

(ナビ) スクリプト[アニメーション(動き)とサウンド(音)]とテーブルについて、準備・前編・後編に分けて紹介します。

[準備]

アニメ(動き)の試作

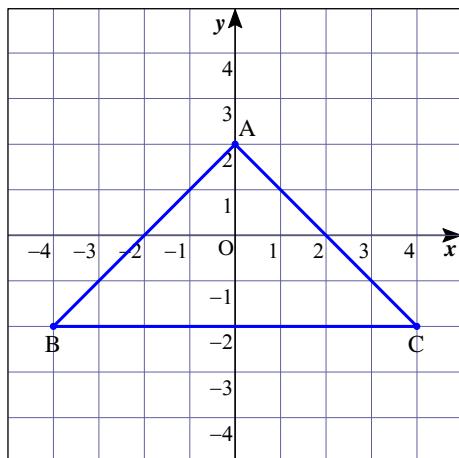
[準備 1] スクリプトの説明

スクリプトとは、GRAPES 内で動く小さなプログラムです。繰り返しやサブスクリプトの呼び出しなどを含んだ制御構造をもち、自動実行条件の設定ができます。

スクリプトを使うと、パラメータの増減やグラフ描画をコントロールすることができ、あらゆる式の計算はもちろん、関数のグラフを波形を持つ音の出力もできます。

データパネルの [ノート] 内の [スクリプト] に記述します。

[PDF マニュアル]
の第 14 章参照



[準備 2] GRAPES ファイルを作る

グラフウインドウのそれらしい位置（座標）で、マウスを右クリックして、プルダウンメニューから [点を打つ]、[A] を押して、点を表示します。続いて、B、C も表示して下さい。ちなみに、[CTRL] キーを押しながら点をドラッグすると格子点にピタッと点を置くことができます。

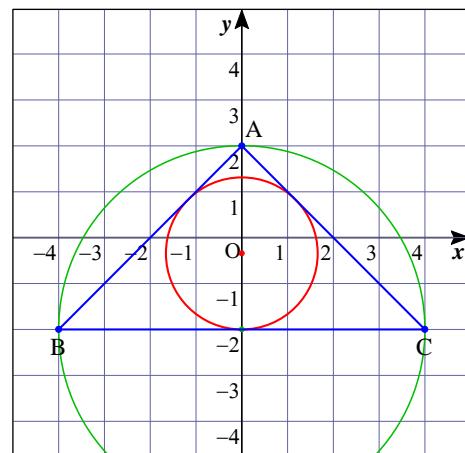
そして、同じくプルダウンメニューから [点を結ぶ] を選び、[連結図形] の [頂点] を A B C とし、図形を [多角形] にしましょう。

続いて、基本図形 P の入力をしますが、図形 P の種類を円とし、△ABC の内接円に設定します。このとき、関数「内心」や「内径」は、関数電卓の [関数 3] タブにあります。



続いて、図形 Q を、三角形 ABC の外接円とします。

色の設定は、適当にして下さい。



[準備 3] スクリプトを書く

さて、スクリプトを記述しましょう。[ノート] の [スクリプト] タブを選び、下記のようにスクリプトを書いてください。右には、スクリプトの解説を載せました。

//AUTO	スクリプト「AUTO」の命名とボタン作成
p:=A.x	p に点Aの x 座標を代入
q:=A.y	q に点Aの y 座標を代入
for @:=0 to 2Pi step Pi/100 …※	$\theta = 0$ から 2π まで、 $\pi/100$ 刻みで繰り返す
A.x:=p+3sin2@	点Aの x 座標に、 $p+3\sin 2\theta$ を代入
A.y:=q+2sin@	点Aの y 座標に、 $q+2\sin \theta$ を代入
draw	図形を描く
next @	θ に次の値を代入し※に戻る

さあ、[AUTO] ボタンを押して、スクリプトを動かしてみましょう。

頂点Aがリサジューを描くと同時に、内接円も、外接円も動きますか。



(ナビ) さあ、続いて本格的な作品(?)を1つ作ってみましょう。

[前編] アニメ(動き)とテーブル

[前編 1] テーブルについて

GRAPES は新機能「テーブル」を持ちました。テーブルとは縦200行、横10列の2,000セルに数値を記憶できる表です。データパネルの [ノート] 内の [テーブル] にあります。

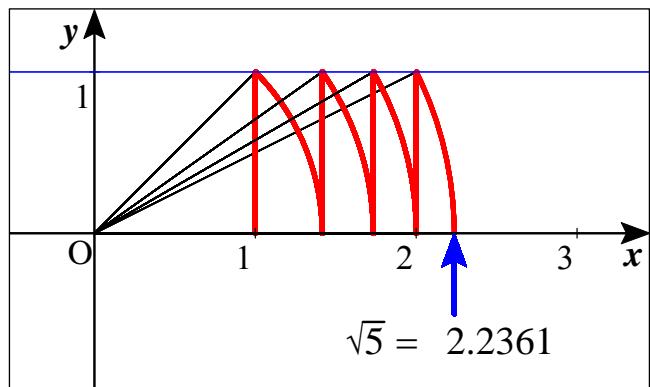
カーソルキーでセル間を移動できます。Enter キーで下のセル、Tab キーで右のセル、Shift+Tab キーで左のセル、マウスクリックで任意のセルに移動します。セルに入力できるのは数値だけです。

[PDF マニュアル]
の第 11 章参照

[前編 2] 題材「 \sqrt{n} の作図とその値」

幅 1 の平行線の一方の直線（図では x 軸）に原点から 1 距離にある点から垂線を立て、もう一方の平行線との交点を考える。その点と原点を結ぶ線分（つまり、直角三角形の斜辺）の長さは $\sqrt{2}$ となります。そこで、 $\sqrt{2}$ の長さの線分を水平まで回転し x 軸との交点を考え、その点から垂線を立て…、と、これを続けていきます。

この動きのイメージを、テーブルを使用したスクリプトによって、アニメーション的（動的）に表現してみましょう。そして、



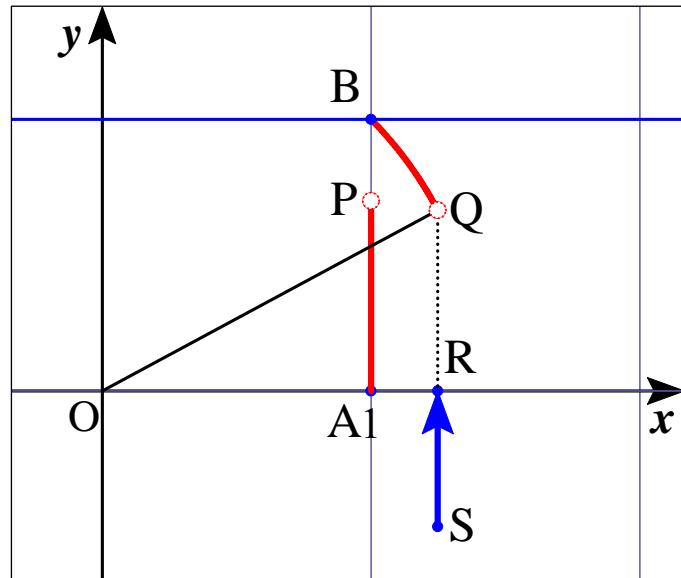
\sqrt{n} の値の計算方法も、この作図方法によって、順次計算していく手法をとります。

(注) この題材は「 \sqrt{n} の値をあらかじめ計算しておいて、それを元に図を描く」という手法ではありません。

[前編 3] 部品を作成

「 \sqrt{n} の作図とその値」を実現するため、右図のように 6 つの点を考えます。点 A(1,0) とし、点 A の真上の点を B とします。点 A から上昇する途中の点を点 P とし、点 B が回転していく途中の点を Q とします。点 Q の真下の矢印を表現するために 2 点 R, S を考えます。

さあ、具体的に GRAPES に入力していきましょう。



右の表を参考に、[基本図形] の作成で [点] を選び、3 点 A, B, P のプロパティを設定して下さい。

(注) (空欄は任意)

入力が終わったら、パラメータ s を動かして点 P の動きを確認してください。

	種類	図形	入力	色	太さ	●○	ラベル
基本 図形	点	A	(1,0)			●	非表示
	点	B	(A.x,1)			●	非表示
	点	P	(A.x,s)	赤	最大	○	表示

次は、点 Q の設定です。点 Q は [基本図形] ではなく、[曲線] の作成で [媒介変数グラフ] を選びます。プロパティは、次のようにして下さい。

	種類	図形	入力	変域	色	太さ	●○	ラベル
曲線	媒介変数グラフ	Q	$rot(B,O,-t)$	$0 \leq t \leq t$	赤	最大	○	表示

変域 $0 \leq t \leq t$ について、変数 t の区間が $[0,t]$ となっていますが、間違いではありません。曲線の下端に点 Q を表示する方法です。

ここでも、パラメータ t を動かして点 Q の動きを確認してください。

あと、2 つの点 R, S を [基本図形] の作成に戻り下の表を参考に設定して下さい。

	種類	図形	入力	色	太さ	●○	ラベル
基本 図形	点	R	(Q.x,1)				非表示
	点	S	$R - (0,0.5)$	青		●	下 $\{Sqrt(!\{k\}) =\} !\{ [O B]\}$

以上で、部品として 6 つの点の入力が済みました。

ここで、[指定領域を拡大] ボタンを使用して、グラフエリアを、
領域 $\{(x,y) \mid 0 \leq x \leq 4, -2 \leq y \leq 2\}$

が大きく表示出来る程度に拡大してみましょう。

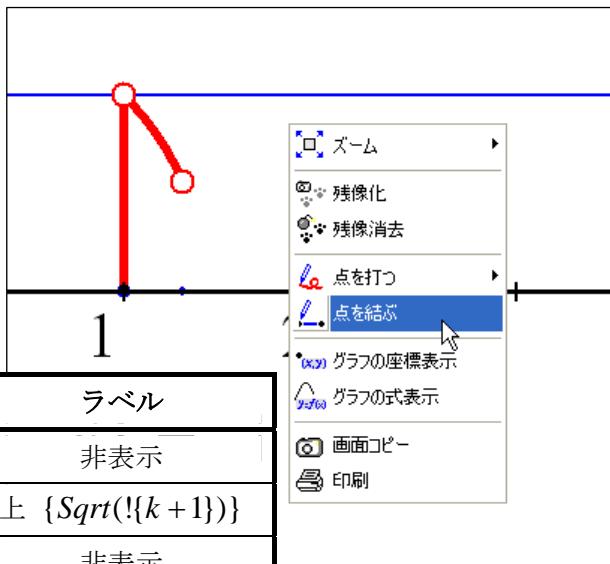
そして、目盛り文字サイズや目盛り幅を広げるボタンなどで、適当に表示を見やすくして下さい。

新たに4つの部品（3つの線分と1つの矢線）を【連結図形】で作ります。

グラフエリア上で、マウスを右クリックし【連結図形】の【点を結ぶ】ボタンを押し込んだ状態にします。

そして、表のように4つの連結図形を作りましょう。このとき、作成は上から順番にして下さい。

記号	頂点	図形	色	太さ	線種	ラベル
L 1	A P	線分	赤	最大	直線	非表示
L 2	O Q	線分	黒	中	直線	上 {Sqrt(!{k+1})}
L 3	S R	矢線	青	最大	直線	非表示
L 4	R Q	線分	黒	中	点線	非表示



4つの作成が終わったら【連結図形】ボタンをもう一度押して元に戻しておいて下さい。

最後に、陽関数の作成で $y1=1$ と入力しましょう。プロパティは適当に…？！

部品が全部そろったところで、パラメータ s, t を動かして動作の確認をして下さい。

◎点Pは上下に移動しますか？

◎点Qは円を描きますか？



【前編 4】スクリプトの記述

「HideScript」は初めは書かないで下さい。

最初は、「// (ダブルスラッシュ)」以降を書いて下さい。こうすることで、ボタンが【ノート】エリアに表示され、動作確認ができます。動作確認後「HideScript」を書き加えて下さい。このような手順が良いと思います。

さあ、メモエリアの【編集】ボタンをクリックし、【スクリプト】タグの背景が薄いブルーの編集窓にスクリプトを順に書いていきましょう。
スクリプトは全部で5つあります。

- ① 「CL」：部品を最初の状態にします。
- ② 「UP」：点Pを点Aから点Bの位置まで移動させます。
- ③ 「DN」：原点中心で点Qを回転しながらx軸に交わる点まで移動します。

- wait(500) : 500msec 作業を止めます。
- setp 0.02 , step 0.015*p の数値を変えると動きの速さを変えることができます。

```

ノート
ステッカー スクリプト テーブル ノート
HideScript//CL
ShowObj(P,S,L3)
HideObj(L1,L2,L4,A,B,Q,R)
s:=0
t:=0

HideScript//UP
ShowObj(L1,P,L3)
Wait(500)
for s:=0 to 1 step 0.02
draw
next s
wait(500)
ShowObj(L2,L4,Q)

HideScript//DN
HideObj(P)
Wait(500)
p:=arg(A,0,B)
For t:=0 to p step 0.015*p
Draw
Next
wait(500)

```

ノート

ステッカー スクリプト テーブル ノート

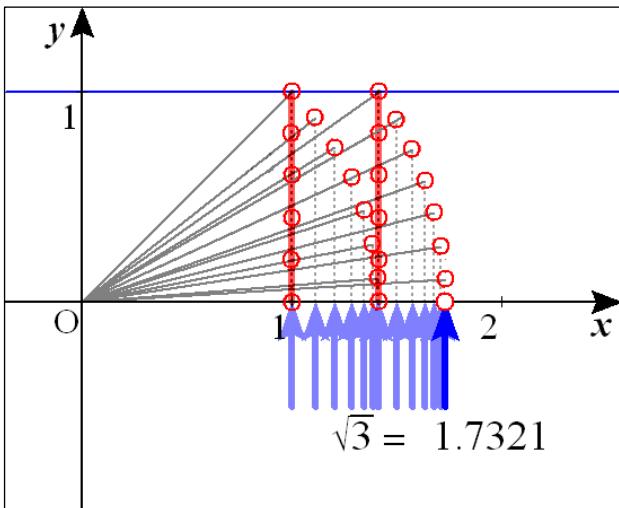
```
//初期化
n:=1
ClrAllCells
Call(CL)
Cells(1):=1
A.x := Cells(n)
draw

//次を求める
Call(UP)
Cells(n+1):=[OB] ←
Call(DN) ←
inc(n)
Call(CL)
A.x := Cells(n) ←
draw
```

OK 適用 キャンセル

[前編 5] 仕上げ

[ステッカー]に右のように入力して下さい。
この説明文が [ノート] に表示されます。



- 残り 2 つのスクリプトを書きましょう。
- ④「初期化」: テーブルを初期状態にし、点 A を定義します。
 - ⑤「次を求める」: 一連の動作を 1 回します。

㊂ \sqrt{n} の値を順次計算していく手法について、

- (1) Cells(n+1):=[OB]
(2) inc(n)
(3) A.x:=Cells(n)

の 3 行で、順次計算していく手法を実現しています。つまり、セルに斜辺の長さを代入し、その値を次の点 A の x 座標に設定しており、帰納的な定義となっています。

全て入力できたなら、動作確認をして下さい。
確認ができたら、「CL」、「UP」、「DN」の前の
「HideScript」を書き加えて下さい。

もう、ほとんど完成ですね。



ノート

ステッカー スクリプト テーブル ノート

{Sqrt(n)}の作図方法の説明

[初期化] で {n=1} がセットされる。
[次を求める] で {Sqrt(n+1)} が求まる。

「編集」>「テーブル」を表示したまま実行ボタンを押してください

表示 背景色 文字色 字サイズ
memo 緑 黒 12

OK 適用 キャンセル

上の図は完成動作イメージです。このような動作になりますね。

さらに、[ノート] の [テーブル] を表示したまま、[次を求める] ボタンを押してみましょう。

\sqrt{n} の値が、順次計算され、テーブルのセルに次々と代入されていく様子が確認できますね。

(ナビ) さて、GRAPES で音を出してみましょう。

[後編] サウンド(音)とテーブル

[後編 1] 音を出す

実際に音を出してみましょう。音を出すには、Play コマンドを用います。Play コマンドのパラメータは次のように 4つあります。

Play ((関数の) 変数, 開始時, 終了時, (音の波形の) 関数)

```
//sound  
Play(x,0,2,sin880Pix)
```

例えば、左のスクリプト sound では、「変数 x の (音の波形の) 関数 $y = \sin 880\pi x$ を 0 秒から 2 秒まで鳴らす」となります。

このスクリプトを実際に入力してみましょう。

そして、スクリプトを実行し、音が出ることを確認しましょう。

また、ステレオで出力したいときには、データを 2-dim のデータを与えます。つまり、

```
Play(x,0,2,(sin880Pix,882Pix))
```

とすれば、左チャンネルからは、440Hz、右チャンネルからは、441Hz の音を出します。

GRAPES が演奏を開始する（音を出す）のは、すべてのデータの計算終了後です。

また、引数なしの Play コマンドは、作業用メモリに残されたデータを演奏します。

[後編 2] GRAPES ファイルを作る

さて、実際に GRAPES ファイルを作ります。テーブルにデータを入力し、そのデータを用いたいろいろな音の波形を表示し、同時にその音を出すファイルを作ります。

3つの部分に分けてファイルの作り方を説明します。

(2・1) 音の波形の関数を作る

データパネルで 2 つの関数定義と、3 つの陽関数を作成して下さい。パラメータは m, a, c, b, d です。(順序に気をつけてください。)

この音の波形の説明を少し加えます。

例えば、 $(a, b, c, d) = (1, 0, 1, 0)$ としたときの関数 $y1$ を t 秒間発音することを考えましょう。つまり、 $\text{Play}(x, 0, t, y1)$ です。

GRAPES のデフォルトでは、 $m = 1$ ですから、 $y1 = g(f(x)) = \sin(2\pi x)$ となります。この関数の周期は、1 となります。振動数は、周期の逆数なので 1Hz です。

人間は、20 Hz から 20 kHz の振動を「音」として感じるようです。したがって、このままで Play コマンドを実行しても「音」になりません。ですから、 $m = 440$ などとしてこの関数を音の波形とします。

ちなみに、ピアノの中央の「ラ」がほぼ 440Hz だそうです。

$y2 = g(g(f(x)))$ や $y3 = g(g(g(f(x))))$ は、関数 $g(x)$ へ代入を繰り返すことで、波形を複雑に変形してくれる変換になっています。最後には、実際に音を聞いて確認してください。

関数定義	$f(x) =$	$2\pi mx$
陽関数	$g(x) =$	$a \sin cx + b \cos dx$
陽関数	$y1 =$	$g(f(x))$
	$y2 =$	$g(g(f(x)))$
	$y3 =$	$g(g(g(f(x))))$

(2・2) テーブルにデータを入力

多くの数値を GRAPES にあらかじめ覚えさせておく、という目的のためにテーブルを使用します。(もちろん、テーブルには、もっと別の使用方法もありますが…。)

関数 $g(x) = a \sin cx + b \cos dx$ の係数の組 (a, b, c, d) に加えて、 t を音の長さ、 m を元になる音の周波数とします。すると、6-dim のベクトルで音を定義できます。

その音を定義する 6-dim のデータを 10 組用意することにしましょう。

The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - Sound_data.xls". The table has columns labeled A through F. Row 1 contains headers: w, x, y, z, t, m. Rows 2 through 10 contain data points. Row 11 is empty. Row 12 is a footer row.

w	x	y	z	t	m
1	2	2	1	1	440
2	0.5	0.5	2		
3		0.3			
4	2	1			
5	2.3				0.7
6	2	3		1.005	2
7	1	4			1
8	2	3	1.5	5	880
9		4	2.5	9	
10	5	3.5	10		
11					
12					

The screenshot shows the GRAPES Note application window. It has tabs for StepCar, Script, Table, and Note. The Table tab is selected, showing a table with columns labeled 1 through 7. Rows 1 through 10 correspond to the data in the Excel table above. Row 11 is empty.

1	2	3	4	5	6	7
1	2	1	1	1	1	440
2	0.5	0.5	2			
3		0.3				
4	2	1				
5	2.3					0.7
6	2	3		1.005	2	
7	1	4			1	
8	2	3	1.5	5		880
9		4	2.5	9		
10	5	3.5	10			
11						

[テーブル] のセルへの数値入力は、直接できますが、エクセルのデータが作ってあります。ファイル名は、「Sound_data.xls」です。そのファイルを開いて、A1 から F10 までを、[コピー&ペースト] して下さい。

セルが空白の場合、そのセル値は代入されません。したがって、 a, b, c, d, t, m のそのセルに対応する値は変更されません。

(2・3) スクリプトについて

さて、スクリプトを記述しましょう。[ノート] の [スクリプト] タブを選び、右のように 3 つのスクリプトを書いてください。
〔注〕「…① 等は入力しません。」

① ダミーのスクリプト「D」

パラメータエリアにパラメータ t を表示させます。

他のパラメータは関数定義や陽関数で定義されているので自動的に表示されています。

② スクリプト「SET」

(a, b, c, d, t, m) に各セルの値を代入するスクリプト。

n 行目のデータを左から順に a, b, c, d, t, m に代入します。

③ スクリプト「N1」

n の値を 1 増やす。ただし、 $n = 10$ の次は $n = 1$ とします。

HideScript//D …①

on t change

HideScript//SET …②

a:=Cells(n,1)

b:=Cells(n,2)

c:=Cells(n,3)

d:=Cells(n,4)

t:=Cells(n,5)

m:=Cells(n,6)

HideScript//N1 …③

inc(n)

if n>10 then n:=1

各行の先頭に書いてある「HideScript」とは、「スクリプトの実行」ボタンを表示しないという命令です。

```

//乱          . . . ④
n:=0
@:=rnd(1)
a:=2+3rnd(1)
b:=2+cos(4@)*rnd(3)
c:=1
d:=cos(2@)*rnd(1)*5
Call(y1)

//初期化      . . . ⑤
n:=1
Call(SET)
Call(y1)

```

④ スクリプト「乱」

$n = 0$ とし、 a, b, c, d に適当な乱数を代入し、サブスクリプト「y1」を呼び出すことによって、関数 $y1$ の音を鳴らします。

⑤ スクリプト「初期化」

$n = 1$ とし、サブスクリプト「SET」を実行した後、サブスクリプト「y1」を実行します。

つまり、プリセット No 1 のデータセットで、関数 $y1$ の音を鳴らすということです。

```
//y1          . . . ⑥
```

```
ShowObj(y1)
```

```
HideObj(y2,y3)
```

```
draw
```

```
Play(x,0,t,y1)
```

```
//y2          . . . ⑦
```

```
ShowObj(y2)
```

```
HideObj(y1,y3)
```

```
draw
```

```
Play(x,0,t,y2)
```

```
//y3          . . . ⑧
```

```
ShowObj(y3)
```

```
HideObj(y1,y2)
```

```
draw
```

```
Play(x,0,t,y3)
```

```
//次のプリセット(y1) . . . ⑨
```

```
Call(N1)
```

```
Call(SET)
```

```
Call(y1)
```

```
//次のプリセット(y2) . . . ⑩
```

```
Call(N1)
```

```
Call(SET)
```

```
Call(y2)
```

```
//次のプリセット(y3) . . . ⑪
```

```
Call(N1)
```

```
Call(SET)
```

```
Call(y3)
```

⑨ スクリプト「次のプリセット(y1)」

サブスクリプト「N1」, 「SET」, 「y1」を連続して実行します。その意味は、 n を 1 つ増やし、新しいプリセットデータを入力し、最後にその関数 $y1$ の音を鳴らします。

⑩, ⑪は、⑨の「y1」を「y2」, 「y3」に読み替えて下さい。



[後編3] グラフの表示と音

さて、もうすぐ、完成ですね。

最後に、[ノート] の [ステッカー] に次のように記入し、GRAPES の実行時にデータが表示されるようにしましょう。

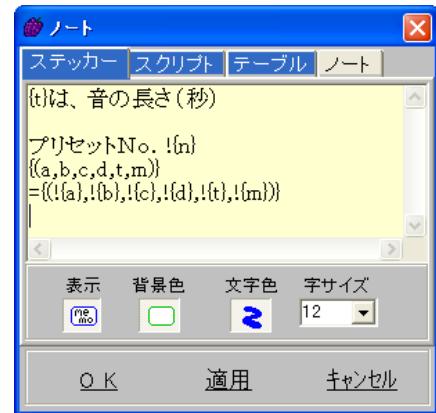
{ } ! () , の記号の区別、有無に注意して下さい。

{t} は、音の長さ（秒）

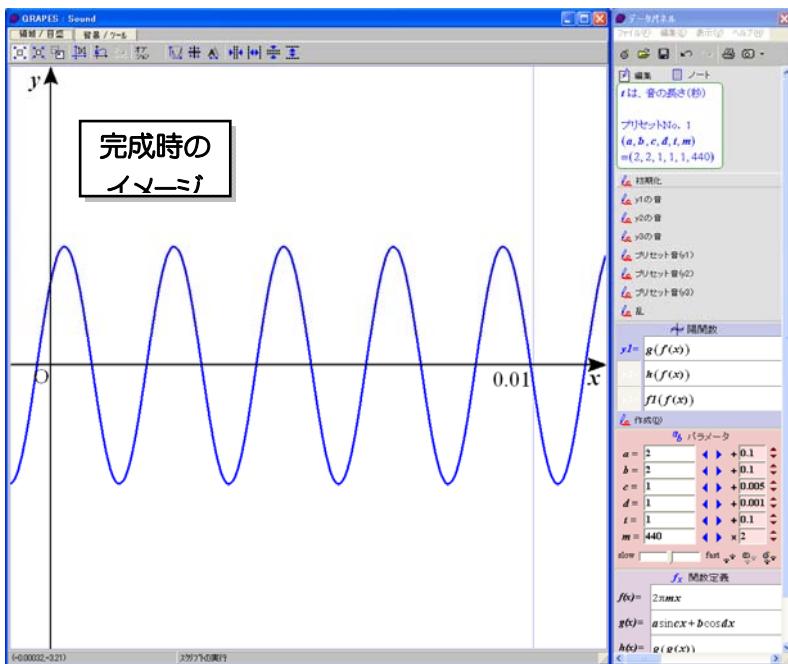
プリセット N o. !{n}

{(a, b, c, d, t, m)}

={{(!{a}, !{b}, !{c}, !{d}, !{t}, !{m})}}



(↑ナビ) 記号が見やすいように、ここでは、半角のブランクが入れてあります。



ボタンの表示順は、スクリプトを書いた順です。

設計通りグラフが表示され、そして、音が出るか、いろいろ確認してみましょう。

スクリプト「乱」を押したとき、どんなグラフと音が出てくるのでしょうか。



コマンドと式 ○スクリプトは、コマンドと式でできています。コマンドに大小文字の区別はありませんが、式は大小文字を区別して書く必要があります。例えば、“Draw”は“DRAW”でも“draw”でも構いませんが、円周率“Pi”を“PI”や“pi”と書いてはいけません。

スクリプトの記述方法 ○順序は気にする必要はありません。

○各スクリプトは、1行の空行で分離して下さい。

スクリプトの個数 ○スクリプトは最大20個作ることができます。けっこう余裕あり。(^-^)v

追加説明 ○最初の1行が、スクリプトボタンとして表示されます。

○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ります。

○1行には1命令しか書くことができません。

○語と語の間には、半角空白を入れます。

[おまけ] スクリプトを使用したファイルいろいろ

[おまけ 1] マニアの部屋について



少し前のバージョンから GRAPES のサンプルの構成が変更されているのを知っていますか。その新しいフォルダーに「マニアの部屋」があります。最後に、これにどうしても触れなくてはなりません。この部屋の中にあるファイルは、是非すべて見ていただきたいのです。

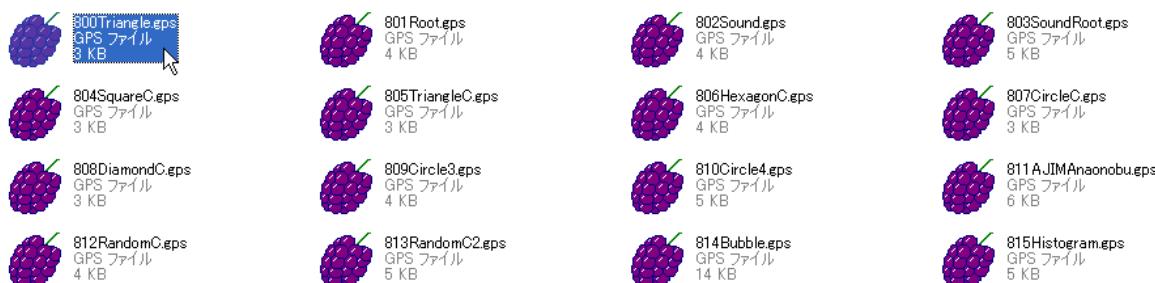
特にお勧めは、「ハノイの塔」です。いや、どれもお勧めです。

- ・ 「2次関数+1次関数」は、「放物線に直線を加えるイメージ」を作り出しています。
- ・ 「サイクロイド_自転車」は、自転車が画面登場し、サイクロイドを描いていきます。
- ・ 「ハノイの塔」は、文字通りハノイの塔のシミュレーションをしてくれます。
- ・ 「ピタゴラスの定理」は、面積の移動で定理が成り立つ事をみせます。
- ・ 「弦の包絡線」は、円周上の等分点を結ぶ線分が美しい模様を作り出します。
- ・ 「楕円内の光の反射」は、楕円形内でビリヤードをするシミュレーションです。
- ・ 「内トロコイドギヤ」は、昔流行った「スピログラフ」のPC版です。
- ・ 「二分法」は、 $\sqrt{5}$ の値を順を追って、正確に求められるシミュレーションです。

[おまけ 2] 今年の講習会用のサンプルについて

配布 CD-Rの中には、テキストで紹介したファイルの他に、別のスクリプトを利用したファイルが入っています。参考にしてください。([堀部] > [GRAPES_BOX])

- ・ このテキストの【準備】のファイル——800Triangle.gps
- ・ このテキストの【前編】のファイル——801Root.gps
- ・ このテキストの【後半】のファイル——802Sound.gps
- ・ 801Root.gps の音付きバージョン——803SoundRoot.gps



- ・ 図形の中に複数円の配置
 - ・ 和算からの問題
 - ・ 重ならない円の配置
 - ・ 泡の上昇
 - ・ ヒストグラムを表示
- 804~808
——809~811
——812、813
——814
——815



参考 URL <http://horibe.jp>