

Explore コマンドの更新

[Explore](#) 関数では、現在のドキュメントやワークシート内に対話型のアプリケーションを作成するための使いやすいツールを提供します。Maple 2015 では、次のような大幅な更新と追加が行われています。

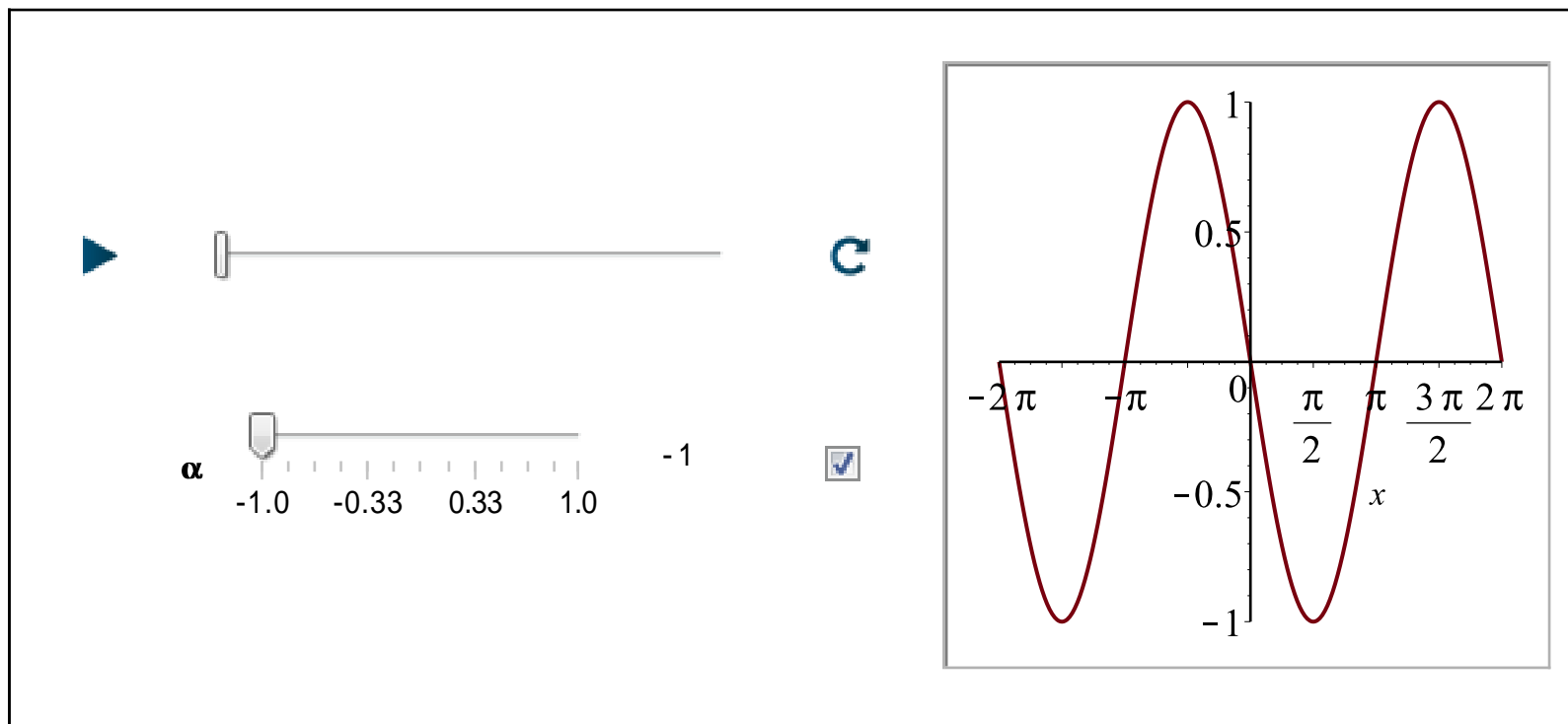
- 2次元プロットでのカスタマイズ可能な対話型マーカーコントロール
- 再生/一時停止/繰り返しアニメーションコントロールの画像
- スライダーコントロールのあるパラメータの 2-D Math
- 縦方向のスライダーコントロールのサポート
- `true` および `false` の値となるパラメータの [CheckBox](#) コントローラ
- リストからの値となるパラメータの [ListBox](#) コントローラ
- 新規ドキュメントとして起動する場合の[スタートアップコード](#)の割り当て
- 新規ドキュメントとして起動する場合の[ドキュメントのプロパティ](#)

追加の詳細と実例は、[Explore のワークシート例](#)を参照してください。

▼ アニメーションコントロールの画像、パラメータ名の 2D Math

[再生/一時停止] ボタンおよび [繰り返し] ボタンに、画像が作成されました。スライダーコントロールの横のパラメータ名は、2-D Math を使用して表示されます。

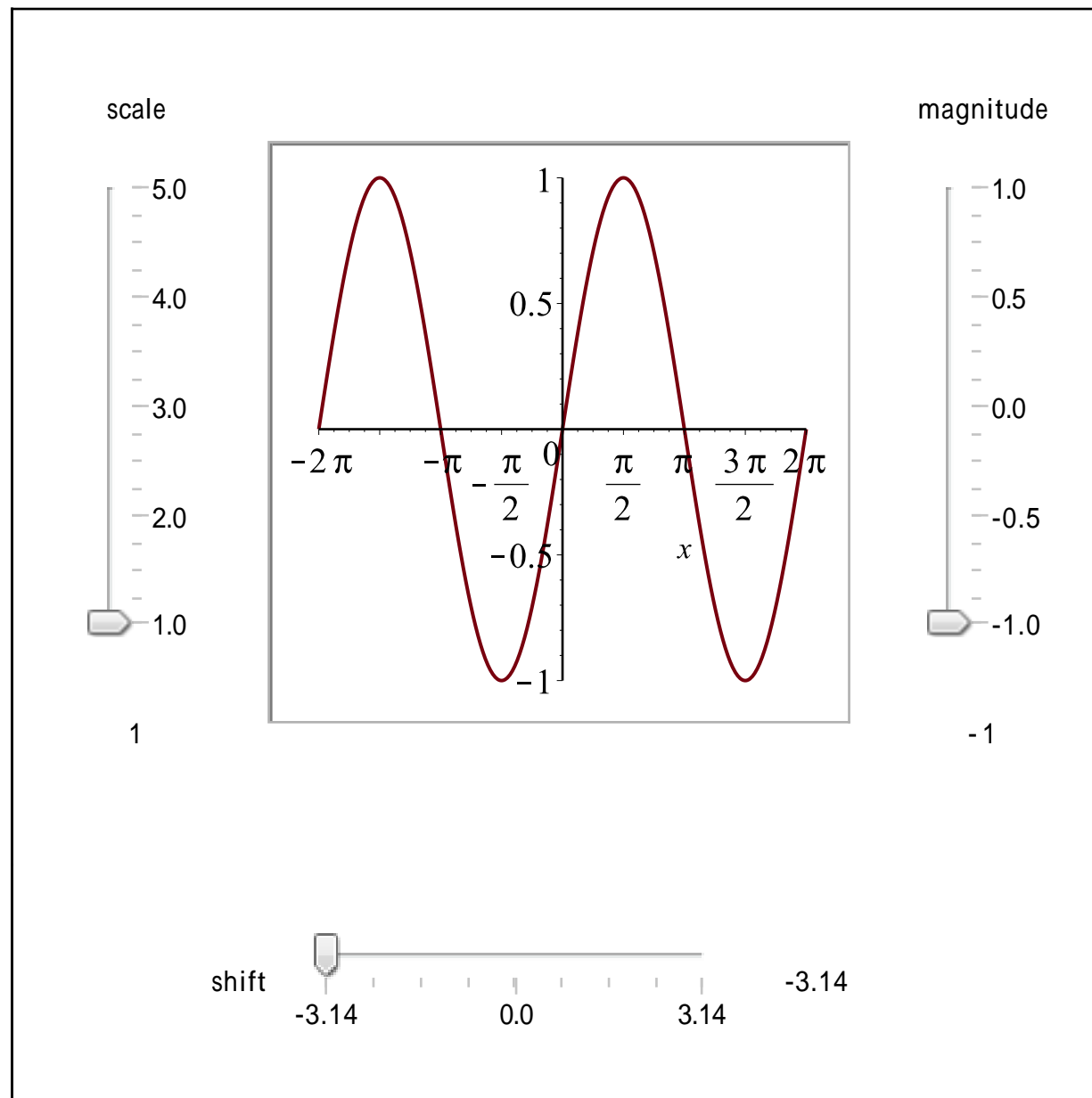
```
Explore( plot(  $\alpha \cdot \sin(x)$ ,  $x = -2 \cdot \pi .. 2 \cdot \pi$ , view=-1..1 ),  
         parameters = [  $\alpha = -1.0 .. 1.0$  ],  
         placement = left, animate, loop, size = [300, 300] )
```



▼ コントロールの方向

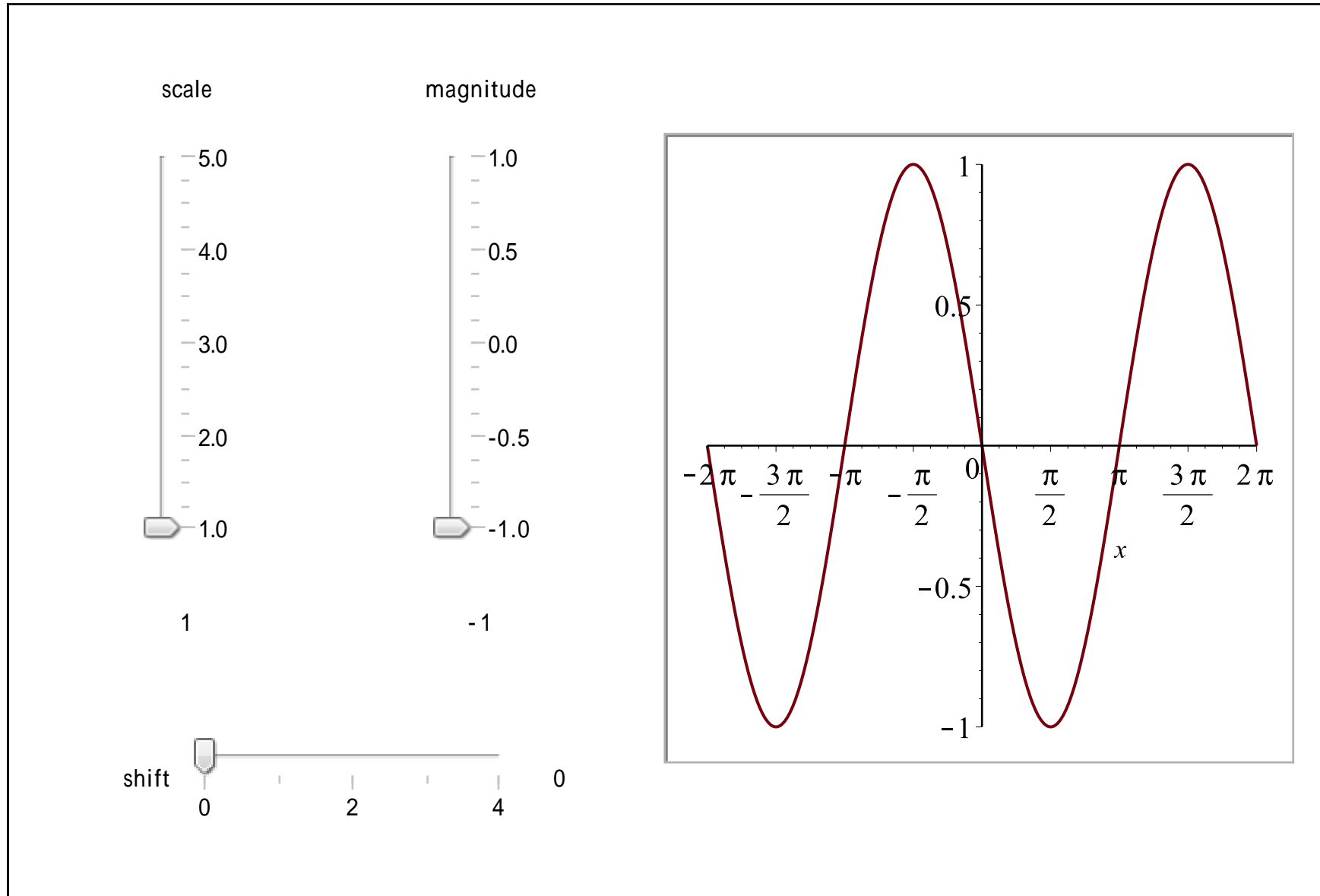
スライダコントロールの方向について、その他のコントロールの特性と同様に指定できるようになりました。すべてのスライダコントロールに個別の方向オプションを使用できますが、これらは各パラメータの個別指定でオーバーライドされます。方向は、コントロール名、現在値のラベル、およびアニメーションチェックボックスのレイアウトにも影響します。

```
Explore( plot(magnitude * sin(scale * x + shift), x = -2 * π .. 2 * π, view = -1 .. 1),
  parameters = [ scale = 1.0 .. 5,
    [shift = -π .. π, placement = bottom, orientation = horizontal],
    [magnitude = -1 .. 1.0, placement = right ] ],
  placement = left, orientation = vertical, size = [300, 300] );
```



いくつかのコントロールが一緒に配置されている場合、これらは方向ごとにグループ化されます。たとえば、中央のプロットコンポーネントの左側に配置される場合、縦方向のスライダは左右に並べられ、上下並びとなる横方向のスライダの上に表示されます。

```
Explore( plot( magnitude * sin( scale * x +  $\frac{shift \cdot \pi}{2}$  ), x = -2 *  $\pi$  .. 2 *  $\pi$ , view = -1 .. 1 ),
  parameters = [ scale = 1.0 .. 5,
                [ shift = 0 .. 4, orientation = horizontal ],
                magnitude = -1 .. 1.0 ],
  placement = left, orientation = vertical );
```

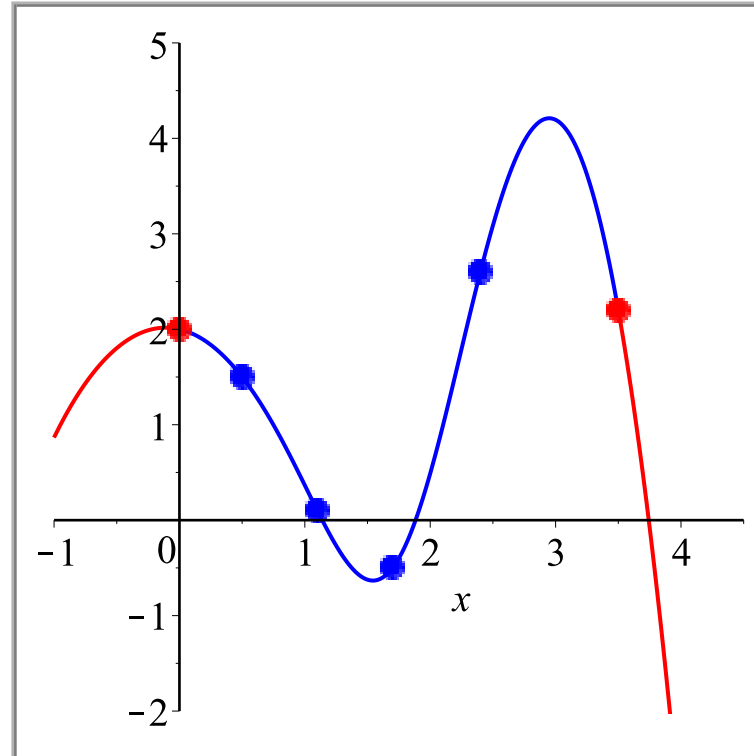



```
markers = [ [ xini[1], y1, color = red ],  
             seq( [xini[i], y||i ], i = 2..N - 1 ),  
             [ xini[N], y||N, color = red ], color = blue, symbol = solidcircle, symbolsize = 25 ],
```

```
initialvalues = [ seq( y||i = yini[i], i = 1..N ), n = 5 ],
```

```
placement = left, size = [500, 500], title = "Spline Interpolation" ) :
```

Spline Interpolation

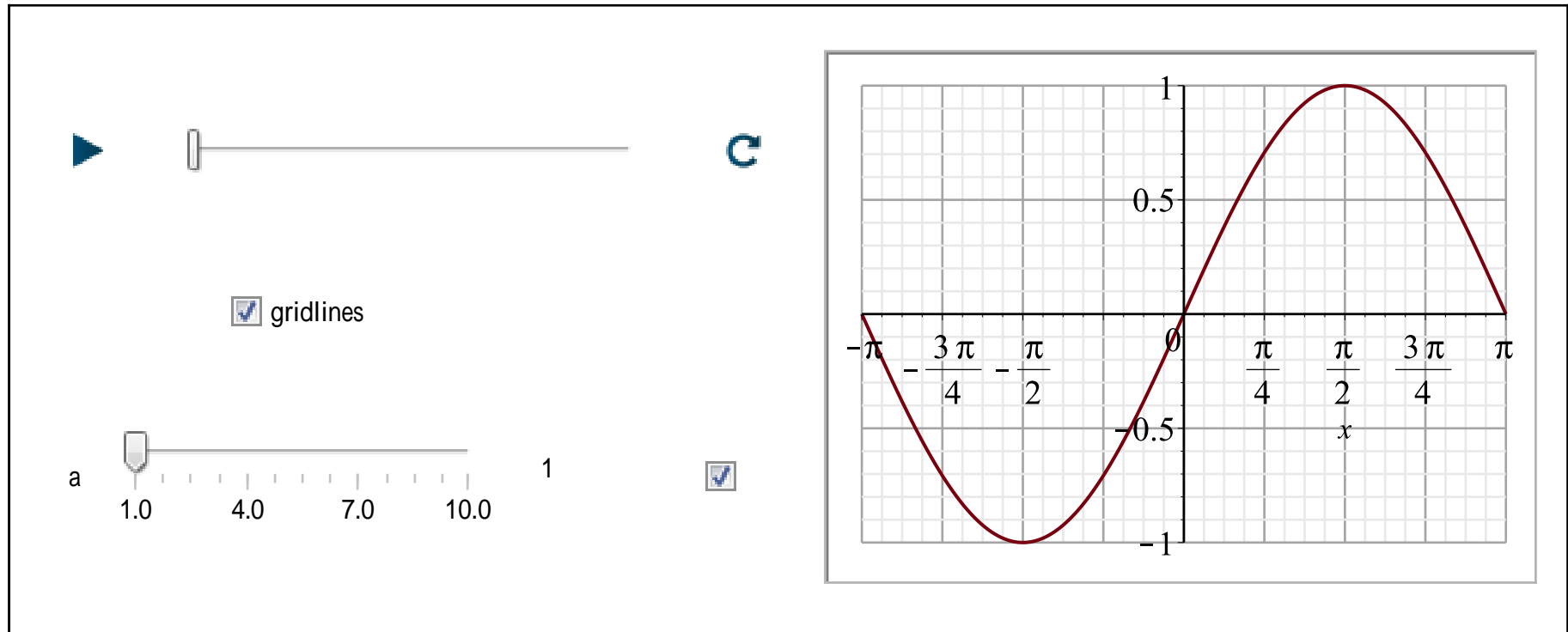


▼ CheckBox コントローラ

`true` および `false` の値となるパラメータとして、[CheckBox](#) コントローラを作成できるようになりました。これは、`true` および `false` エントリを含む 2 つの要素リストによって指定されたパラメータ値のデフォルトです。

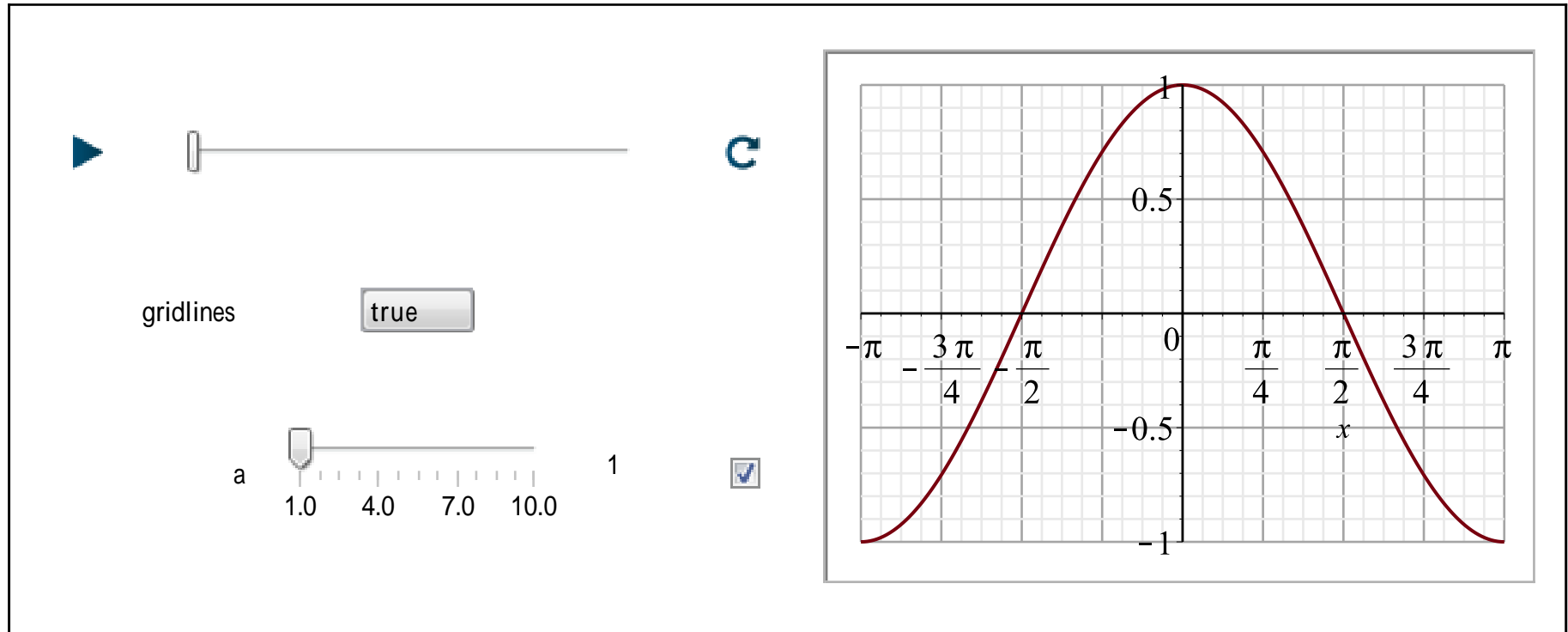
以下の例では、ラベル `gridlines` のパラメータ `tf` は `CheckBox` コンポーネントで制御されています。アニメーションの再生中でも `gridlines` のチェックボックスを選択または選択解除できます。

```
Explore( plot( sin( a * x ), x = -π..π, view = -1..1, gridlines = tf ),  
  parameters = [ [ tf = [ true, false ], label = gridlines ],  
    a = 1.0..10 ],  
  placement = left, size = [ 400, 300 ], loop, numframes = 20 )
```



パラメータ `tf` に `controller=combobox` を設定して同じ例を構築することはできますが、`CheckBox` コンポーネントのようにアニメーションの再生中に `gridlines` の値を変更することはできません。

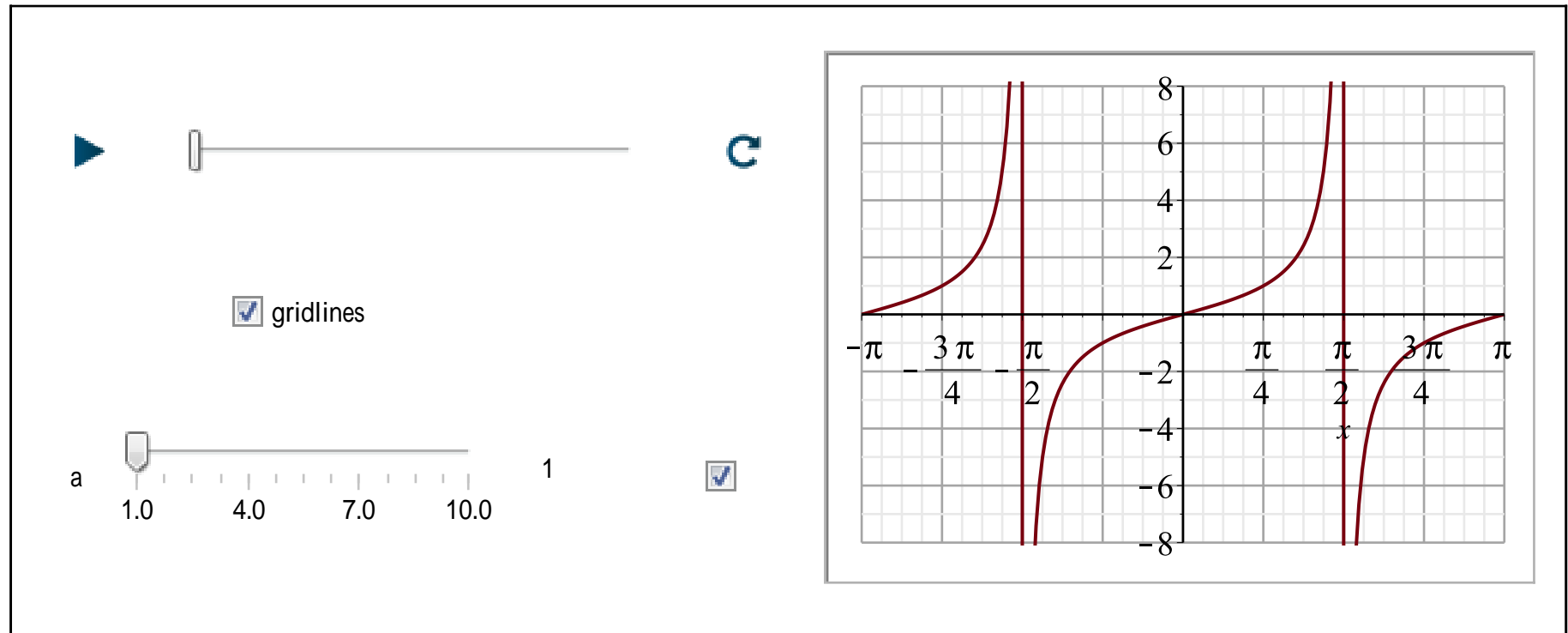
```
Explore( plot(cos(a * x), x=-π..π, view=-1..1, gridlines = tf),  
  parameters = [ [ tf = [true, false], label = gridlines, controller = combobox ],  
    a = 1.0..10 ],  
  placement = left, size = [400, 300], loop, numframes = 20 )
```



コントローラを `checkbox` に指定した場合は、値のリストを与える必要はありません。

```
Explore( plot(tan(a * x), x=-π..π, view=-8..8, gridlines = tf),  
  parameters = [ [ tf, label = gridlines, controller = checkbox ],  
    a = 1.0..10 ],
```

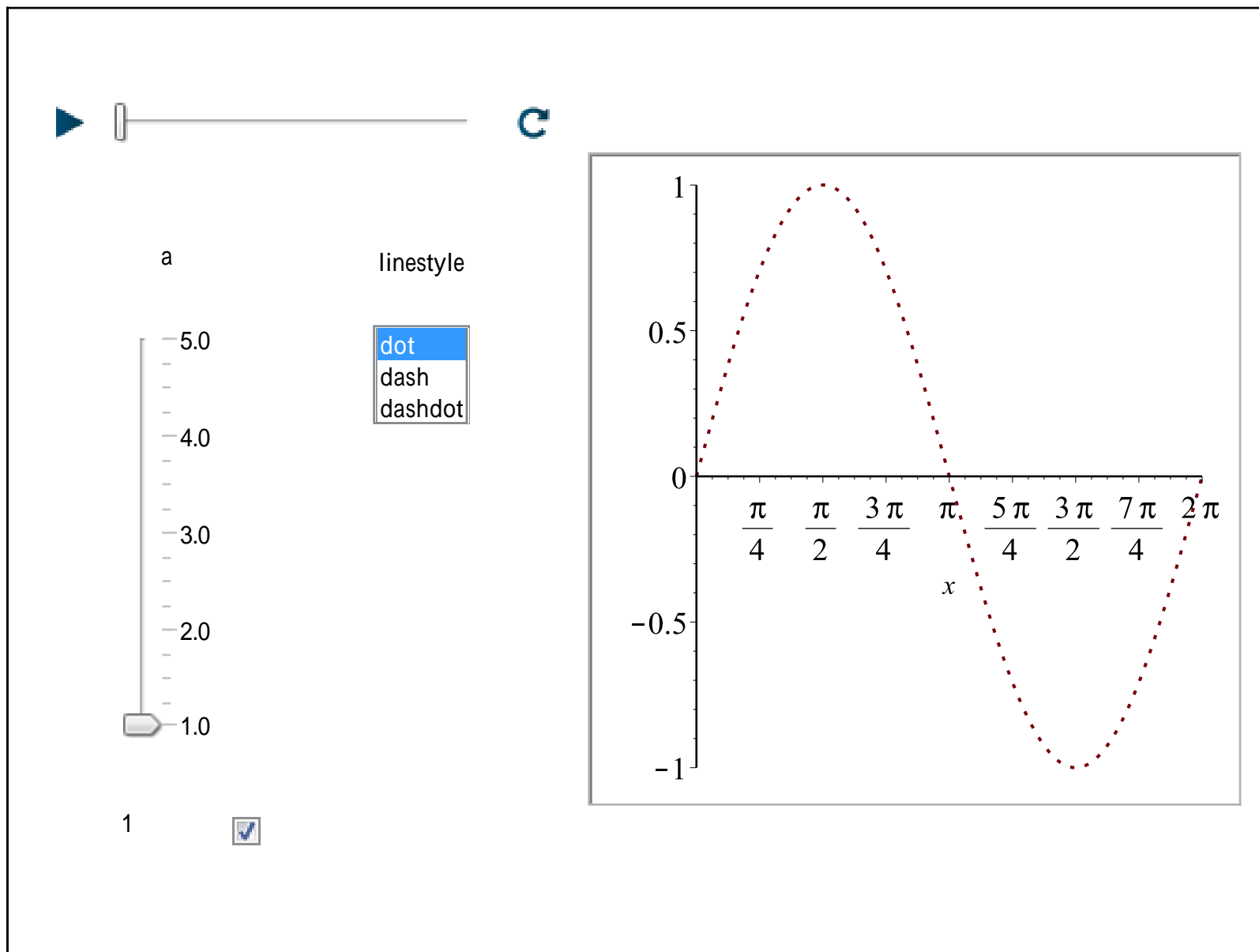
placement = left, size = [400, 300], loop, numframes = 20)



▼ ListBox コントローラ

特定のリストから値を取得するパラメータとして、[ListBox](#) コントローラを作成できるようになりました。

```
Explore( plot( sin( a * x ), x = 0 .. 2 * pi, linestyle = k ),  
  parameters = [ [ a = 1.0 .. 5.0, animate ],  
                [ k = [ dot, dash, dashdot ], controller = listbox, label = linestyle ] ],  
  orientation = vertical, placement = left, loop, size = [ 400, 400 ] );
```



▼ 新規ドキュメントにおけるスタートアップコードの定義

Explore の `newsheet` オプションによって、現在のドキュメントではなく新規ドキュメントでエクスプローラを開始することができます。これは、アプリケーションや代入文の作成時に便利です。旧バージョンの Maple では、Maple で定義されていない名前にエクスプローラが依存する場合 (たとえば自分で書いたプロシージャまたはモジュール)、これらの名前のコードを新規ドキュメントにコピーし、エクスプローラを実行する前にコードを実行する必要がありました。このプロセスは、新しい `startupdefinitions` オプションによって、簡単化されています。

`startupdefinitions` オプションでは、新規ドキュメントの [スタートアップコード](#) に代入文を挿入します。新規ドキュメントを開くと、これらの代入文が自動的に実行され、エクスプローラを開始できるようになります。以下のセクションでは、この操作方法を表す 2 つの例をあげます。最初の例は、新しいドキュメントのスタートアップコード領域にプロシージャを挿入する方法を示しています。2 番目の例は、スタートアップコード領域に未評価のモジュールを挿入する方法を示しています。

▼ プロシージャの挿入

以下の例では、Cobweb プロシージャによって、ロジスティック写像 $r \cdot x \cdot (1-x)$ のくもの巣プロットが生成されます。

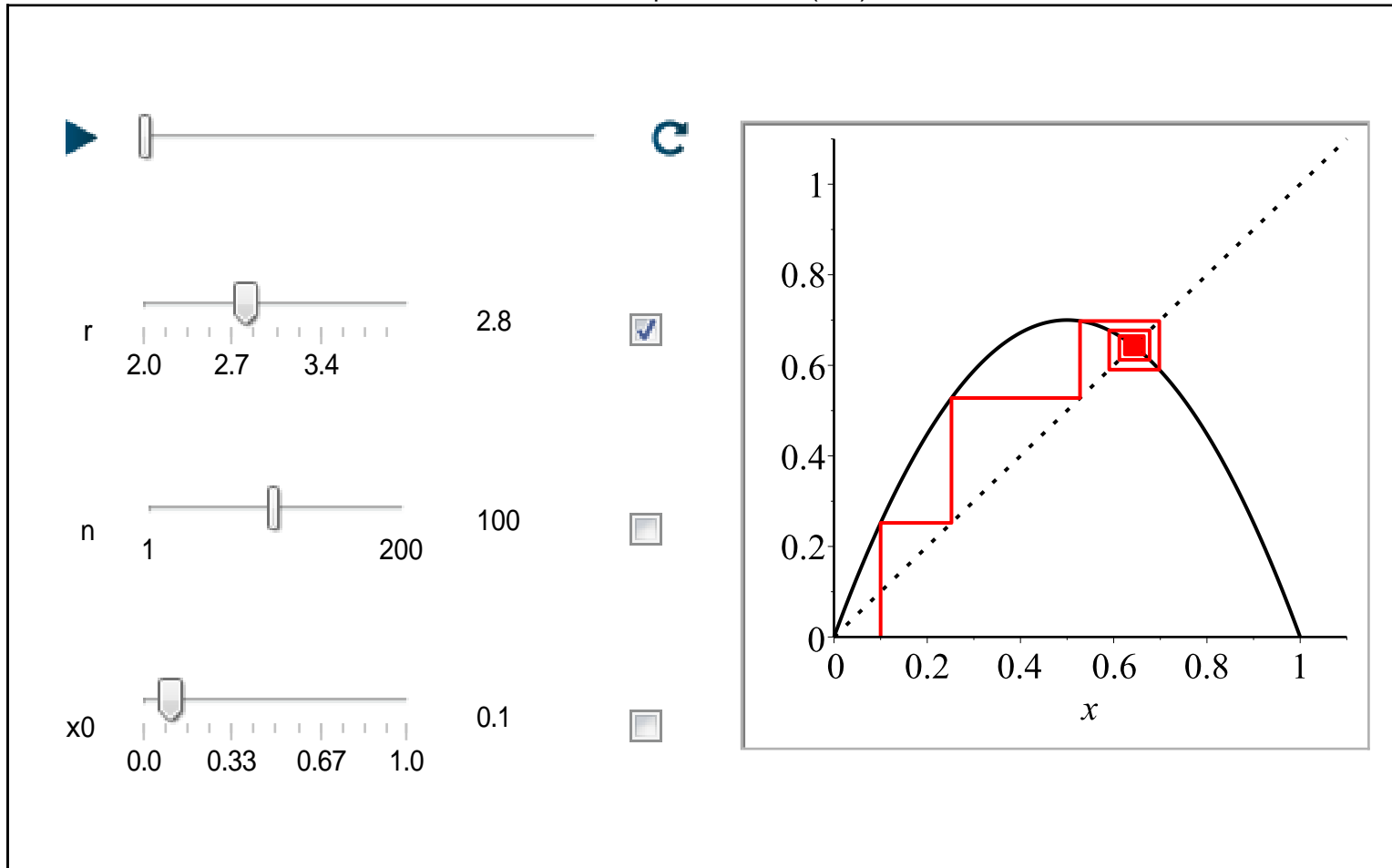
`newsheet` オプションによって、新規ドキュメントでエクスプローラが開始されます。`startupdefinitions` オプションでは、Cobweb プロシージャを Cobweb 名に割り当てるコードを、新規ドキュメントのスタートアップコード領域に挿入します。これにより、新規ドキュメントの起動後すぐにエクスプローラを開始することができます。

```
Cobweb := proc( r, x0, { numpoints :: posint := 1 } )
local f, i, M, N, x;
  N := 2·numpoints;
  f := (x, r) → r·x·(1-x);
  M := Matrix(N, 2, datatype = float[8]);
  M[1, 1], M[1, 2] := x0, 0;
  M[2, 1], M[2, 2] := M[1, 1], f(M[1, 1], r);
  for i from 3 to N-1 by 2 do
    M[i, 1], M[i, 2] := M[i-1, 1], f(M[i-1, 1], r);
    M[i+1, 1], M[i+1, 2] := M[i, 2], M[i, 2];
  end do;
plots:-display( plot(x, x = 0..1.1, color = black, linestyle = "dot"),
                plot('f'(x, r), x = 0..1, color = black),
                plot(M, style = line, color = red),
                view = [0..1.1, 0..1.1]);
end proc;
```

```
Explore( Cobweb( r, x0, numpoints = n ),
         parameters = [ r = 2.0..4.1, [ n = 1..200, animate = false ],
```

```
[x0 = 0.0..1.0, animate = false ] ,  
initialvalues = [ r = 2.8, x0 = 0.1, n = 100],  
animate, loop, placement = left,  
startupdefinitions = [ Cobweb],  
newsheet, showbanner = true, title = "Cobweb plot for r*x*(1-x)" );
```

Cobweb plot for $r \cdot x \cdot (1-x)$



以下の例で、現在のワークシートにおける名前 RGBOverlap に対する代入文には、モジュールの定義が含まれています。

この特定のモジュールには、ワークスペースとして動作する R というローカルメンバーがあります。この適用モジュールが最初に引数付きで呼び出されると、ワークスペースは初期化され、R には大きな配列が割り当てられます。これにより、モジュールは初回の呼び出しによって修正されます。

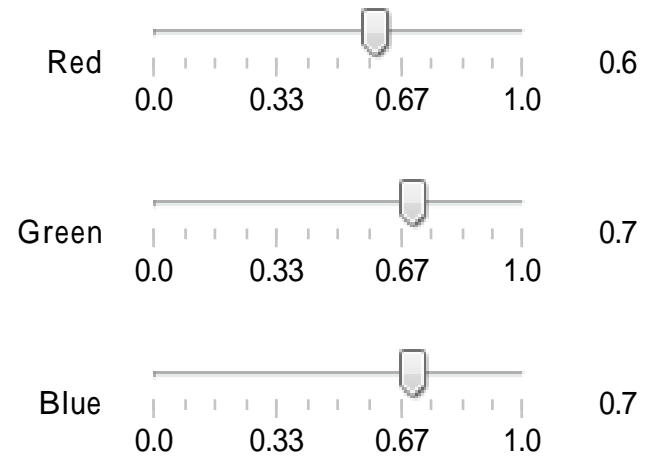
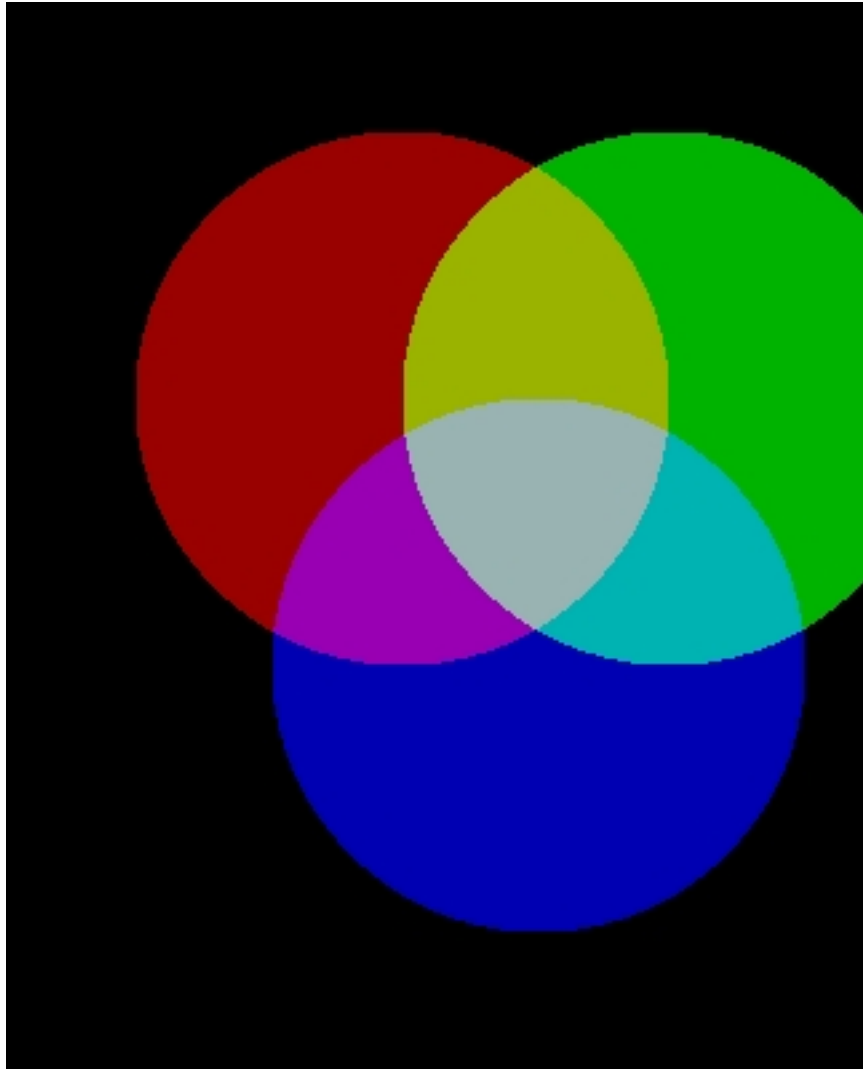
ただし、より小さな、未修正かつ初期化されていないモジュールを作成する、以下のオリジナルのステートメントは、エクスプローラで開かれる新規ドキュメントのスタートアップコード領域に引き続き配置することができます。これは、オリジナルの代入文の右辺を非評価引用符で囲むことで実現できます。以下の代入文で、右辺が一重右引用符 (') で囲まれている点と、Explore の `startupdefinitions` オプション内で名前 RGBOverlap を指定するために使用される後続の方程式の形式に注意してください。

```
RGBOverlap := module( )
local ModuleApply, init, f, sr, sg, sb, n, n1, n2, c, tc, R;
init := proc( )
  n := 200; n1, n2 := trunc(n/4), trunc(n/2);
  c := Matrix(n, 'datatype'='float[8]',
    subs(d2 = n2,
      (i, j) → `if`((d2-i)^2 + (d2-j)^2 < (d2)^2, 1, 0)));
  tc := Matrix(n, 'datatype'='float[8]');
  R := Array(1..2*n, 1..2*n, 1..3, 'datatype'='float[8]');
end proc;
f := proc(x)
  LinearAlgebra:-MatrixAdd(tc, c, .0, x, 'inplace');
  NULL;
end proc;
ModuleApply := proc(tr, tg, tb)
  if not n :: posint then init( ); end if;
  if sr ≠ tr then
    f(tr);
    sr, R[n1..n1 + n, n1..n1 + n, 1] := tr, tc;
  end if;
  if sg ≠ tg then
    f(tg);
    sg, R[n1..n1 + n, 2*n-5*n1..2*n-n1, 2] := tg, tc;
  end if;
  if sb ≠ tb then
    f(tb);
    sb, R[2*n-5*n1..2*n-n1, 1 + n2..n + n2, 3] := tb, tc;
```

```
end if;  
R;  
end proc;  
end module':
```

```
Explore( RGBoverlap( Red, Green, Blue ),  
  parameters = [ Red = 0.0 ..1, Green = 0.0 ..1, Blue = 0.0 ..1 ],  
  initialvalues = [ Red = .7, Green = .7, Blue = .7 ],  
  placement = right,  
  startupdefinitions = [ RGBoverlap = eval(RGBoverlap, 1) ],  
  newsheet, title = "The additive RGB color space" );
```

The additive RGB color space



▼ ドキュメントのプロパティ

新規ドキュメントでエクスプローラを開始するために `newsheet` オプションを使用するときは、`documentproperties` オプションによってドキュメントの [Document Properties](#) メタデータが設定されます。

新しいドキュメントの [ドキュメントのプロパティ] を表示するには、[ファイル] > [ドキュメントのプロパティ] を選択します。

たとえば、以下のコマンドによって新しいドキュメントでエクスプローラを開始できます。`documentproperties` オプションによって、新しいドキュメントの Author および Subject 属性のメタデータ値をそれぞれ「John Smith」と「Fractal explorations」に設定できます。

$$\text{Explore} \left(\text{Fractals:-EscapeTime:-Mandelbrot} \left(300, -1.78 + \frac{-1 - 1.5 I}{e^{\text{zoom}}}, -1.78 + \frac{2.2 + 1.5 I}{e^{\text{zoom}}}, \right. \right. \\ \left. \left. \text{iterationlimit} = a, \text{output} = \text{layer1} \right), \right)$$

`parameters = [a = 50..300, zoom = 0.0..8.0],`

`documentproperties = ["Author" = "John Smith", "Subject" = "Fractal explorations"],`

`newsheet, animate, placement = left`)



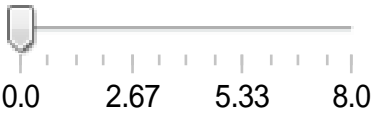
a



50



zoom



0

