますが, 使用教科書により学習学年が異なります。また, 媒介変数表示された関数のグラフ表示のさせ方 (第 5 章) も理解しているものとします。

最初に, 媒介変数を利用したグラフアートの作品例を紹介しておきましょう。$y = f(x)$ タイプの関数を利用するのと比べて, 縦線の一部や円を綺麗に描画することができます。いずれも, 平成 18 年度の 2 年生が作成したものです。

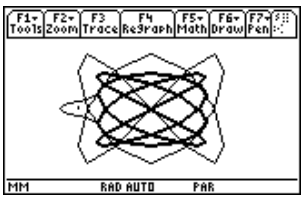


図 1：かめさん

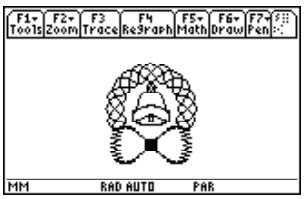


図 2：リース

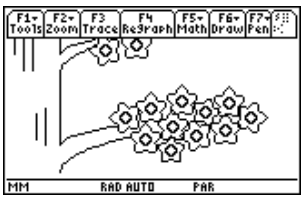


図 3：春の訪れ

図 4：雪だるま

11.2.1　媒介変数表示によるグラフアートの作成

媒介変数表示された関数のグラフを繋いで絵 (アート) を作成するには, 個々の関数のグラフの描画範囲を指定する必要があります。その指定は, [|] を利用する。 [♦] [F1] の関数定義画面に戻り, 描画

範囲を指定したい関数の x か y の式を指定して, その一番最後で [|] を押し, その後に t の範囲を指定します。その指定は, x か y の片方だけでよく, たとえば円の上半分だけを表示させるには, [|] の後に $0 < t < \pi$ を追加します (図 28, 図 29)。範囲指定は, 「$0 < t < \pi$」という形で入力します。この範囲指定を, 個々の関数について指定していくことになります。

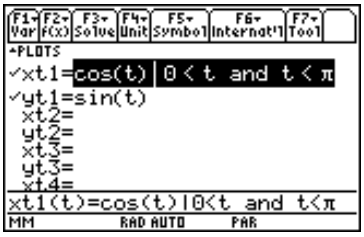


図 28

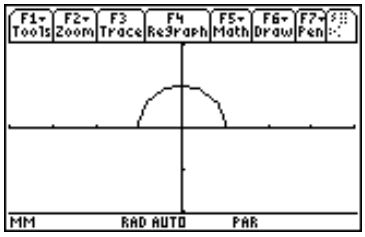


図 29

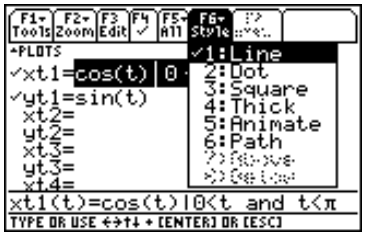


図 30

グラフの描画スタイルの設定

個々のグラフの描画方法も細かく指定することができます。標準では, t の値を少しずつ変えながら (x, y) を計算し, その点を線分 (Line) で繋いでグラフが描画されます。その繋ぎ方を, 点線 (Dot) や太線 (Thick) などに変更することができます。その指定は, 関数の定義画面 ([♦] [F1]) で描画方法を指定したい関数にカーソルをあて, [2nd] [F1] を押すと現れるメニュー画面 (図 30) で行います。個々の関数に対して行なうことになりますが, その指定は x, y のいずれか一方に対して行えば十分です。なお, いろいろな線種でグラフを描画できますが, $y = f(x)$ のときのような「塗りつぶし」の機能は利用できません。

グラフの描画形式の設定

座標軸の表示の有無など, 画面全体のフォーマットを細かく指定するには, [F1] 9 (または, [♦] [|]) を押します (図 23)。前節と同様に, 「Graph order」はグラフを描画する順番, 「Grid」は画面全体に格子模様を表示するかどうか, 「Axes」は座標軸を表示するかどうか, 「Leading Cursor」はグラフ描画のとき先頭にカーソルを表示するかどうか, そして「Labels」は座標軸の名前を表示するかどうかを指定できます。

条件判断コマンド「When」

条件判断のコマンドに「When」を利用できます。このコマンドを利用すると, いろいろな条件に応じて複数の式を切り替えて 1 つの関数として定義することができます。When の書式は「When(条件, 真の場合の式, 偽の場合の式)」です。例えば, $0 < t < \pi$ では単位円の上半分 $x = \cos t$, $y = \sin t$ であり, $\pi < t < 2\pi$ では放物線 $x = \cos t$, $y = 1 - \cos^2 t$ とする場合は,

$$x = \cos(t), \quad y = \mathrm{when}(t < \pi, \sin(t), 1 - (\cos(t))^2)$$

として定義します。when を入れ子にすることで, 複雑なグラフも一つの関数で定義できます。

11.2.2　グラフアートの具体的な構成関数

ここでは, 最初に提示したグラフアートの具体的な関数について紹介しておきましょう。図 1 の構成関数は次の 5 つの媒介変数表示の関数で作成されています。(2) と (4) が亀の首の部分であり, (5) は亀の目の部分です。(3) のグラフには, 太線指定がなされています。

(1) $\begin{cases} x = 4\sin(4t) \\ y = 3\sin(6t) \end{cases}$

(2) $\begin{cases} x = -\cos(4t) - 4 \\ y = t \,|\, t < 4/5 \end{cases}$

(3) $\begin{cases} x = 3\sin(4t) \\ y = 2\sin(3t) \end{cases}$

(4) $\begin{cases} x = -\cos(4t) - 4 \\ y = -t \,|\, t < 4/5 \end{cases}$

(5) $\begin{cases} x = -4 \\ y = 1/7 \end{cases}$

図 2 のリースは 14 組の関数で構成され, 柄の部分は 3 本の曲線で描かれています。その一つは

$$x = \frac{1}{2}\sin(5t)\cdot\cos(t) + 2\cos(t), \quad y = \frac{1}{2}\sin(5t)\cdot\sin(t) + 2\sin(t)$$

という関数です。また, リボンの右側は次の関数です。

$$x = 2\sin(t/2) + \frac{1}{4}\sin(23.5t), \quad y = \sin(t) - 2$$

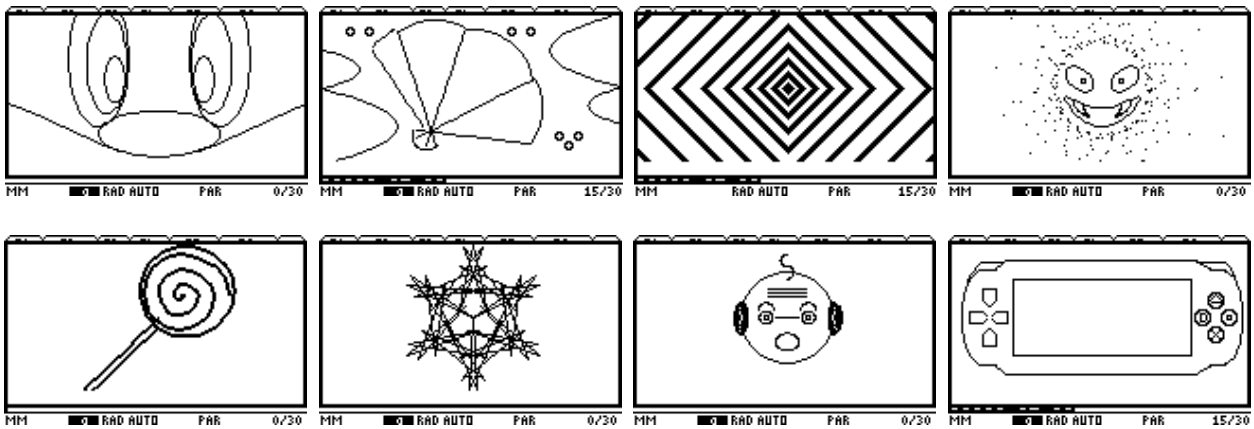
図 3 は 41 組の関数で構成されており, 個々の花びらは 1 組の媒介変数表示の関数で描画されます。微妙に形が異なっており, 作成した学生のこだわりがうかがわれます。1 つの花びらは, たとえば次のような関数で描画されます。

$$x = -\frac{1}{7}\cos(t)\cdot(5 - \sin(5t)), \quad y = -\frac{1}{7}\sin(t)\cdot(5 - \sin(5t))$$

いずれも, いろいろな試行錯誤をしながら学生が自分で見い出したものです。単なる試行錯誤ではなく, 式を変えることにより x はどのような変化をするか, y はどのような動きをするのかなどを理解して, 関数の式と実際の点の動きとの対応関係を把握できるようになることが重要です。関数の式を見ているだけではそのような理解は得られません。数ナビで実際にグラフを描画させ, 多数のグラフを眺めているうちにそのような理解が得られていきます。

次の作品も, 2 年生の作品です。それぞれ, どのような関数で作成されているか考えてみてください。

,



このマニュアルは, 独立行政法人日本科学技術振興会の平成 22 年度〜24 年度科学研究費補助金 基盤研究 (C) 課題番号 22500830 (研究代表者：梅野善雄) の支援を受けて作成されたものです。