Correcting a math problem

AcroTeX.Net
September 5, 2017

D. P. Story

dpstory@acrotex.net
Version 1.0

1. Introduction

The objective-style math problem that uses \RespBoxMath has been around throughout the history of exerquiz. Normally, the student is presented with a math problem and a *empty* input box into which he/she enters a response. However, you can pose a question with the input text field already populated with text; the student is then expected to edit the text to obtain the correct answer. A small, but interesting variation.

For single variable questions, as in question 1 of both the immediate feedback (shortquiz env.) and delayed feedback (quiz env.) quizzes demonstrate. What is more interesting and challenging is to pose a problem containing multi-letter variables with alternate appearances; these alternate appearances may be initially visible (as in problem 2) or may not be initially visible (as in problem 3).

This file demonstrates a new type (a variation on an old type, actually) of a math fill-in problem. The \RespBoxMath command is used but the field is given an initial value. The initial value given is an *incorrect answer* and the problem for the student is *to edit the field* and *to correct the answer* given.

Quiz Correct the following expressions by editing the input field. Passing is to make no more than 2 errors total. Use 'theta' for θ in some of the questions below.

1. $\dfrac{d}{dx} c\cos(2x^2) =$ c sin(4x) □

We demonstrate how to set up this kind of problem *with* multi-letter variables and initial alternate appearances.

2. $\dfrac{d}{d\theta}(\theta^2\sin(\theta)) =$ -θ^2 cos(θ) + 2 sin(θ) □

Now the same problem but *without* initial alternate appearances.

3. $\dfrac{d}{d\theta}(\theta^2\sin(\theta)) =$ -theta^2 cos(theta) + 2 sin(theta) □

Clear 0

Now the same set of questions in a quiz environment.

Begin Correct each by editing the input text fields. Where applicable, use 'theta' for the variable θ.

1. $\dfrac{d}{dx} c\cos(2x^2) =$ c sin(4x)

We demonstrate how to set up this kind of problem *with* initial alternate appearances.

Answer:

2. $\dfrac{d}{d\theta}(\theta^2 \sin(\theta)) =$ -θ^2 cos(θ) + 2 sin(θ)

Now the same problem *without* initial alternate appearances.

3. $\dfrac{d}{d\theta}(\theta^2 \sin(\theta)) =$ -theta^2 cos(theta) + 2 sin(theta)

End Score: Correct

Answer:

2. Some technical details and comments

Initializing single-variable or multi-variable problems that have no alternate appearances is not a problem, see the source file for problems 1 and 3; use `\V` and `\DV` to assign their initial values.

When the variables have alternate appearances and you want these alternate appearances to be initially visible, then it becomes a little tricky. Two methods have been developed. We use problem 2 to illustrate the methods.

Method 1 This method uses a special formatting command `\formatInitAltApprs` that is passed as the value of the key `\AddAAFormat` within the optional arguments of `\RespBoxMath`, like so,

```
\RespBoxMath[\rectW{2.5in}\AddAAFormat{\formatInitAltApprs}
  \DV{\unicodeStr(AltApprs)}\V{\unicodeStr(AltApprs)}]
  {theta^2 cos(theta)+2theta sin(theta)}(\rpl{theta->\utheta}{x})
  {3}{.0001}{[0,1]}
```

In the first line above, we have `\AddAAFormat{\formatInitAltApprs}`, which signals to apply the alternate appearances initially. Use `\formatInitAltApprs` only in the context of having alternate appearances (the '`->`' notation is used within the first argument of `\rpl`, as seen above). This method is used in problem 2 in the `shortquiz` environment above.

Method 2 The second method, that is used in problem 2 in the `quiz` environment uses an *external mechanism.*

```
\bInitAltAppr\RespBoxMath[\rectW{2.5in}\DV{\unicodeStr(AltApprs)}
  \V{\unicodeStr(AltApprs)}]
  {theta^2 cos(theta)+2theta sin(theta)}(\rpl{theta->\utheta}{x})
  {3}{.0001}{[0,1]}\eInitAltAppr
```

Notice that `\RespBoxMath` and its arguments are enclosed in the `\bInitAltAppr`/ `\eInitAltAppr` command pair. `\bInitAltAppr` basically defines the necessary code for setting the initial alternate appearances through the `\AddAAFormat` key (as well as other things), while `\eInitAltAppr` reverses the 'damage' done by `\bInitAltAppr`.

There are a couple of advances to this method: (1) You can enclose more than one `\RespBoxMath` problems all of which have initial alternate appearances with a single pair of `\bInitAltAppr`/`\eInitAltAppr` commands; (2) the `\addToAction` command can be effectively used to add in code.

```
\bInitAltAppr
\addToAction*{\AddAAFormat}{console.println("Begin init vars";\r}
\addToAction{\AddAAFormat}{\r console.println("Done init vars";}
\RespBoxMath[\rectW{2.5in}\DV{\unicodeStr(AltApprs)}
  \V{\unicodeStr(AltApprs)}]
  {theta^2 cos(theta)+2theta sin(theta)}(\rpl{theta->\utheta}{x})
  {3}{.0001}{[0,1]}
\eInitAltAppr
```

`\addToAction*` adds code prior to the already defined `\AddAAFormat` code, while `\addToAction` appends its code to the `\AddAAFormat` code.