

[Explore](#) コマンドを使用すると、対話型のアプリケーションやデモをすばやく簡単に作成することができます。任意の式やプロットを調べるためのさまざまな埋め込みコンポーネントを使用して、対話型な Explore を構築します。これらの自動構築されたコンポーネントを新しいワークシートに挿入し、簡単に保存したり共有したりすることができます。

Maple 17 で Explore コマンドが再実装され、機能の拡張と新機能の追加が行われました。最も大きな変更点は、次のとおりです。

- 調査コンポーネントを既存のワークシートまたはドキュメントに自動的に挿入できるようになりました。
- プログラミングで Explore コマンドを実行する時にパラメータを指定できるようになりました
- 以前に割り当てられた変数 (データを含む) を対象の式で参照できるようになりました。
- Explore コマンドを [plots:-interactive](#) コマンドで使用するようになりました。これには、[PlotBuilder](#) による使用も含まれます。

Explore コマンドで提供されているすべてのオプションの詳細は、[Explore](#) ヘルプページを参照してください。

注意 : 次の例を使用するには、このヘルプページをワークシートとして開き、ワークシートを実行してください。

## ▼ パラメータと初期値

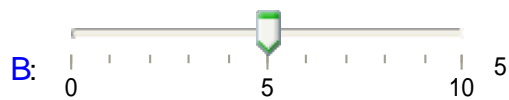
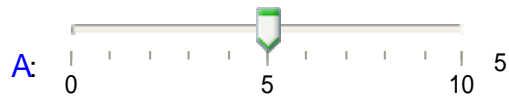
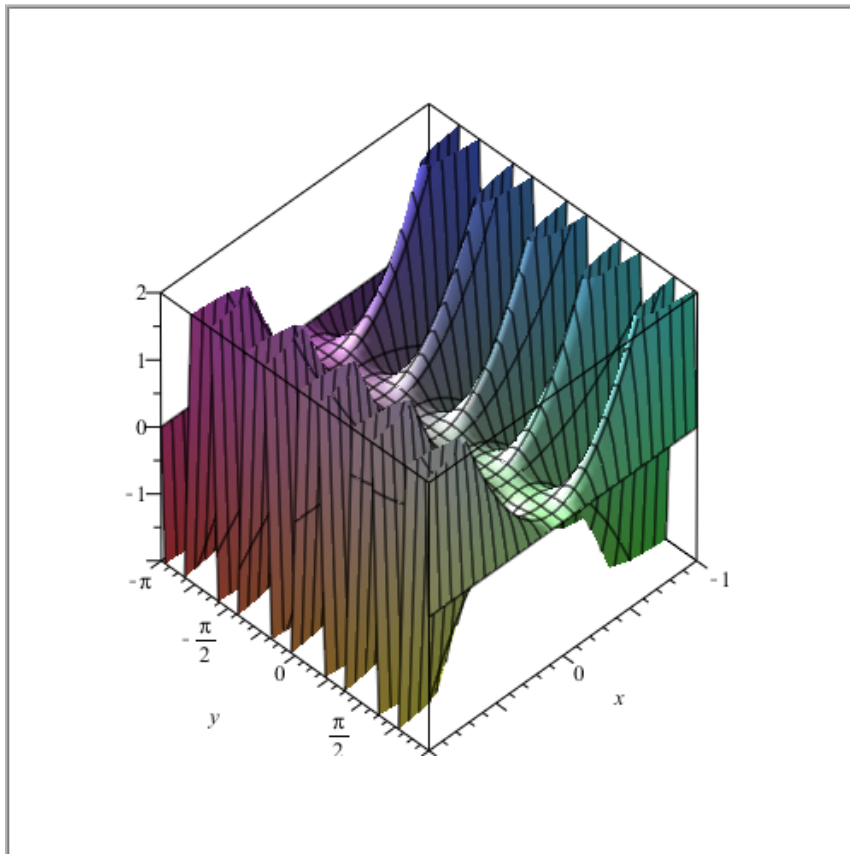
Explore コマンドに追加された新しい `parameters` オプションを使用して、プロットの式に含まれている未知の名前のどれをスライダーを使用して調べるかを指定できます。さらに、`initialvalues` オプションを使用して、最初の評価で使用するパラメータ値を指定することができます。

これら 2 つの新規オプションによって、Explore コマンドのプログラムでの使用がより実用的になりました。以下の 2 つのコマンドコールを使用して、効果を比較します。

最初の例では、ポップアップダイアログが表示され、パラメータ A と B の範囲が提案されます。Explore をクリックすると、提案された値が使用されます。

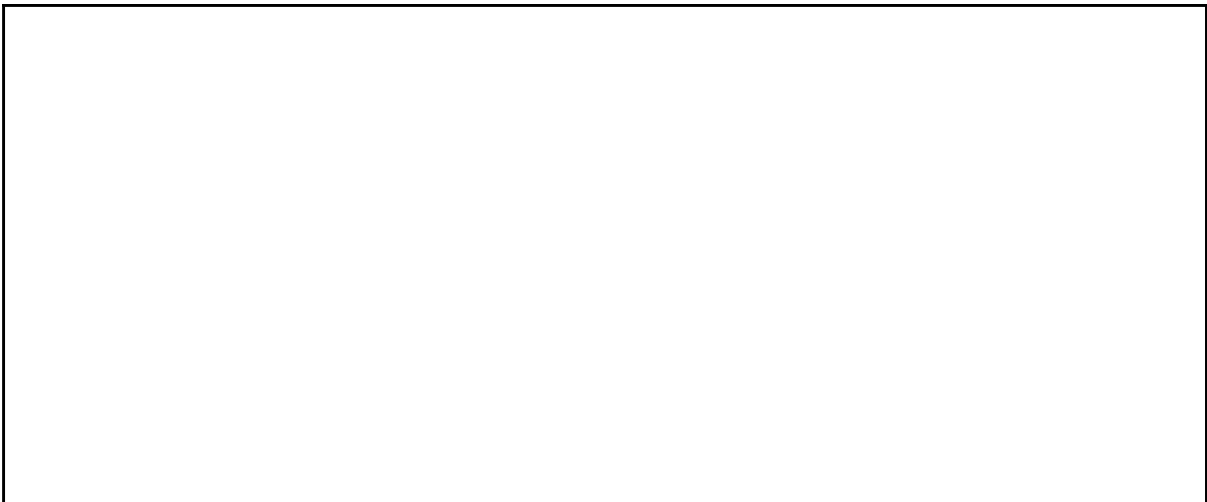
> `Explore(plot3d(sin(A y) B x^2, x = -1 .. 1, y = -pi .. pi, view = -2 .. 2))`

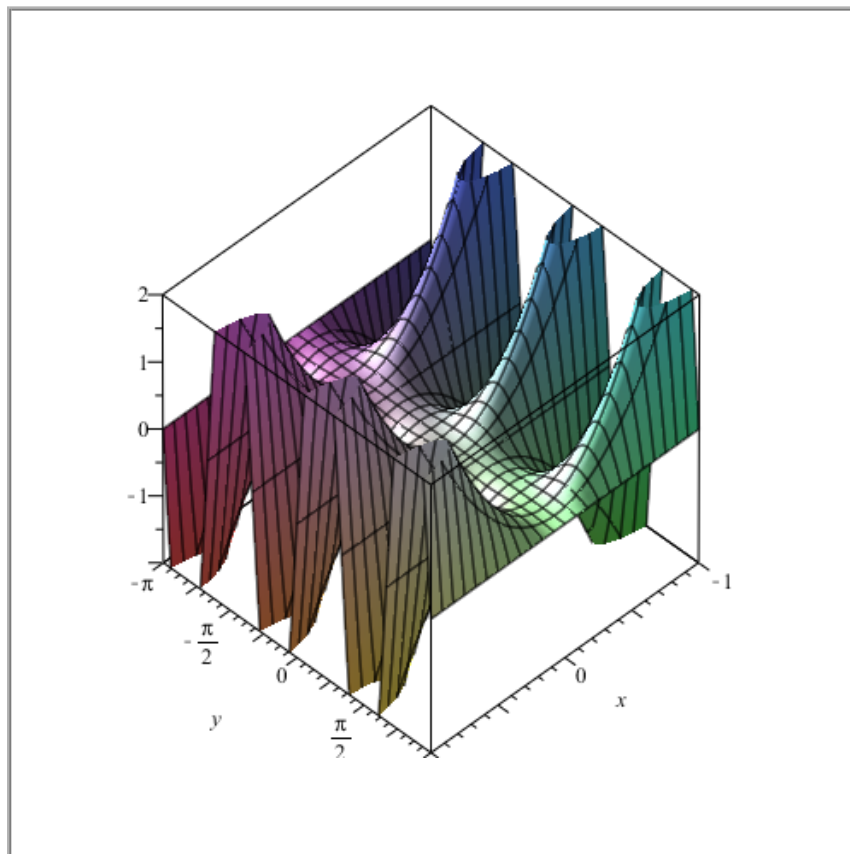




2 つ目の例では、ポップアップダイアログをスキップできます。

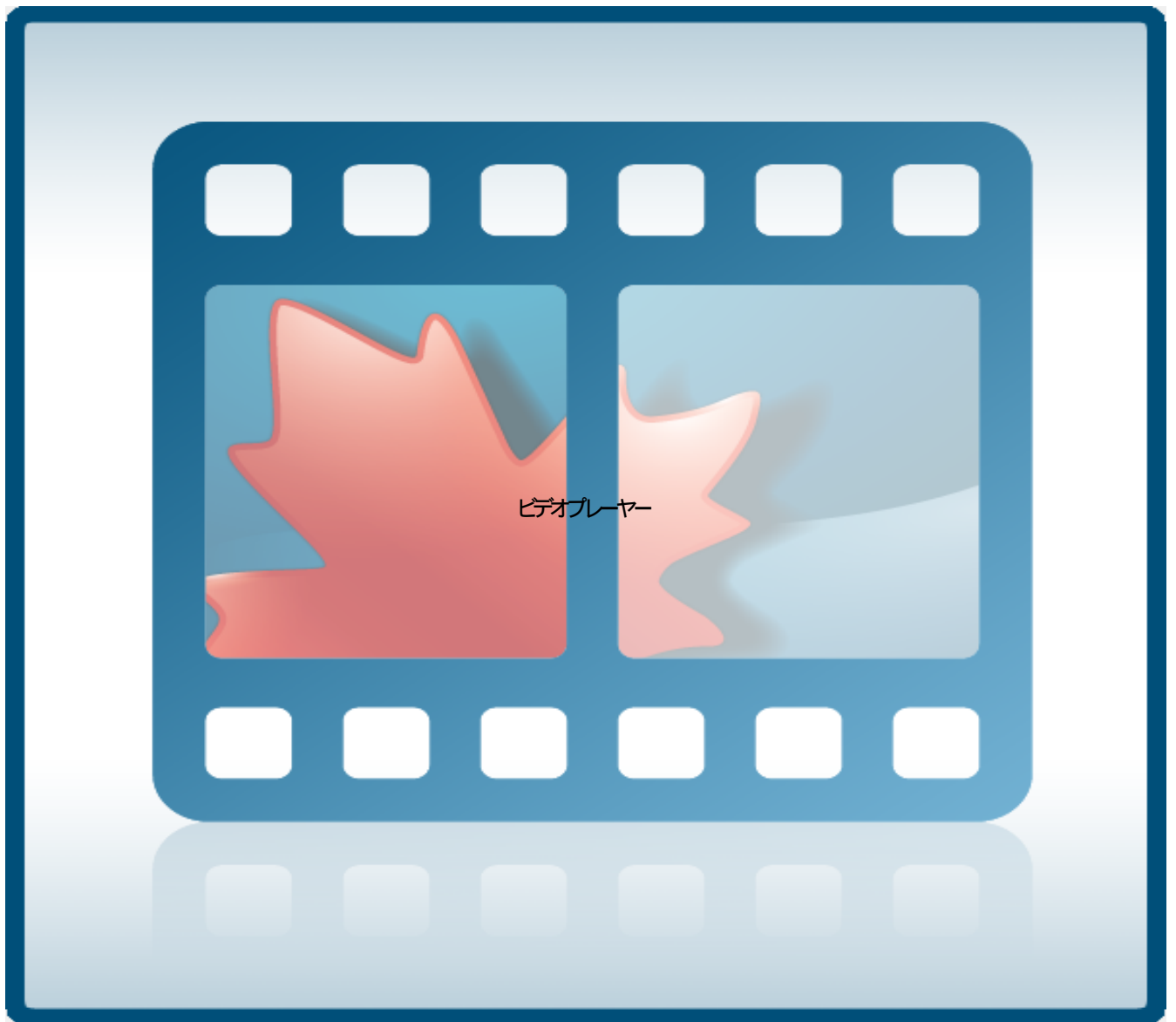
```
> Explore(plot3d(sin(A·y)·B·x^2, x = -1 .. 1, y = -π .. π, view = -2 .. 2),
  parameters = [A = 0 .. 10, B = 0 .. 10.0],
  initialvalues = [A = 3, B = 4])
```





## ▼ 動画による Explore の例

以下のビデオでは、前述のコマンドを使用して Explore の使用方法を動画で説明します。



## ▼ 高度な例

### ▼ 前回割り当てられたデータの使用

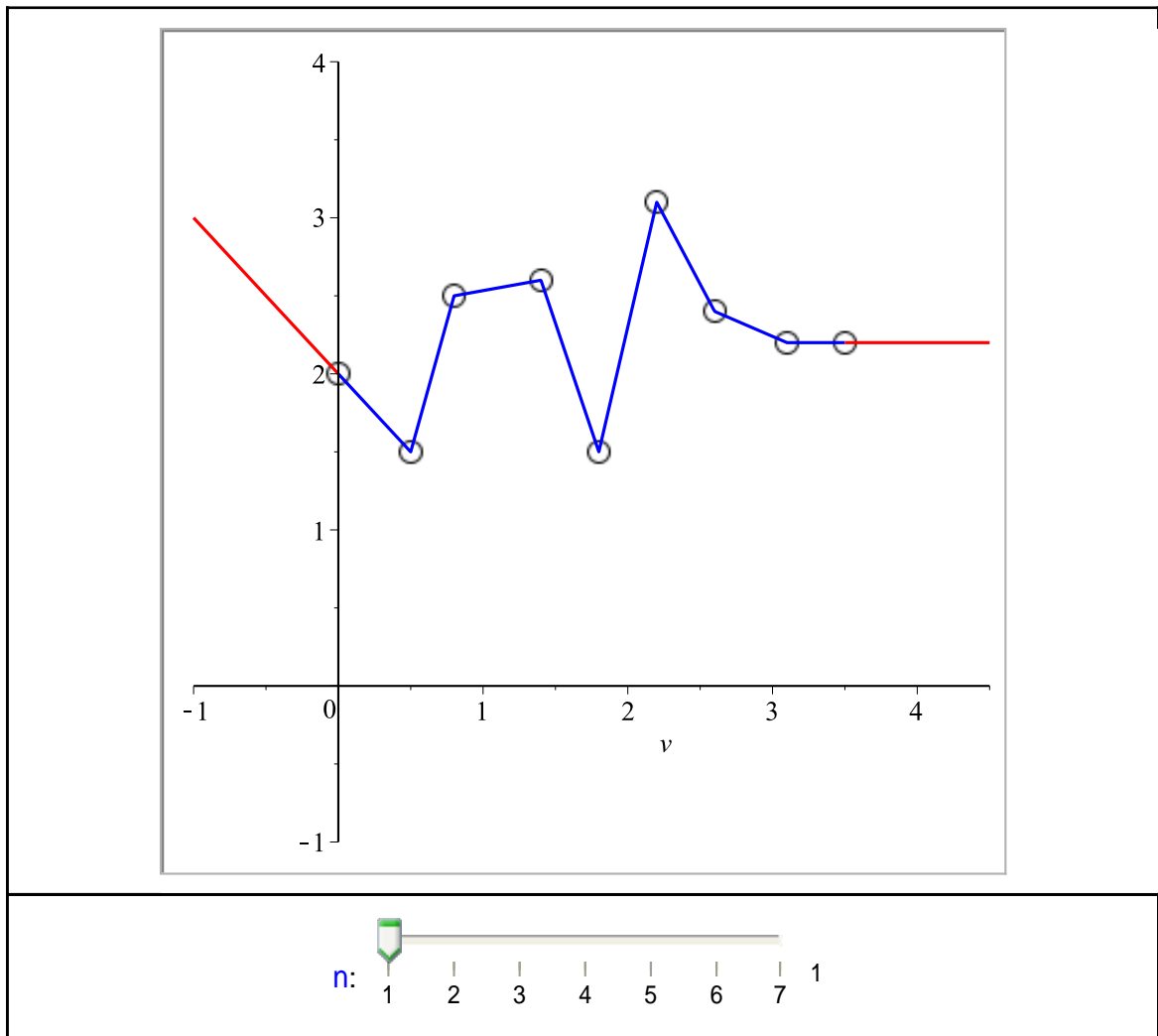
以下の例は、同じセッションのそれより前の段階で名前に割り当てられていたデータを使用する方法を説明しています。

- > `data := [ [0, 2], [  $\frac{1}{2}$ , 1.5 ], [0.8, 2.5], [1.4, 2.6], [1.8, 1.5], [2.2, 3.1], [2.6, 2.4], [3.1, 2.2], [3.5, 2.2] ] :`
- > `with(CurveFitting) :`
- > `Explore(plots:-display( plot([data, Spline(data, v, degree = n)], v = 0 .. 3.5, view = -1 .. 4, style = [point, line], color = [black, blue],`

```

        symbol = circle, symbolsize = 20),
        plot(Spline(data, v, degree = n), v = -1.0 .. 0., view = -1
        .. 4, color = red),
        plot(Spline(data, v, degree = n), v = 3.5 .. 4.5, view = -1
        .. 4, color = red)),
        parameters = [n = 1 .. 7])

```



## ▼ 関数コールによる Explore

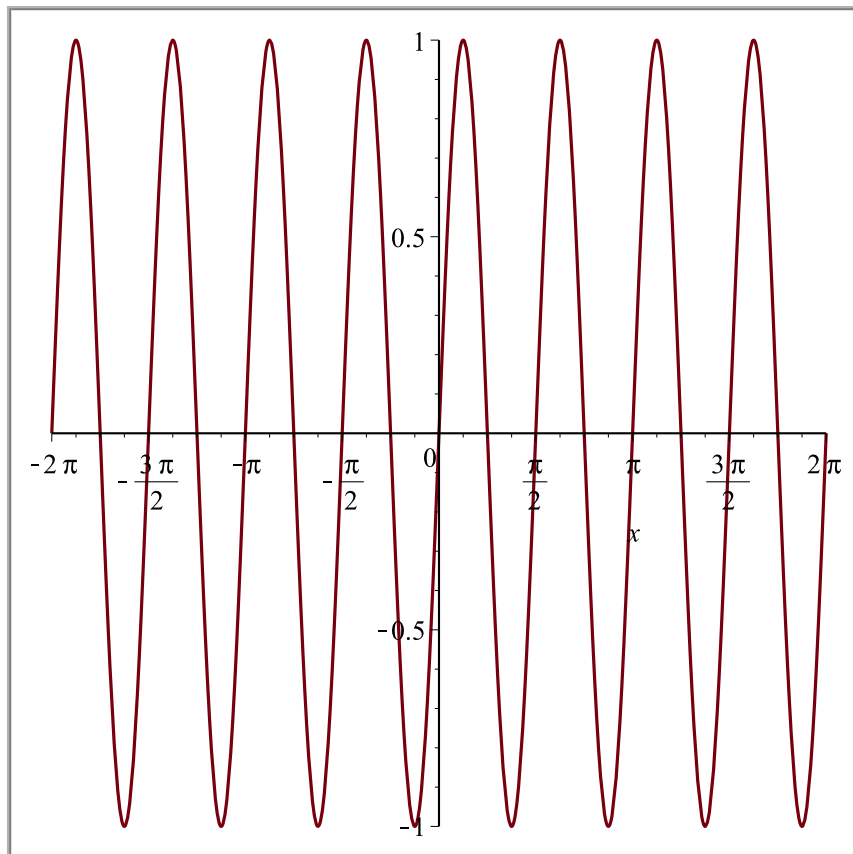
Explore コマンドの効果的な使用方法として、以前に定義された関数を Explore コマンドでコールすることができます。

ここで重要なことは、プロットなどの可視化されるオブジェクトをプロシージャが返す必要があるということです。

```

> f := proc(a)
    local b;
    b := a^2;
    return plot(sin(b * x), x = -2 * pi .. 2 * pi)
end proc;
> Explore(f(p), parameters = [p = -2.0 .. 2.0])

```



次のサブセクションは、より詳細な例でこの機能を説明しています。

## ▼ パラメータを使用する数値常微分方程式

値が割り当てられていない記号パラメータのある初期値問題 (IVP) があるとします。 [dsolve.numeric](#) の [interactive](#) 機能は、パラメータの複数の値で数值的に解を計算する効果的なメカニズムを提供します。

Explore コマンドを使用して、このような解空間とやりとりを行うことができます。

```
> sol := dsolve( {α·(diff(x(t), t)) = .7·(y(t)-x(t)) - sin(x(t))^2,
                α·(diff(y(t), t)) = .7·(x(t)-y(t)) + z(t),
                β·(diff(z(t), t)) = -y(t),
                x(0) = 3,
                y(0) = 0,
                z(0) = -3 },
                {x(t), y(t), z(t)},
                numeric, 'parameters' = [α, β], 'output' = 'listprocedure' ) :

> caller := proc(a, b) sol('parameters' = [α'=a, β'=b]); sol end proc :
```

```

> F := proc (A, B)
  plots:-odeplot(caller(A, B), [x(t), y(t), z(t)], t=0 .. 500,
    'style' = line, 'numpoints' = 10000, 'labels' = [x, y, z],
    'axes' = 'box', 'color' = ('COLOR') ('HUE', (1/10) * A))
end proc:

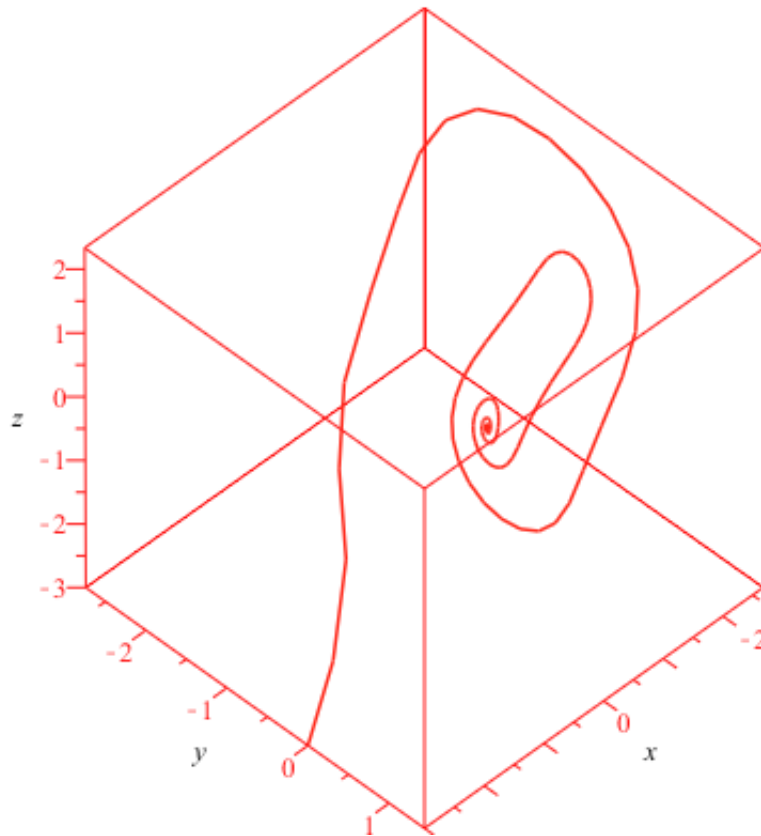
```

パラメトリック値  $\alpha=0.1$  および  $\beta=0.2$  における解のプロットを生成できるようになりました。

```

> F(0.1, 0.2)

```



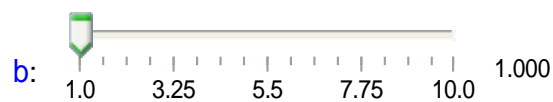
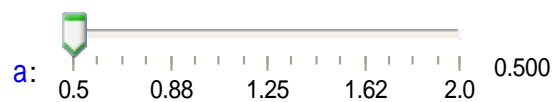
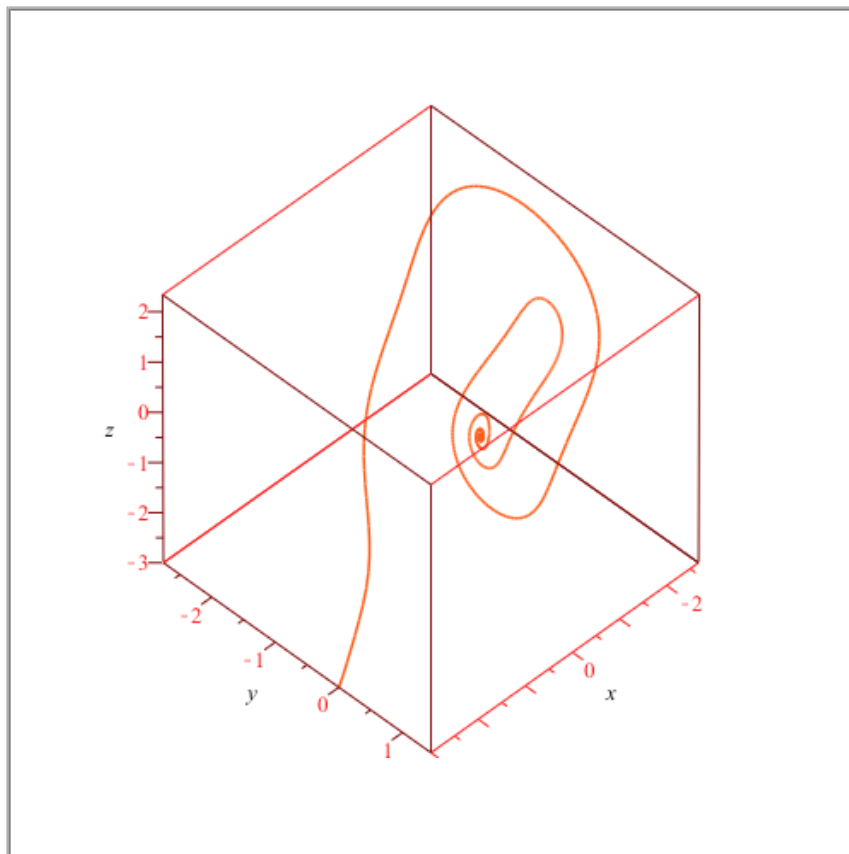
また、パラメータ値を変更しながら、このようなプロット群からの Explore 作成もできます。これは、Explore コマンドで関数をコールすることにより達成できます。

```

> Explore(F(a, b), 'parameters' = [a = 0.5 .. 2.0, b = 1.0 .. 10.0])

```





## ▼ アニメーションによる Explore

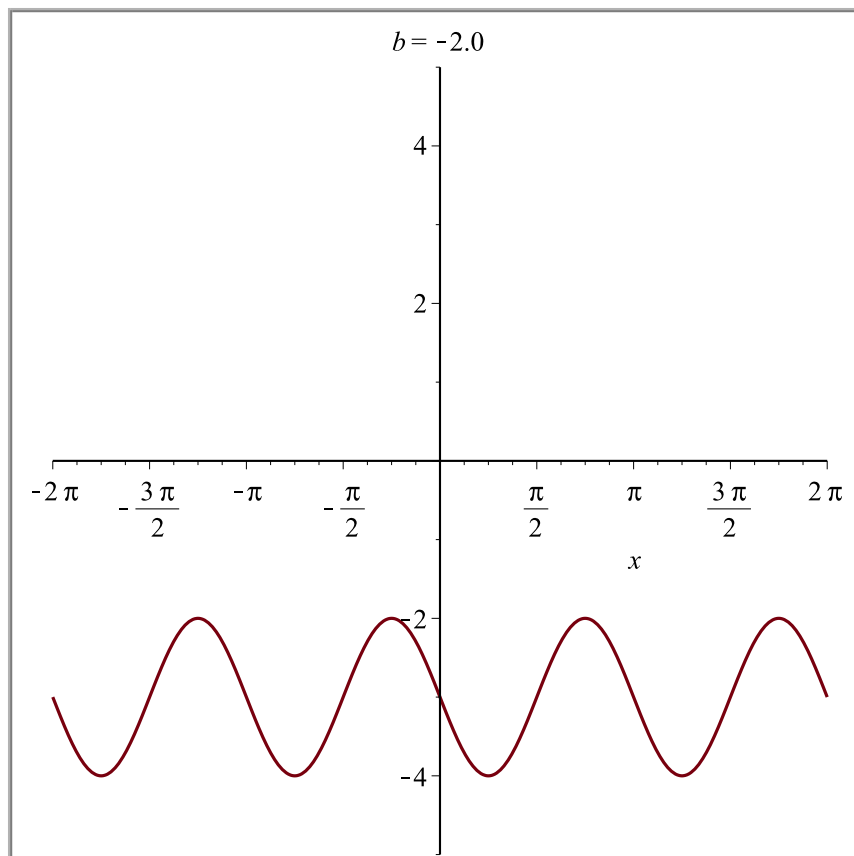
以下の Explore コマンドの実行後に、マウスでプロットを選択し、トップメニューからアニメーションを実行することができます。デフォルトでは、埋め込みコンポーネントはアニメーションを繰り返して再生します。アニメーションの再生中にスライダーを動かして、アニメーションのインスタンスの変化を確認できます。

スライダーを動かすたびに、新しいスライダー値に対するアニメーションがそのタイミングで作成されます。

```
> Explore(plots:-animate(plot, [a + sin(b x), x = -2 π..2 π, view = [-2 π..2 π, -5 ..5]], b
= -2.0 ..2.0),
parameters = [a = -3.0 ..3.0])
```







## ▼ 対話式プロット

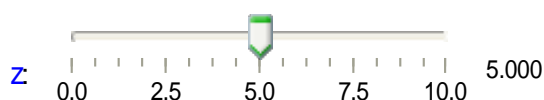
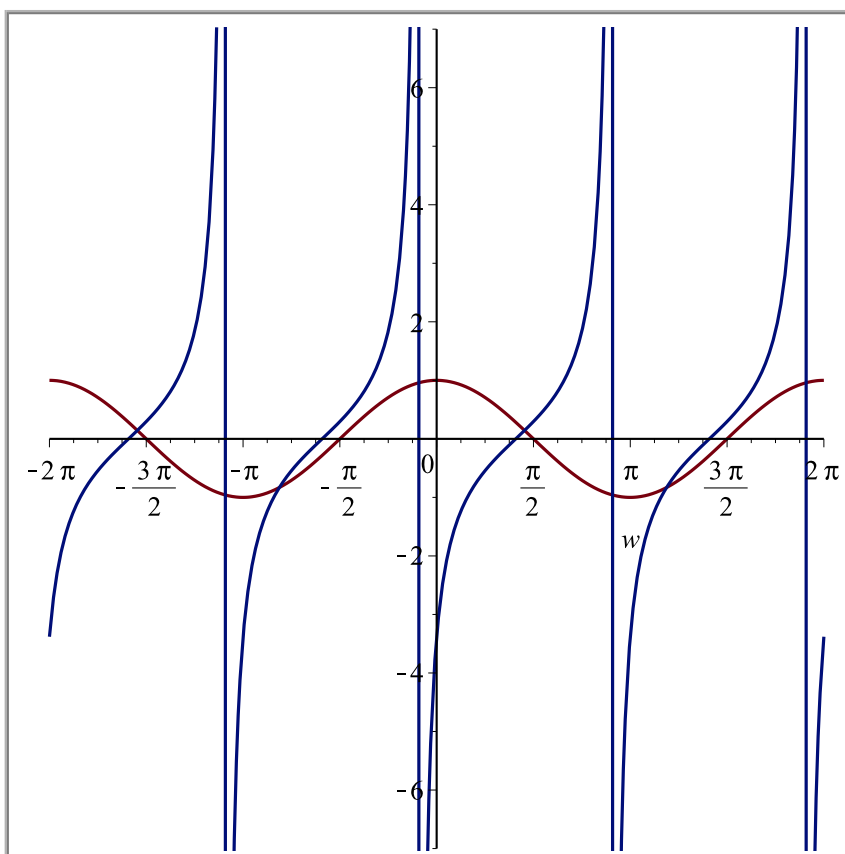
`plots:-interactive` コマンドによって生成される対話式プロットで、新しい Explore 機能を使用できるようになりました。

以下のコマンドを実行すると、ポップアップダイアログが生成されます。ポップアップダイアログの一番上にあるドロップダウンメニュー (コンボボックス) から [1 をパラメータとする対話的プロット] を選択し、[プロット] をクリックします。

結果は現在のワークシートに埋め込みコンポーネントとして表示されます。

> `plots:-interactive({cos(w), tan(w + z)}, variables = [w, z])`





## ▼ コンテキスト処理メニュー

plots:-interactive 機能は、[コンテキスト処理メニュー](#)を使用してクリックだけでアクセスすることもできます。コンテキスト処理メニューを使用すると、マウスだけで式に操作を適用できます。メニューは 2-D Math 出力 (実行後) および 2-D Math 入力 (実行前) の両方で使用できます。

最初に以下のコマンドを実行してから、出力された式を右クリック (Macintosh では Control を押しながらクリック) すると、関連するコンテキスト処理メニューが表示されます。ポップアップメニューから [プロット] オプションを選択します。[プロット] のサブメニューから、[プロットビルダー] を選択します。[\[プロットビルダー\]](#) ウィンドウが表示されます。一番上のドロップダウンメニューから [1 をパラメータとする対話的プロット] を選択し、[プロット] ボタンを押して終了します。

ワークシートの場合、上記の操作を実行すると、Explore コマンドの実行処理や埋め込みコンポーネントが挿入されます。ドキュメントの場合は、埋め込みコンポーネントは表示されますが、Explore コマンドのコールは直接には表示されません。

>  $e^x - ax^3$

## ▼ 簡単なアプリケーション作成

Explore コマンドは、Maple でアプリケーションを作成する最も速くて簡単な方法です。上述のいずれかの例をワークシートで実行して保存するだけで、アプリケーションを作成できます。

アプリケーションを少し綺麗にした場合のアドバイスは次のとおりです。

まず、Explore で埋め込みコンポーネントのテーブルが作成される場合、このテーブルは Maple で出力として扱われることに注意してください。Maple の他のコマンドと同じように、元の Explore コマンドコールを変更したり削除したりした場合、挿入されたコンポーネントの値も変更または削除されます。

挿入した埋め込みコンポーネントをより永続的な方法で保持するには、その埋め込みコンポーネントを含むテーブル全体をコピーして貼り付けます。現在のワークシートの別の場所にテーブルを貼り付けると、そこに含まれるコンポーネントは元の Explore コマンドとその出力テーブルが削除された場合でも元のとおり機能します。

これを試すには、以下の手順を実行します。

1. 上述のいずれかの例で、プロットとスライダーを選択してハイライト表示します。テーブル全体を選択する必要があります。プロットの前の領域を選択してから、マウスをスライダーコンポーネントの下領域までドラッグします。
2. 右クリックして [コピー] を選択します。
3. ワークシートの別の場所に貼り付けて、元の Explore コマンドと出力を削除します。
4. ワークシートの [スタートアップ](#) 領域に、非表示にしたいコードを移動します。
5. ワークシートを保存して、新しいスタンドアロンアプリケーションを作成します。

## 参照

[Explore、動的アプリケーションワークシート](#)