採点パッケージ

Maple 18 の新しい Grading パッケージには、対話型のクイズやアプリケーションを作成するための新しく改善されたツールが用意されています。Quiz コマンドによって、学生に質問してその回答を採点する、対話型の質問を Maple で生成することができます。これにより、概念を実践、調査、強化する機会を学生に簡単に与えることができます。

さらに、Grading パッケージには、対話型のアプリケーションを構築する際に使用できるロバストなコマンドツールキットがあります。たとえば、対話的にグラフ化して式 (任意の線形、2 次、絶対値関数など) で表すことを学生に求めるアプリケーションを作成したとします。グラフは、放物線の頂点や方程式の根などの特定の点を指定するなど、いくつかの方法で入力できます。これにより、GradePlot コマンドは、学生のグラフと実際の関数グラフを比較して、カスタマイズ可能な許容誤差でプロットを採点するために使用できるようになります。

```
> with(Grading):
```

クイズの生成

Quiz コマンドでは、質問を表示し、回答を受領して採点する、簡単な対話型ツールを生成します。

クイズの最も簡単な形式では、質問と回答の 2 つの引数を取ります。

```
> Quiz("What is 1 + 1?", 2)
```

What is 1 + 1?

Check Answer

選択問題や正誤問題を生成することもできます。

```
> Quiz("What is 1 + 1?", 3, [0, 1, 2, 3, 4], 'style'='multiplechoice')
```

What is 1 + 1?

Check Answer

- [ ] 4
- [ ] 3
- [ ] 1
- [ ] 2
- [ ] 0

アルゴリズム問題を生成できます。

```
> Quiz("Is the following number prime?", true,
        proc( ) nextprime(rand(2..20)( )) end,
        'style'=truefalse )
```

Is the following number prime?

Check Answer

Try Another

19

- [ ] True
- [ ] False

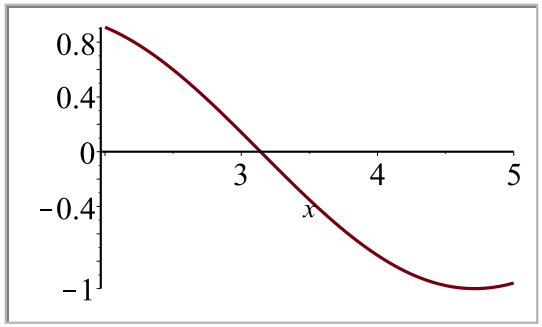
プロット問題を採点することもできます。

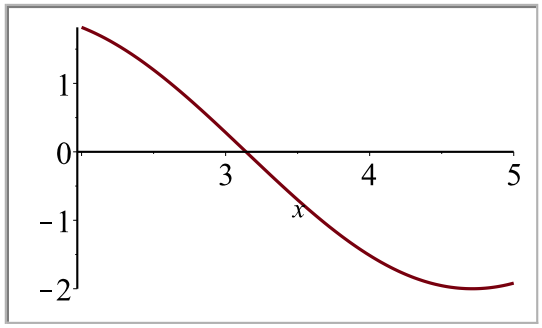
```
> Quiz("Which of the following could be portions of the plot of sin(x)?", {1, 2},
```

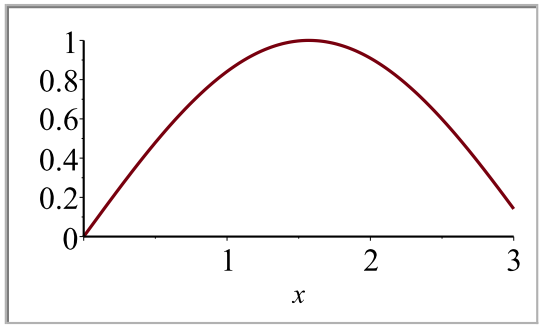
```
[plot(sin(x), x = 0 ..rand(2 ..3)( )),
 plot(sin(x), x = rand(2 ..3)( ) ..5),
 plot(2· sin(x), x = rand(2 ..3)( ) ..rand(4 ..7)( ) )],
   style = multipleselect,
   gradebyindex = true,
   size = [250, 150])
```

Which of the following could be portions of the plot of sin(x)?

Check Answer







クイズの例題ワークシートにその他の例を挙げています。

プロットされた関数の採点

GradePlot コマンドでは、簡単な数学関数の表現を構築し、採点することができます。

以下の例では、QuadraticFunction コンストラクタを使用し、2 つのうちのいずれかの方法で 2 次関数を表すオブジェクトを作成することができます。

- 1 つの式がある場合
- または、頂点を表す点と関数上の別の点の、2 つの点がある場合

教員によって設定された解

```
> correctquadratic := QuadraticFunction(x^2-4*x+4);
```

correctquadratic := << QuadraticFunction: x^2-4*x+4 >> (2.1

正解

```
> answer1 := QuadraticFunction([2, 0], [0, 4]);
```

answer1 := << QuadraticFunction: v^2-4*v+4 >> (2.2

小さな誤差が許容される場合は、正解です。プロットされた関数と関連して、いくらかの誤差がある点が提供される可能性が高いです。

```
> answer2 := QuadraticFunction([2.0001, -0.0023], [0.0126, 4.0108]);
```

answer2 := << QuadraticFunction: 1.015934496*v^2-4.063941171*v+4.061844369 >> (2.3

不正解

```
> answer3 := QuadraticFunction([3, -0], [0, 4]);
```

answer3 := << QuadraticFunction: 4/9*v^2-8/3*v+4 >> (2.4

GradePlot コマンドでは、希望の解と提供された回答を比較し、採点結果として 0 か 1 を返します。採点はプロットと関連しているため、プロット範囲を指定する必要があります。厳しい許容誤差が指定されている場合、2 つ目の回答は不正解となる点に注意してください。

```
> pview := [-10..10, -10..10]:
> GradePlot(correctquadratic, answer1, pview);
```

1. (2.5

```
> GradePlot(correctquadratic, answer2, pview);
```

1. (2.6

```
> GradePlot(correctquadratic, answer2, pview, 'tolerance'= 0.0001);
```

$$0. \tag{2.7}$$

```
> GradePlot(correctquadratic, answer3, pview);
```

$$0. \tag{2.8}$$

構築された関数での作業

Grading パッケージのコンストラクタを使用して作成した関数について、さらに詳細を得るために利用できるコマンドがいくつかあります。以下にいくつかの例を示します。

```
> a1 := AbsoluteValueFunction(2 * abs(x - 3) + 1);
```

$$a1 := \mathit{<<\ AbsoluteValueFunction:\ 2*abs(x-3)+1\ >>} \tag{3.1}$$

```
> a2 := AbsoluteValueFunction([2.5, 0.8], [2, 2]);
```

$$a2 := \mathit{<<\ AbsoluteValueFunction:\ 2.400000000*abs(v-2.5)+.8\ >>} \tag{3.2}$$

オブジェクトに関連した式を得るには、GetExpression コマンドを使用できます。

```
> GetExpression(a1), GetExpression(a2);
```

$$2\,|x-3|+1,\ x,\ 2.400000000\,|v-2.5|+0.8,\ v \tag{3.3}$$

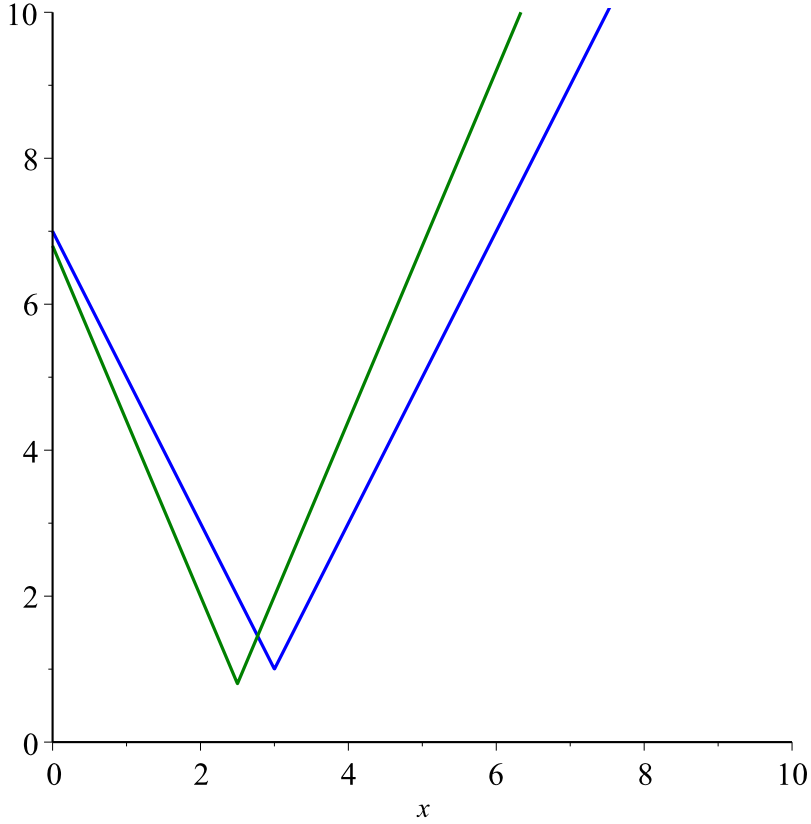
オブジェクトを構築するために使用された元のデータを得るには、GetData コマンドを使用します。

```
> GetData(a2);
```

$$[\ \mathit{<<\ GridPoint:\ [2.5,\ .8]\ >>},\ \mathit{<<\ GridPoint:\ [2,\ 2]\ >>}\ ] \tag{3.4}$$

オブジェクトを描画するには、Draw コマンドを使用します。

```
> plots:-display(Draw(a1, [0..10, 0..10], 'color'= "Blue"), Draw(a2, [0..10, 0..10], 'color'= "Green"));
```



採点用アプリケーションの作成

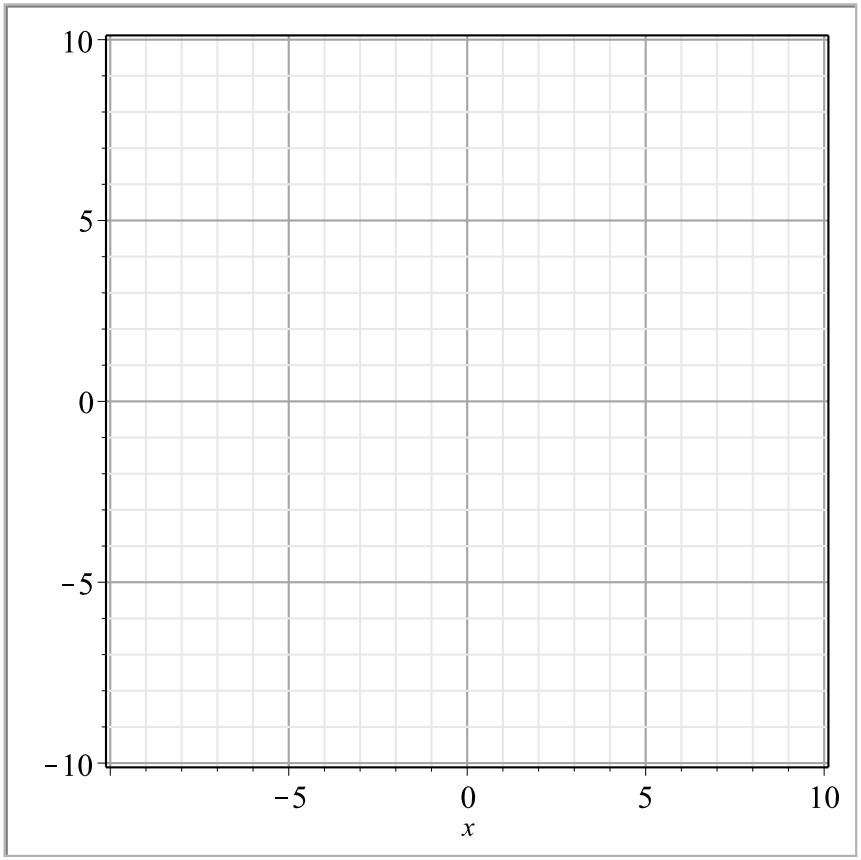
採点用アプリケーションを作成するために、埋め込みコンポーネントで Grading パッケージを効果的に使用することができます。通常、以下のコンポーネントは 1 つのドキュメント内に配置され、コードはスタートアップコード領域に配置されますが、ここでは Grading コマンドの簡単な使用方法を示すために表示されています。

以下のコンポーネントを使用する前に、以下のコードエディタでコードを実行してください。以下の例では、ユーザは 2 つの点をクリックすることで与えられた線を描画することができます。

DrawLine - MathApp では、このアプリケーションのより完全なバージョンを利用できます。

```
Actions := module()
    export Initialize, Click, Reset, Grade;
    local DoPlot, ref, userline, userpts, x, plotview, npts;
    uses DocumentTools, Grading;

    Initialize := proc()
        ref := LinearFunction(2*x-5);
        plotview := [-10..10, -10..10];
        userpts := Array(1 .. 2, i->NULL);
        Reset();
    end proc;

    Reset := proc()
        userline := NULL;
        userpts[1] := NULL;
        userpts[2] := NULL;
        npts := 0;
        DoPlot();
        SetProperty("TextArea0", 'value', "");
    end proc;

    Click := proc()
        local w;
        w := GridPoint(map2(GetProperty, "Plot0", ['startx', 'starty']));
        if npts<2 then
            npts := npts+1;
            userpts[npts] := w;
        end if;
        userline := `if`(npts<2, NULL, LinearFunction(userpts[1], userpts[2],
            'variable'=x));
        DoPlot();
    end proc;

    DoPlot := proc({drawsolution := false})
        local plts;
```

Draw the graph of $2 \cdot x - 5$



Grade Reset

▶ Pages That Link to This Page