Introduction to WeBWorK

The easiest way to get started with WeBWorK is to just start authoring problems. I included some links at the end of this document with sample problems and more specific problem techniques. Because WeBWorK is an open source platform you can get it to do almost anything you want. This document contains some basic tips on how to get started, and some specific techniques that I found useful.

INTRODUCTION: 2

MANAGING PROBLEM SETS 2

CHANGING THE PROBLEM SET AND COURSE INFO 3

STUDENT SPECIFIC CHANGES 3

CREATING QUIZZES 3

BASIC SYNTAX 4

- GENERIC PROBLEM 4
- MACROS 4
- DESCRIPTIONS, TAGGING, AND COMMENTS 5
- VARIABLES AND ARRAYS 5
- RANDOMIZING 6
- FOR AND IF STATEMENTS : 6
- DISPLAYING TEXT 7
- SOLUTIONS 8

MATH OBJECTS 9

ANSWER EVALUATORS 10

PROBLEM TECHNIQUES 10

- STRINGS IN STUDENT ANSWERS 10
- TOLERANCE IN NUMERICAL ANSWERS: 11
- LISTS AS ANSWERS: 12
- SHOWING CORRECT ANSWERS: 12
- ANSWER HINTS 13
- HINTS WITH PENALTIES 13

PROBLEM TEMPLATES 15

- TAYLOR POLYNOMIALS 15
- NEWTON'S METHOD 19
- CURVE SKETCHING 21
- INVENTORY CONTROL 1 25
- INVENTORY CONTROL 2 29
- LOANS AND MORTGAGES 1 33
- LOANS AND MORTGAGES 2: MOST EXPENSIVE HOUSE POSSIBLE 37
- DEFINITION OF A DERIVATIVE 1 41
- DEFINITION OF A DERIVATIVE 2 43

INDEX OF ECONOMIC QUESTIONS 45

HELPFUL LINKS 45

Introduction:

(from http://webwork.maa.org/wiki/Introduction)

WeBWorK is a free Perl-based system for delivering individualized homework problems over the web. It was originally developed in 1995 by Profs. Arnold Pizer and Michael Gage at the University of Rochester Department of Mathematics, for use in mathematics instruction. It is now supported by a team of developers from several institutions and is used for a variety of subjects.

WeBWorK enhances the educational process in several ways. By providing students with immediate feedback as to the correctness of their answers, students are encouraged to make multiple attempts until they succeed. By individualizing problems, cheating is discouraged. By providing instructors with real-time statistics, lesson plans can be customized to better serve students.

The major way in which WeBWorK differs from other web-based homework systems is in the way problems are written. The PG ("problem generation") language allows the inclusion of both Perl and LaTeX code, allowing problem authors to take advantage of the syntactic efficiency of Perl and the typographical flexibility of LaTeX (which is pretty much necessary for rendering mathematics expressions).

The process of defining a problem is highly modular, with various pluggable display macros, answer evaluators, and graders. This allows for a high degree of freedom in defining how problems behave.

A library of canned problems is provided, with over 12,000 problems contributed to the National Problem Library (or NPL) at last count. A large collection of macro files which simplify writing new problems is also available. (Use the MathObjects macros provided by Davide Cervone where ever possible to simplify the writing of the problem and to make it easier to maintain.)

WeBWorK uses Apache with mod_perl, MySQL, LaTeX, dvipng, the GD graphics library, and several CPAN modules. WeBWorK is being used on Linux, FreeBSD, and Solaris, and will run on any UNIX-like system that supports its dependencies.

Managing Problem Sets

This is the method I followed to create problem sets.

First go into the “HMWK Sets Editor” section and create a new homework set. Go into “Edit Assigned Users” for that problem set, and add yourself. This will allow you to try the problem.

Inside the edit problems section, you can add a new problem. When you save the problem use the format: setProblemSetName/problemName. For example is you were making a problem in the “Limits” problem set, and you wanted to call the problem “squeezeThm”; you should use the convention “setLimits/squeezeThm”. Using this format will save the question to the “setLimits” folder, or create one if it does not already exist. Once you have created a problem you can edit it from the “File Manager” section, or the “HMWK Sets Editor Section”.

> **WARNING**: After a long period of inactivity the site will log you out when you refresh the page. If you edit the questions on the site (instead of copy and pasting from a text file), you can spend a long time without refreshing a page. I had a copy of instances when I tried to save and lost all my work because of this. Get into the habit of making a copy of the problem before you save it.

Changing the Problem Set and Course Info

When you go into the "Homework Sets" section at the top of the menu page, you will be brought to a page that has a list of all the homework sets. At the right of the screen there is a bar that says "Course Info [edit]". If you click on the edit button you will be brought to a screen similar to the problem sets editor. You do not need any special formatting for this section. When you are done, use the "save to" function.

Each of the problem sets also has a "Problem Set Info" section that you can edit. If you do wish to edit this section you will have to you proper problem formatting. Click on "Edit Assigned Users" next to the problem set you wish to edit. Navigate to the student for whom you like to edit the problem set. You will be given a list of options.

Student Specific Changes

You can make changes to the problem sets that only affect one student. This would include, changing due dates, changing the weight or maximum number of attempts allowed.

Go into the "HMWK Sets Editor". At the top of the page there is an option to: "Edit individual versions". You will then get a list of students. Click on "Edit data for [student name]".

Creating Quizzes

Any problem set that you make can be turned into a timed quiz.

Go into the "HWRK Sets Editor: and choose the problem set you want. At the top of the page there will be a drop down menu next to "Assignment Type". Choose "Gateway/quiz", and then "save changes". You will get a new list of options on how you would like to structure the test.

Note: When you test the quiz you will have an option to check answers and show solutions. The student will not have the same options.

Caution: I got error messages in the questions after switching to quiz mode, and then back to homework mode. I ended up having to copy and paste each question into a new file. (There was probably a simpler solution, but that was the only thing I could come up with).

Basic Syntax

Generic Problem

Your problems should follow this basic set-up:

```
# Description
## Tagging

DOCUMENT();

loadMacros(
   "PGstandard.pl",
   "MathObjects.pl",
);

# The context does not have to be numeric
Context(“Numeric”);

# Print problem number and point value (weight) for the problem
TEXT(beginproblem() );

Context()->texStrings;
BEGIN_TEXT
…
END_TEXT
Context()->normalStrings;

# This is the answer evaluator. Here “ans” is not yet defined
ANS($ans->cmp() );

Context()->normalStrings;
BEGIN_SOLUTIONS
…
END_SOLUTIONS
Context()->texStrings;

ENDDOCUMENT();
```

Macros

There are many specialized problem techniques that require you to load separate macros. A basic problem will only need "PGstandard.pl", and "MathObjects.pl". Refer to the above sample to see how to load them.

Descriptions, Tagging, and Comments

To put comments in the body of the text of the problem you have to use the "#" symbol. This will allow you to include text that does not affect the rendering of the problem. It is good practice to include comments so that it is easier for others to edit your problems in the future. You should also include a description at the beginning of the problem.

Right after the description you can include various "tags" which allow WeBWorK to index your problem in the problem library. These tags need two pound symbols instead of one.

Example of the first few lines of a problem:

```
# DESCRIPTION
# A simple sample problem that asks students to
# differentiate a trigonometric function.
# WeBWorK problem written by Gavin LaRose,
# <glarose(at)umich(dot)edu>
# ENDDESCRIPTION

## DBsubject('WeBWorK')
## DBchapter('Demos')
## DBsection('Problem')
## KEYWORDS('')
## TitleText1('')
## EditionText1('')
## AuthorText1('')
## Section1('')
## Problem1('')
## Author('Gavin LaRose')
## Institution('UMich')
```

Variables and Arrays

In Perl syntax, variables are represented by a "$" symbol and arrays by an "@" symbol. Arrays do not have to be initialized, and their size is by how many variables you store in it. All statements in Perl must end with a semicolon. With some exceptions white space does not have any meaning.

Example:

```
$a = 5;
@b = (3,4,6);
```

The previous example creates an array of length 3. If you would like to access the value stored in the second indexed position of the array you can use: $b[2]. Remember that array indexing starts at 0, so $b[2] will return "6".

Randomizing

There are two main ways to randomize *numerical values.*

1)

```
    $a = random(2,10,2);
```

This creates a random of the form 2+2n between 2 and 10 where n is an integer.

In general:

```
    $a = random($startValue,$endValue,$interval);
```

2)

```
    $b = random_list(2,5,3,24,4);
```

This chooses a number from a list of arbitrary length.

You can also randomize *text and formulas*.
Example:

```
    @per = (‘monthly’, ‘annually’,’weekly’);
    $tag = random(0,2,1);
    $period = $per[$tag];
```

Here the value stored in “tag” randomizes which index of the array will be accessed.

For and If Statements :

http://perldoc.perl.org/perlintro.html

The basic structure of an “if” statement is:

```
if(condition)
{ … }
elsif(another condition)
{ … }
else
{ …}
```

One of the uses for the if statement is to make sure the random variables make sense within the context of the problem. For instance if you had the function f(x) = (x+$a)/(x+$b), you could use if statements to make sure that “a” does not equal “b”: if($a == $b) {$a = $a +1; }.

There are many ways to do a “For” statement. The one I use is:

```
For my $i (0..3) { … }
```

The “my $i” creates a local variable that is only active inside the for loop.

Conditions:

Numeric comparisons follow the usual conventions (>= for greater than or equal to, < for less than). The only thing you have to remember is to use “ = =” for equal to, and “=!” for not equal to.

String comparisons use two letter abbreviations. The most important one is “eq” for equal to. Any comparisons for greater than (gt) or less than (lt) follow the acii value of the letters, where letters later in the alphabet are rated higher.

Ex:

```
If($a eq ‘monthly’) { … }
```

Displaying Text

Use the following code for text that you want to include in the problem:

```
Context()->texStrings;
BEGIN_TEXT
 …
END_TEXT
Context()->normalStrings;
```

The “texStrings” and “normalStirngs” are context flags, which are discussed in the Math Objects section. If you are not using Math Objects, you do not need to include them.

Simple Commands Within the TEXT Section:
These commands do not need to end with a semicolon.
$BR : Line Break
$PAR : New paragraph
$BOLD : Begin bold
$EBOLD : End bold (any text between $BBOLD and $EBOLD will be bolded)
Note: Simply putting a space between paragraphs will not create a new paragraph when the problem is displayed. You must include the $PAR command.

LaTeX commands:
Many simple LaTeX commands can be used within the TEXT section of the problem.
\(LaTeX commands\) : displays on the same line on which it was typed.
\[LaTeX commands\] : creates a new paragraph to display, and puts everything in the “\displaystyle” mode.

Limits:
If you want the code for a “limit” to display properly you have to be in the displaystyle mode. This means explicitly including “\displaystyle” or using square brakets \[…\]\, which will put the text into its own paragraph.

Ex.

```
\( \displaystyle \lim_{x \to 0} f(x) \) # limit of f(x) as x approaches zero
```

```
# OR
\[\lim_{x \to 0} f(x)\] # this displays the same thing in a new paragraph
```

You can display any variables in the problem text by including their reference either within the text itself, or within the brackets with the LaTeX code.

Ex: Consider the function \(f(x) = frac{x+$a}{x-2}\)
  or: The initial population is $a.

Peicewise functions:
Ex.

```
 \[f(x) = \begin{cases}
      x\sin\left(\frac{1}{x}\right) &  x \neq 0\\
      0 & x = -$b\\
    \end{cases}\]
```

Perl Commands:
Any Perl commands must be included within curly brackets. The main reason to include Perl commands is put in a space for student answers.

Ex: \(f'(x) = \) \{ans_rule(40)\}

\{ans_rule(40)\} leaves an answer blank 40 characters long in the text. There are built in subroutines that will store any value the student types in the blank.

Solutions

If you would like to include a solution section in the problem you can include the following code. The solution section will not be displayed until after the dew date of the problem set.

```
Context()->texStrings;
BEGIN_SOLUTION
 …
END_SOLUTION
Context()->normalStrings;
```

Another way to include a solution section is to use the following code.

```
Context()->texStrings;
SOLUTIONS(EV3(<<'END_SOLUTION) );
…
END_SOLUTION
Context()->normalStrings;
```

There is no difference between the two. "BEGIN_SOLUTION" is just a shortcut for "SOLUTIONS(EV3(<<'END_SOLUTION));"

Math Objects

Math Objects are a newer addition to WeBWorK. You will notice that a lot of the problems in the problem library do not use them. They are both easier to use and to maintain than the old system.

Each problem is given a “context” that specifies how the math objects will behave. The most common type of context is “Numeric”. This context allows you to specify functions, take their derivatives, and evaluate them at a given point. There are also “context flags” which allow you to modify the context. For example, the “Numeric” context has “x” as the only variable you can use in function. If you would like to include a different variable (say “y”) then you include the line:

```
Context()->variables->add(y=>"Real");
```

Example defining functions

```
# Defines the function
$f = Formula("x^2");

#Takes the derivative of the function
$dx  = $f->D;

#Evaluates the derivative at the value represented by the variable a
$dxa = $fx->eval(x=>"$a");
```

You can also multiply functions together. For example if you have:

```
$f = Formula(“x^2”);
$g = Formula(“cos(x)”);
$h = $f*$g;
```

The formula contained in $h will be: x^2cos(x). You have to make sure that both things you are multiplying together are actually formulas.

```
#this will not work
$f = Formula(“sin(x)”);
$g = x*$f; # x is not a Formula so this equation will not work
```

For a more complete treatment of Math Objects:
http://webwork.maa.org/wiki/Introduction_to_MathObjects#List_of_Basic_Contexts

Answer Evaluators

Math Objects make answer evaluation easy, because the basic answer checker subroutines are already coded into the objects. If there are more than one questions in the problem, just put the code for the answer checkers in the order that they appear in the problem.

Example: Say the answer is “x^2”.

```
$ans = Formula(“x^2”);
ANS($ans->cmp() );
```

You have to make sure that your answer is a math object.

Example: (WRONG)

```
ANS(4->cmp() ); # Doesn’t work because “4” is not an object.
```

Example: (Right)

```
ANS(Compute(“4”)->cmp() );
         #OR
ANS(Real(“4”)->cmp() );
          #OR
$ans = Compute(“4”);
ANS($ans->cmp() );
```

Function answer elvaluators:
You do not have to worry about the form in which a function is entered. The evaluator works by using test points, and comparing the behaviour of the correct answer to the student’s answer.

Problem Techniques

Strings in Student Answers

http://webwork.maa.org/wiki/StringsInContext

Many of the problems in the problem library were created before math objects and have different answer checkers. With these older answer checkers you have to specify whether you are dealing with a number or a function, and you have to include more macros at the beginning of the problem.

The only time I had to use the old answer checker was with limit questions where the answer could be ‘DNE’, but even here you don’t need the old answer checker. You can actually use strings with Math Objects, you just have to specify the words you are using within the context.

For example, if you wanted to make ‘over’ and ‘under’ valid answers you could include the following code.

```
Context()->strings->add(over=>{},under=>{});
```

Then your answer could take the form:
$ans = Compute("over");

You can also specify an alias for your string so that two different strings give a correct answer. For example, here both "none" and "N" mean the same thing.

```
Context()->strings->add(none=>{},N=>{alias=>"none"});
```

By default the stings are not case sensitive. If you would like to make them case sensitive you can use a context flag:

```
Context()->strings->add(none=>{caseSensitive=>1});
# the word none is now case sensitive
```

Caution: There are some strings (ex: pi, DNE, INF) that are already defined. If you try to specifically add them to the context you will get an error message.

Tolerance in numerical answers:

http://webwork.maa.org/wiki/NumericalTolerance

The default tolerance in numerical answers is +/- .005.

You can you a Context Flag to modify this setting for all the answers in the question:

```
Context("Numeric")->flags->set(
  tolerance=>0.0001,
  tolType=>"absolute"
);
```

Alternatively you can make the tolerance specific to one answer checker you can use:

```
$ans = Compute("1.5708")->with(
  tolType=>'absolute',
  tolerance=>'.0001'
);
```

In the above examples "tolType" can either be "absolute" or "relative" (the default value). Relative specifies a percent error.

You can also have tolerance in Formula answers. The answer checker for Formulas works by evaluating the student and correct answers at certain test points. The discrepancy between the two answers forms the basis of the tolerance test.

Ex.

```
$ans = Formula("(2)^(1/2)x")->with(
  tolType=>'absolute',
  tolerance=>'.1'
);
```

Lists as Answers:

http://webwork.maa.org/wiki/ListAnswers
One of your answers could be a list of numbers: for example if you asked the student for all the x-intercepts of a function.

Ex.

```
$roots = List(3,2);
ANS($roots->cmp() );
```

By default the student will get partial credit and will be prompted with hints if they input the wrong number of items in the list. You can change these defaults by using the following commands when you call the answer evaluator:

```
ordered=>0 # makes the order of the answers important
showHints=>0
showLengthHints=>0
partialCredit=>0
```

For example if you did not want the student to receive partial credit you would write:

```
ANS($roots->cmp(partialCredit=>0) );
```

If the student enters more numbers then they are suppose to, there mark will be the number of correct answers divided by the number of answers they inputted. In the above example, if the student entered "1,2,3" then they would get 67% on the question.

If you have suggested to the student in the question that there could be more than one number in the answer then you should always use a list to compute the answer. For example if you have asked the student for all the roots of an equation, even if there is only one you should use a List Object to compute the answer. This ensures that the student does not an error message if they enter more than one number into the answer evaluator.

Showing Correct Answers:

http://webwork.maa.org/wiki/StringsInContext
By default the problems are set up to tell the student if their answers are correct. If there is an incorrect answer it will simply tell them that one of their answers is incorrect without telling them which one.

If you would like to tell the student exactly which question is incorrect use the code:

```
$showPartialCorrectAnswers = 1;
```

If you don't want the student knowing whether any of their answers are correct until after the due date use the code:
$showPartialCorrectAnswers = 0; # use this for multiple choice questions

Answer Hints

You can include hints that appear after a student has entered a specific answer.

For example if the correct answer is "-sin(x)" but the student enters "sin(x)":

```
…
loadMacros(
"PG.pl",
"MathObjects",
"answerHints", # be sure to include this macro
);

…
$ans = Formula("-sin(x)");
ANS(->cmp()
->withPostFilter(AnswerHints(
  Formula("sin(x)") => "What your sign!",
))
);
```

Alternatively you can include a hint that doesn't show up until the student has tried the question at least once:

```
Context()->texStrings;
HINT(EV3(<<'END_HINT'));
Be careful with your signs in this question!
END_HINT
Context()->normalStrings;
```

The word "HINT" will automatically appear so you do not need to type it in.

Hints with Penalties

You can give student's hints that take off points if they are used:

```
…
loadMacros(
"PG.pl",
"MathObjects",
"problemPanic", # be sure to include this macro
);

…
```

```
# this is what displays before any marks are taken off
# the penalty specifies what percentage of the points are deducted
Context()->texStrings;
BEGIN_TEXT
  \{Panic::Button(label => "Hint", penalty => .25)\}
    (you will lose 25% of your points if you do)
END_TEXT
Context()->normalStrings;

# this is what displays once the student has pushed the hint button
 if ($panicked) {
      BEGIN_TEXT
       $PAR $BBOLD Hint: Use the substitution t = x^2.
      END_TEXT
    }

# this allows the student to get part marks and see which answers they got right
install_problem_grader(~~&avg_problem_grader);

# You must include this line if you want marks deducted
Panic::GradeWithPenalty;
```

Problem Templates

Most of these questions are designed to be very easy to make copies of with only minor changes required. I have made red anything that needs to be changed if you are to make a different question with the code.

Taylor Polynomials

```
DOCUMENT();

loadMacros(
  "PGstandard.pl",
  "MathObjects.pl",
);


TEXT(beginproblem());


$showPartialCorrectAnswers = 1;


Context("Numeric");
Context()->strings->add(over=>{},under=>{});

#Number of decimals to use in the approximation
$dec = 3;

$smallDec = 10**{-$dec};

# What degree Taylor Polynomial do you want
$n = 2;
$n1 = $n+1; # This is so that you can calculate a remainder

# Around which value are you centering the approximation
$a = 8;

# For which value of x would you like to evaluate the function
$x = 10;

# The formula you would like to approximate
$f[0] = Formula("(1+x)^(1/2)");

# The zero degree approximation
$tn[0] = $f[0]->eval(x=>"$a");
```

```
# code to compute the nth degree approximation
$nfact = 1;
for my $i (1..$n)
{
$f[$i] = $f[$i-1]->D;
$fa = $f[$i]->eval(x=>"$a");
$tn[$i] = $tn[$i-1]+$fa*($x-$a)**($i)/$nfact;
$nfact = $nfact*($i+1);
}

# computing the error
$f[$n1] = $f[$n]->D;
$upF = $f[$n1]->eval(x=>"$a");
$error = $upF*($x-$a)**($n1)/$nfact;
$sError= sprintf("%0.4f",$error);
if($sError < 0) { $sError = $sError*-1; }

$ans = Compute("$sError")->with(
  tolType=>'absolute',
  tolerance=>'.0005'
);

# Is the nth polynomial an over or under estimate
# by default strings are case insensitive
$size = Compute("under");

#How many decimal places do you want your answer to display
# In this question there are four decimal places
 for my $i (1..$n)
{
$sTn[$i] = sprintf("%0.4f",$tn[$i]);
}

Context()->texStrings;
BEGIN_TEXT
Consider the function \(f(x) = \sqrt{x+1}\).
$BR Let \(T_n\) be the \(n^{th}\) degree Taylor approximation of \(f($x)\) about \(x =
$a\).

$PAR Find:
END_TEXT
Context()->normalStrings;

for $j (1..$n)
{
Context()->texStrings;
BEGIN_TEXT
$BR \(T_{$j} = \) \{ans_rule(10)\}
END_TEXT
Context()->normalStrings;
}

Context()->texStrings;
BEGIN_TEXT
$BR $BBOLD Use $dec decimal places in your answer, but make sure you carry all
decimals when preforming calculations $EBOLD
$PAR

\(T_{$n}\) is an \{ans_rule(10)\} (over/under) estimate of \(f($x)\).
$PAR

If \(R_{$n}\) is the remainder given by the Lagrange Remainder Formula:
$BR \(\left|R_{n}\right| \le \) \{ans_rule(10)\}
$PAR

END_TEXT
Context()->normalStrings;

#Adjust the tolerance in the answer
for my $i (1..$n)
{
ANS(Compute("$sTn[$i]")->cmp(tolType=>'absolute',
          tolerance=>.000005) );
}

ANS($size->cmp() );
ANS($ans->cmp() );


# You will have to change most of the solution to fit the question
Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
The \(n^{th}\) degree approximation of \(f($x)\) about \(x=$a\) is given by:
$BR \(T_n = f($a) + f'($a)($x-$a) + \cdots + \frac{f^n($a)}{n!}($x-$a)^n\)

$PAR
\(f'(x) = \frac{1}{2\sqrt{1+x}}\),
$PAR \(f''(x) = -\frac{1}{4(1+x)^{\frac{3}{2}}}\),
$PAR \(f'''(x) = \frac{3}{8(1+x)^{\frac{5}{2}}}\).

$PAR
Therefore,
$BR \(T_1 = $sTn[1]\),
$BR \(T_2 = $sTn[2]\),

$PAR
\(f''(x)\) is always negative in the interval [$a,$x] so \(T_{$n}\) is a underestimate of
```

```
\(f($x)\).

$PAR
By the Legrange Remainder Theorem we know we can find an M such that:
$BR \(\left|R_{$n}\right| \le \frac{M}{$n1 !}\left|$x-$a\right|^{$n1}\), where \(M \ge
\left|f^{$n1}(t)\right|\) for all \(t\) in the interval [$x,$a].

$PAR
By examining \(f'''(x)\) we find that it is biggest when \(t=$a\).   Therefore, a suitable
choice is \(M=f'''($a)\).

$PAR
By calculation we find:
\(\left|R_$n\right| \le $sError\).

END_SOLUTION
Context()->normalStrings;

ENDDOCUMENT();
```

Newton’s Method

```
DOCUMENT();      # This should be the first executable line in the problem.

loadMacros(
"PGstandard.pl",
"MathObjects.pl",
);

Context("Numeric");

# Adjust the tolerance for your answers
Context()->flags->set(
  tolerance=>0.0001,
  tolType=>"absolute"
);

# Of what formula are you trying to find the roots
$f = Formula("ln(x)+x-4");
$df = $f->D;

# What is the initial guess
$x[0] = 3;
# How many iterations Newton’s Method do you want
$n = 3;

for my $i (1..$n)
{
$x[$i] = $x[$i-1] - ($f->eval(x=>"$x[$i-1]"))/($df->eval(x=>"$x[$i-1]"));
}

# How many decimal places would you like in the answers
# This question has 5 decimal places
# It is a good idea to have one more decimal place then the tolerance on the
# question
for my $i (1..$n)
{
$xs[$i]=sprintf("%0.5f",$x[$i]);
}

Context()->texStrings;
BEGIN_TEXT
Use Newton's method to find the roots of the equation: \(\ln(x) = 4-x\).  Start with
\(x_0 = $x[0]\). $BR Report your answers to 4 decimal places.
END_TEXT
Context()->normalStrings;


for $j (1..$n)
```

```
{
Context()->texStrings;
BEGIN_TEXT
$BR \(x_{$j} = \) \{ans_rule(10)\}
END_TEXT
Context()->normalStrings;
}

Context()->texStrings;
BEGIN_TEXT
$PAR
$BBOLD CAUTION: $EBOLD Carry all decimal places when performing calculations.

END_TEXT
Context()->normalStrings;

for my $i (1..$n)
{
$ans[$i] = Compute("$xs[$i]");
ANS($ans[$i]->cmp() );
}
# All of the solution will have to be changed
Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
We have \(\ln(x) = 4-x\), so \(0 = \ln(x) +x-4 \).

$PAR If \(f(x) = \ln(x)+x-4\), then \(f'(x) = \frac{1}{x}+1\).

$PAR
Newton's Method says:
\(x_{n+1} = x_n - \frac{f(x)}{f'(x)}\)
\( = x_n - \frac{ln(x_n)+x_n-4 }{\frac{1}{x_n}+1}\)

$BR \(x_1 = $ans[1]\)
$BR \(x_2 = $ans[2]\)
$BR \(x_3 = $ans[3]\).

END_SOLUTION
Context()->normalStrings;

ENDDOCUMENT();
```

Curve Sketching

This question asks the student to find asymptotes, intercepts and critical points. It divides the function into intervals and asks the student for the shape of the graph in each interval. It is the best way I could think of to do curve sketching questions online.

```
DOCUMENT();

# Make sure to include all of these macros
loadMacros(
  "PGstandard.pl",
  "parserPopUp.pl",
  "PGgraphmacros.pl",
  "MathObjects.pl",
  "unionTables.pl",
  "PGgraphmacros.pl",
);

# This is creates a graph to represent each possible shape of the graph
@e[0] = ".25*(x-1)^2 + 1 for x in <1,4> using color:red and weight:2";
@e[1] = ".25(x-4)^2 + 1  for x in <1,4> using color:red and weight:2";
@e[2] = "-.25(x-1)^2 +4 for x in <1,4> using color:red and weight:2";
@e[3] = "-.25(x-4)^2 + 4 for x in <1,4> using color:red and weight:2";


@egraph = ( );
for my $i (0..3) {
$egraph[$i] = init_graph(-.5,-.5,5,5,'axes'=>[0,0]);
$egraph[$i]->lb('reset');
$egraph[$i]->lb(new Label(4.5,0.25,'x','black','center','middle'));
$egraph[$i]->lb(new Label(0.25,4.5,'y','black','center','middle'))
}

for my $i (0..3) {
  plot_functions( $egraph[$i], $e[$i]);
  $egraph[$i]=image(insertGraph($egraph[$i]),
  width=>200,height=>200,tex_size=>450);
}

# The popup number refers to a graph
# There is one popup for each interval on the function
# By interval I mean a section between critical points
# 1 is increasing concave up
# 2 is decreasing concave up
# 3 is increasing concave down
# 4 is decreasing concave down
# You have to change the right most number which refers to the correct answer
# The number of popups depends on the number of critical points
$popup1 = PopUp(["?","1","2","3","4"], "1");
$popup2 = PopUp(["?","1","2","3","4"], "4");
```

```
$popup3 = PopUp(["?","1","2","3","4"], "3");
$popup4 = PopUp(["?","1","2","3","4"], "2");

TEXT(beginproblem());

$showPartialCorrectAnswers = 1;

# You have to change what information you would like the student to enter
# ex intercepts  asymptotes
# You will have to change the number of answer spaces and popups there
# are depending on the number of critical points
Context("Numeric");

Context()->texStrings;
BEGIN_TEXT
Consider the function \(f(x) = e^{-\frac{x^2}{2}}\).

$PAR
The y-intercept is at: \(y= \) \{ans_rule(10)\}
$BR
There a horizontal (or oblique) asymptote at \(y = \) \{ans_rule(10)\}
$BR $BBOLD NOTE: $EBOLD This could be a function of x.

$PAR
If \(A\), \(B\), and \(C\) are critical points, then \(f(x)\) can be broken up into the
following unique intervals:
$BR (\(-\infty,A\)), (\(A,B\)), (\(B,C\)), (\(C,\infty\)).
$PAR
\(A =\) \{ans_rule(10)\} $BR
\(B = \) \{ans_rule(10)\} $BR
\(C = \) \{ans_rule(10)\} $PAR

$PAR For each of the intervals listed below, select the graph best representing the
shape of the function in that interval.

$PAR

(\(-\infty,A\)): \{ $popup1->menu() \}
$BR $SPACE $SPACE $SPACE

 (\(A,B\)): \{ $popup2->menu() \}
$BR $SPACE $SPACE $SPACE

(\(B,C\)): \{ $popup3->menu() \}
$BR $SPACE $SPACE $SPACE

 (\(C,\infty \)): \{ $popup4->menu() \}

$BR
```

```
$BR
$BCENTER

\{
BeginTable().
  AlignedRow([$egraph[0],$egraph[1]]).
  TableSpace(5,0).
  AlignedRow(["1","2"]).
  TableSpace(25,6).
  AlignedRow([$egraph[2],$egraph[3]]).
  TableSpace(5,0).
  AlignedRow(["3","4"]).
EndTable();
\}

$BR
(Click on a graph to enlarge it.)
$ECENTER
$PAR

END_TEXT
Context()->normalStrings;

# Asymptotes
$va = Compute("1");
$oa = Formula("0");

# Critical Points
$ans1=Compute("-1");
$ans2=Compute("0");
$ans3=Compute("1");

ANS($va->cmp() );
ANS($oa->cmp() );

ANS($ans1->cmp() );
ANS($ans2->cmp() );
ANS($ans3->cmp() );

ANS( $popup1->cmp() );
ANS( $popup2->cmp() );
ANS( $popup3->cmp() );
ANS( $popup4->cmp() );

################# SOLUTIONS #######################

# This creates a graph to be displayed in the solution
# You have to change for what interval the graph is displayed
# The arguments for init_graph are xmin ymin xmax ymax
```

```
$gr = init_graph(-3,-.1,3,1.5,
axes=>[0,0],
grid=>[1,1],
size=>[400,400]
);

# This is where you specify the actual function
# You can specify for which interval you would like the function displayed
# In this question the interval is -3 to 3
add_functions($gr, "e^(-x^2/2) for x in <-3,3>" .
  " using color:blue and weight:2");

# You will have to change a lot of the solutions but the basic template can still be used
Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR

$BCENTER
\{ image( insertGraph($gr),
width=>200,height=>200,tex_size=>800 ) \}
$BR
(The correct graph of \(f(x)\))
$ECENTER

By inspection we can see that  \(\displaystyle \lim_{x \to \infty} f(x) = \lim_{x \to -
\infty} f(x)=0\).  Also, \(f(0) = 1\), so the the y-intercept is at \(y=1\).
$PAR
\(f'(x) = -x e^{-\frac{x^2}{2}}\).  From this we can find that there are critical points at
\(x=0\).
$PAR
\(f''(x) = e^{-\frac{x^2}{2}} (x+1)(x-1)\).  There are critical points at \(x=1, -1\).
$PAR
Therefore: $BR \(A=-1\), $BR \(B=0\), $BR \(C=1\).
$PAR
By examining the sign of the first and second derivative in the given intervals we find
\(f(x)\):
$BR
(\(-\infty,A\)): Increasing and concave up: therefore best represented by graph "1".
$BR  $SPACE $SPACE $SPACE (\(A,B\)): Increasing and concave down: therefore best
represented by graph "4".
$BR $SPACE $SPACE  $SPACE (\(B,C\)): Decreasing and concave down: therefore best
represented by graph "3".
$BR $SPACE $SPACE $SPACE (\(C,\infty \)): Decreasing and concave up: therefore best
represented by graph "2".

END_SOLUTION
Context()->normalStrings;

ENDDOCUMENT();
```

Inventory Control 1

Asks the student for inventory costs as a function of the number of items in each order

```
# Because of the randomization in this question all you need to do is find and replace the word
# TV with a different good
# You might also want to change the range of values the variables can take to make
# the question makes sense
# I would just stick with products where the numbers I have make sense because they
# were chosen to make the solution nice
DOCUMENT();

loadMacros(
  "PGstandard.pl",
  "MathObjects.pl",
  "answerHints.pl",
  "problemPanic.pl",
  "PGcourse.pl",
);

Context("Numeric");

# Carrying Cost for one TV
$h = random(20,40,2);

# Cost per delivery
$k = list_random(5,20,45,80);

# Number of TVs store expects to sell
$d = 40*$h;

# Delivery set up cost for each TV
$p = random(1,5,1);

$h2 = $h/2;
$pd = $p*$d;
$dk = $d*$k;
$dk2 = 2*$d*$k;
$rule = ($dk2/$h)**(1/2);


$x = $rule;

$f = Formula("($d*$k)/x + $p*$d + $h2*x");
$f1 = $f->eval(x=>"1");
$fd = $f->eval(x=>"$d");
$fx = $f->eval(x=>"$rule");

TEXT(beginproblem());
```

```
$showPartialCorrectAnswers = 1;



Context()->texStrings;
BEGIN_TEXT
An electronics store expects to sell $d TVs at a steady rate next year. The manager
of the store plans to order these TVs from the manufacturer by placing several orders
of the same size spaced equally throughout the year. The ordering cost for each
delivery is $$k for the setup costs and $$p per TV. The carrying costs, based on the
average number of TVs in inventory, amount to $$h per year for one TV.

$PAR
If \(C(x)\) is the inventory cost (which is the sum of the ordering costs and the carrying
costs) and \(x\) is the $BBOLD number of TVs in each order $EBOLD.
$BR \(C(x) = \) \{ans_rule(40)\}

$PAR
How many TVs the manager should request each time she places an order to minimize
the inventory cost? \{ans_rule(10)\}

$PAR


$PAR
END_TEXT
Context()->normalStrings;

# Problem hint that gives the cost equation but takes away 50 percent of the marks
Context()->texStrings;
BEGIN_TEXT
  \{Panic::Button(label => "Click here if  you cannot come up with an equation", penalty
=> .5)\}
   (you will lose 50% of your points if you do)
END_TEXT
Context()->normalStrings;


 if ($panicked) {
     BEGIN_TEXT
      $PAR $BBOLD Hint: $EBOLD \(C(x) = \frac{$dk}{x} + $pd + $h2 x\).
     END_TEXT
   }

# this allows the student to get part marks and see which answers they got right
install_problem_grader(~~&avg_problem_grader);

Panic::GradeWithPenalty;
```

```
ANS($f ->cmp() );

# These if else statements determine if cost minimization occurs at the end points
# ei 1 TV per order and d TVs per order
if($f1<$fx) {
    if($f1<$fd)
      {$x = 1; $ans = Compute("1");}
     else {$x = $d; $ans=Compute("$d"); }
   ANS($ans->cmp()
->withPostFilter(AnswerHints(
  Compute("$rule") => "Did you consider the end points?",
))
);
}

elsif($fx<$fd) {$x = $rule; $ans = Compute("$rule");
                              ANS($ans->cmp() );
}

else
{
$x = $d; $ans = Compute("$d");
 ANS($ans->cmp()
->withPostFilter(AnswerHints(
  Compute("$rule") => "Did you consider the end points?",
))
);
}


Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
The total number of orders needed is: \(\left(\frac{$d}{x}\right)\).  Since the cost per
TV in the shipment does not depend on the size of the shipment, the total shipment
cost is: \(\left($k\frac{$d}{x} + $pd\right)\).  The average number of TVs in the
inventory will be half of the shipment size, so the total carrying cost will be:
\(\left($h\frac{x}{2}\right)\).

$PAR Therefore: \(C(x) = \frac{$dk}{x} + $pd + $h2 x\).

$PAR
We can find max/min values by solving for \(x\) when \(C'(x) =0\).
$BR \(C'(x) = \frac{$h x^2 - $dk2}{x^2}\)
$BR \(x^2= \frac{$dk2}{$h}\)
$BR \(x = $rule\).
```

```
$PAR

You now need to check \(C(x)\) at this value, as well as at the end points (\(x=1,$d\)).
$BR \(C(1) = $f1\)
$BR \(C($d) = $fd\)
$BR \(C($rule) = $fx\)

$PAR Therefore the shipment size should be \($x\) TVs.
END_SOLUTION
Context()->normalStrings;

ENDDOCUMENT();
```

Inventory Control 2

This questions asks the student for a inventory cost as a function of number of orders placed

```
# Because of the randomization in this question all you need to do is find and replace the word
# laptops with a different good
# You might also want to change the range of values the variables can take to make
# the question makes sense
# I would just stick with products where the numbers I have make sense because they
# were chosen to make the solution nice

DOCUMENT();

loadMacros(
  "PGstandard.pl",
  "MathObjects.pl",
  "answerHints.pl",
  "problemPanic.pl",
  "PGcourse.pl",
);

Context("Numeric");

# Carrying Costs
$h = random(5,15,2);

# Number of laptops the store expects to sell
$c = random(9,20,1);
$d = 4*$c**2;

# Set up cost for each delivery
$k = 2*$h;

# Additional cost per laptop
$p = random(1,5,1);


$pd = $p*$d;
$dh = $d*$h;
$k2 = 2*$k;
$d2 = $d/2;
$rule = $c;
$x = $rule;

$f = Formula("$k*x + $d/(2*x) + $pd");
$f1 = $f->eval(x=>"1");
$fd = $f->eval(x=>"$d");
$fx = $f->eval(x=>"$rule");
```

```
TEXT(beginproblem());

$showPartialCorrectAnswers = 1;

Context()->texStrings;
BEGIN_TEXT
An electronics store expects to sell $d laptops at a steady rate next year. The manager
of the store plans to order these laptops from the manufacturer by placing several
orders of the same size spaced equally throughout the year. The ordering cost for each
delivery is $$k for the setup costs and $$p per laptop. The carrying costs, based on the
average number of laptops in inventory, amount to $$h per year for one laptop.

$PAR
If \(C(x)\) is the inventory cost (which is the sum of the ordering costs and the carrying
costs) and $BBOLD \(x\) $EBOLD is the $BBOLD number of orders $EBOLD.
$BR \(C(x) = \) \{ans_rule(40)\}

$PAR
How many orders should the store manager place? \{ans_rule(10)\}

$PAR


$PAR
END_TEXT
Context()->normalStrings;

# Problem hint that gives the cost equation but takes away 50 percent of the marks
Context()->texStrings;
BEGIN_TEXT
  \{Panic::Button(label => "Click here if  you cannot come up with an equation", penalty
=> .5)\}
   (you will lose 50% of your points if you do)
END_TEXT
Context()->normalStrings;


 if ($panicked) {
     BEGIN_TEXT
      $PAR $BBOLD Hint: $EBOLD \(C(x) = $k x + \frac{$d2}{x} + $pd\).
     END_TEXT
   }

# this allows the student to get part marks and see which answers they got right
install_problem_grader(~~&avg_problem_grader);

Panic::GradeWithPenalty;

ANS($f->cmp() );
```

```
# These if else statements determine if cost minimization occurs at the end points
# ei 1 TV per order and d TVs per order
if($f1<$fx) {
    if($f1<$fd)
      {$x = 1; $ans = Compute("1");}
     else {$x = $d; $ans=Compute("$d"); }
   ANS($ans->cmp()
->withPostFilter(AnswerHints(
  Compute("$rule") => "Did you consider the end points?",
))
);
}

elsif($fx<$fd) {$x = $rule; $ans = Compute("$rule");
                           ANS($ans->cmp() );
}

else
{
$x = $d; $ans = Compute("$d");
 ANS($ans->cmp()
->withPostFilter(AnswerHints(
  Compute("$rule") => "Did you consider the end points?",
))
);
}




Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
Since the cost per laptop in the shipment does not depend on the size of the shipment,
the total shipment cost is: \(\left($k x+ $pd\right)\).  The average number of laptops
in the inventory will be half of the shipment size \(\left(\frac{1}{2}\frac{$d}{x}\right)\),
so the total carrying cost will be: \(\left($h\frac{$d2}{x}\right)\).

$PAR Therefore:  \(C(x) = $k x + \frac{$d2}{x} + $pd\).

$PAR
We can find max/min values by solving for \(x\) when \(C'(x) =0\).
$BR \(C'(x) = \frac{$k2 x^2 - $dh}{x^2}\)
$BR \(x^2= \frac{$dh}{$k2}\)
$BR \(x = $rule\).
```

```
$PAR

You now need to check \(C(x)\) at this value, as well as at the end points (\(x=1,$d\)).
$BR \(C(1) = $f1\)
$BR \(C($d) = $fd\)
$BR \(C($rule) = $fx\)

$PAR Therefore there should be $x shipments.

END_SOLUTION
Context()->normalStrings;

ENDDOCUMENT();
```

Loans and Mortgages 1

```
 # This question randomizes the compounding period, the interest rate, the initial
#mortgage value, and the amortization period.
# Just copying and pasting this code will create an entirely different question.

DOCUMENT();

loadMacros(
  "PGstandard.pl",
  "MathObjects.pl",
  "problemPanic.pl",
);

Context("Numeric");
Context()->flags->set(
 tolerance=>0.05,
 tolType=>"absolute"
);

$showPartialCorrectAnswers = 1;

#Range of values for the mortgage
$p = random(250000,750000,50000);

$rate = random(3.5,7.5,.5);
$r = $rate/100;

# Randomizing mortgage periode
@compound=('bi-weekly','monthly','semi-annually','annually','bi-monthly');
$tag = random(0,4,1);

# number of months in each mortgage period
@periods = (24,12,2,1,6);
$per = $periods[$tag];

$comp = $compound[$tag];

$years = random(5,30,5);
$n = $years*12;

Context()->texStrings;
BEGIN_TEXT
Suppose you take out a mortgage for $$p at $rate% interest per year compounded
$comp.  If your mortgage is amortized over $years years, what is your monthly
mortgage payment?  How much interest will you pay the lender by the end of the
mortgage?

$PAR
```

```
What is the monthly interest rate corresponding to the effective annual rate?
$BR \(r_m = \) \{ans_rule(10)\}

$PAR
What are the monthly payments? \{ans_rule(10)\}

$PAR
What is the total interest paid? \{ans_rule(10)\}

$PAR

END_TEXT
Context()->normalStrings;

$ear = (1+$r/$per)**($per);
$ra = ($ear-1)*100;
$goodear = sprintf("%0.2f",$ra);

$rm = $ear**(1/12)-1;
$ans1 = $rm*100;


$y =  -$p*$rm/((1+$rm)**(-$n)-1);
$y1 = sprintf("%0.2f",$y);

$ti = $n*$y-$p;
$ti1 = sprintf("%0.2f",$ti);

$a = sprintf("%.04f",$rm);
$month = Compute("$a*100");


 ANS(Compute($month)->cmp(tolType=>'absolute',
           tolerance=>.005) );


 ANS(Compute($y1)->cmp(tolType=>'absolute',
           tolerance=>1) );
 ANS(Compute($ti1)->cmp(tolType=>'absolute',
           tolerance=>1) );


# A problem hint that gives the student the monthly rate and deducts 25% of the marks
Context()->texStrings;
BEGIN_TEXT
  \{Panic::Button(label => "Click here if you cannot come up with a monthly rate ",
penalty => .25)\}
   (you will lose 25% of your points if you do)
END_TEXT
```

```
Context()->normalStrings;

 if ($panicked) {
     BEGIN_TEXT
      $PAR $BBOLD Hint: $EBOLD The monthly rate is $month%.
     END_TEXT
   }

# this allows the student to get part marks and see which answers they got right
install_problem_grader(~~&avg_problem_grader);

Panic::GradeWithPenalty;

# The if else statements correspond to different solutions for
# Different amortization periods
if($comp eq 'annually')
{
Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
We don't need to calculate an EAR.
$BR We know that \(r_A = (1+r_m)^{12} - 1\)
$BR Therefore: \(r_m = (r_A+1)^{\frac{1}{12}}-1\), and the monthly rate is $month%.

$PAR
From the equation for mortgage and loan calculations we know that:
$BR \(Y = -\frac{P (r_m)}{(1+r_m)^{-N}-1}\),
$BR \(= -\frac{$p ($a)}{(1+$a)^{-$n}-1}\),
$BR \(=  $p.\)

$PAR
To find the total interest paid you need to calculate the total value of your payments
and subtract the initial value of loan.
$BR \(Interest = N(Y) - P\),
$BR \(= $n ($y1) -$p\),
$BR\(= $ti1.\)

END_SOLUTION
Context()->normalStrings;
}

elsif($comp eq 'monthly')
{
Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
We know that the monthly interest rate is just \(\frac{$rate}{12} = $month%\).

$PAR
From the equation for mortgage and loan calculations we know that:
$BR \(Y = -\frac{P (r_m)}{(1+r_m)^{-N}-1}\),
$BR \(= -\frac{$p ($a)}{(1+$a)^{-$n}-1}\),
$BR \(=  $y1.\)

$PAR
To find the total interest paid you need to calculate the total value of your payments
and subtract the initial value of loan.
$BR \(Interest = N(Y) - P\),
$BR \(= $n ($y1)-$p\),
$BR\(=  $ti1.\)

END_SOLUTION
Context()->normalStrings;
}


else
{
Context()->texStrings;
BEGIN_SOLUTION
$PAR SOLUTION $PAR
First we need to calculate an EAR.
$BR Since there are $per compounding periods per year we know that \(EAR =
((1+\frac{$r}{$per})^{$per}-1)100%\).
$BR \(EAR = $goodear%.\)

$PAR Now we need to calculate the corresponding monthly rate.  We know that \(r_A =
(1+r_m)^{12}\)
Therefore: \(r_m = (r_A)^{\frac{1}{12}}-1\), and the monthly rate is $month%.

$PAR
From the equation for mortgage and loan calculations we know that:
$BR \(Y = -\frac{P (r_m)}{(1+r_m)^{-N}-1}\),
$BR \(= -\frac{$p ($a)}{(1+$a)^{-$n}-1}\),
$BR \(= $y1.\)

$PAR
To find the total interest paid you need to calculate the total value of your payments
and subtract the initial value of loan.
$BR \(Interest = N(Y) - P\),
$BR \(= $n ($y1)-$p\),
$BR\(=  $ti1.\)

END_SOLUTION
Context()->normalStrings;
}

ENDDOCUMENT();
```

Loans and Mortgages 2: Most expensive house possible

```
# This question randomizes the compounding period, the interest rate, the initial
#mortgage value, and the amortization period.
# Just copying and pasting this code will create an entirely different question.
# Student determines what the most expensive house they can afford
DOCUMENT();

loadMacros(
  "PGstandard.pl",
  "MathObjects.pl",
  "problemPanic.pl",
);

Context("Numeric");
Context()->flags->set(
 tolerance=>0.05,
 tolType=>"absolute"
);

$showPartialCorrectAnswers = 1;

# Monthly Payments
$y = random(500,3000,50);

# Interest Rate
$rate = random(3.5,8.5,.5);
$r = $rate/100;

@compound=('bi-weekly','monthly','semi-annually','annually','bi-monthly');
$tag = random(0,4,1);

#Months in each period
@periods = (24,12,2,1,6);
$per = $periods[$tag];

$comp = $compound[$tag];

# Length of mortgage
$years = random(5,30,5);
$n = $years*12;

Context()->texStrings;
BEGIN_TEXT
Suppose you can afford to pay at most $$y per month for a mortgage payment.  If the
maximum amortization period you can get is $years years, and you must pay $rate%
interest per year compounded $comp, what is the most expensive house you can buy?
How much interest will you have paid to the lender at the end of the loan?
```

```
$PAR
What is the monthly interest rate corresponding to the effective annual rate?
$BR \(r_m = \) \{ans_rule(10)\}

$PAR
What is the price of the most expensive house you can buy? \{ans_rule(10)\}

$PAR
What is the total interest paid? \{ans_rule(10)\}

$PAR

END_TEXT
Context()->normalStrings;

$ear = (1+$r/$per)**($per);
$ra = ($ear-1)*100;
$goodear = sprintf("%0.2f",$ra);

$rm = $ear**(1/12)-1;
$ans1 = $rm*100;

$p =  -$y*((1+$rm)**(-$n)-1)/$rm;
$p1 = sprintf("%0.2f",$p);

$ti = $n*$y-$p;
$ti1 = sprintf("%0.2f",$ti);

$a = sprintf("%.04f",$rm);
$month = Compute("$a*100");


 ANS(Compute($month)->cmp(tolType=>'absolute',
            tolerance=>.005) );


 ANS(Compute($p1)->cmp(tolType=>'absolute',
            tolerance=>1) );
 ANS(Compute($ti1)->cmp(tolType=>'absolute',
            tolerance=>1) );

Context()->texStrings;
BEGIN_TEXT
  \{Panic::Button(label => "Click here if you cannot come up with a monthly rate ",
penalty => .25)\}
   (you will lose 25% of your points if you do)
END_TEXT
Context()->normalStrings;
```

```
 if ($panicked) {
      BEGIN_TEXT
       $PAR $BBOLD Hint: $EBOLD The monthly rate is $month%.
      END_TEXT
    }

# this allows the student to get part marks and see which answers they got right
install_problem_grader(~~&avg_problem_grader);

Panic::GradeWithPenalty;

if($comp eq 'annually')
{
Context()->texStrings;
SOLUTION(EV3(<<'END_SOLUTION') );
$PAR SOLUTION $PAR
We don't need to calculate an EAR.
$BR We know that \(r_A = (1+r_m)^{12} - 1\)
$BR Therefore: \(r_m = (r_A+1)^{\frac{1}{12}}-1\), and the monthly rate is $month%.

$PAR
From the equation for mortgage and loan calculations we know that:
$BR \(P = -Y\frac{(1+r_m)^{-N}-1}{r_m}\),
$BR \(= -$y\frac{(1+$a)^{-$n}-1}{$a}\),
$BR \(=  $p1.\)
$PAR Therefore, the most expensive home you can buy is $$p1.

$PAR
To find the total interest paid you need to calculate the total value of your payments
and subtract the initial value of loan.
$BR \(Interest = N(Y) - P\),
$BR \(= $n ($y) -$p1\),
$BR\(= $ti1.\)

END_SOLUTION
Context()->normalStrings;
}

elsif($comp eq 'monthly')
{
Context()->texStrings;
SOLUTION(EV3(<<'END_SOLUTION') );
$PAR SOLUTION $PAR
We know that the monthly interest rate is just \(\frac{$rate}{12} = $month%\).

$PAR
From the equation for mortgage and loan calculations we know that:
$BR \(P = -Y\frac{(1+r_m)^{-N}-1}{r_m}\),
$BR \(= -$y\frac{(1+$a)^{-$n}-1}{$a}\),
$BR \(=  $p1.\)
$PAR Therefore, the most expensive home you can buy is $$p1.

$PAR
To find the total interest paid you need to calculate the total value of your payments
and subtract the initial value of loan.
$BR \(Interest = N(Y) - P\),
$BR \(= $n ($y) -$p1\),
$BR\(= $ti1.\)

END_SOLUTION
Context()->normalStrings;
}

else
{
Context()->texStrings;
SOLUTION(EV3(<<'END_SOLUTION') );
$PAR SOLUTION $PAR
First we need to calculate an EAR.
$BR Since there are $per compounding periods per year we know that \(EAR =
((1+\frac{$r}{$per})^{$per}-1)100%\).
$BR \(EAR = $goodear%.\)

$PAR Now we need to calculate the corresponding monthly rate.  We know that \(r_A =
(1+r_m)^{12}\)
Therefore: \(r_m = (r_A)^{\frac{1}{12}}-1\), and the monthly rate is $month%.

$PAR
From the equation for mortgage and loan calculations we know that:
$BR \(P = -Y\frac{(1+r_m)^{-N}-1}{r_m}\),
$BR \(= -$y\frac{(1+$a)^{-$n}-1}{$a}\),
$BR \(=  $p1.\)
$PAR Therefore, the most expensive home you can buy is $p1 dollars.

$PAR
To find the total interest paid you need to calculate the total value of your payments
and subtract the initial value of loan.
$BR \(Interest = N(Y) - P\),
$BR \(= $n ($y) -$p1\),
$BR\(= $ti1.\)
END_SOLUTION
Context()->normalStrings;
}

ENDDOCUMENT();
```

Definition of a Derivative 1

This question forces the student to do the question in steps and go through the procedure of computing the derivative by the definition

```
DOCUMENT();
loadMacros(
"PGstandard.pl",
"MathObjects.pl",
);

$showPartialCorrectAnswers = 1;

Context("Numeric");
Context()->variables->add(h=>'Real');

# Define the function of which you want the student to take the derivative
$a = random(1,9,1);
$b = random(1,9,1);
if($a == $b) {
        $b = $b+1;
                }
$f = Formula("(x+$a)/(x+$b)");
$df = $f->D;
$bma = $b-$a;

$ans1 = Formula("($b-$a)h/((x+h+$b)(x+$b))");


TEXT(beginproblem() );
Context()->texStrings;
BEGIN_TEXT
In this question you will use the definition of a derivative as a limit to find the derivative
of \(f(x) = $f \).
$PAR
\( \displaystyle \frac{f(x+h)-f(x)}{h} = \) \{ans_rule(40)\}
$PAR
Using your answer from the previous question:
$BR
\(f'(x) = \) \{ans_rule(40)\}
END_TEXT

Context()->normalStrings;
ANS($ans1 -> cmp()  );
ANS($df -> cmp() );

$apb = $a+$b;
$ab = $a*$b;

# You will have to change most of the solution
```

```
Context()->texStrings;
SOLUTION(EV3(<<'END_SOLUTION'));
$PAR SOLUTION $PAR
\(  \displaystyle \frac{f(x+h)-f(x)}{h}\)
$BR \(\displaystyle =\frac{x+h+$a}{x+h+$b}-\frac{x+$a}{x+$b}\)
$BR \(\displaystyle =\frac{(x+h+$a)(x+$b)-(x+$a)(x+$b+h)}{(x+$b+h)(x+b)}\)
$BR \(\displaystyle =\frac{x^2+$apb x +hx +$b h +$ab-x^2-$apb x -hx-$a h -
$ab}{(x+$b+h)(x+b)}\)
$BR \(\displaystyle = $ans1\).

$PAR
$BR \(\displaystyle f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}\)
$BR \(= \displaystyle \lim_{h \to 0} \frac{1}{h} $ans1\)
$BR \(= \displaystyle \lim_{h \to 0} \frac{$bma}{(x+h+$b)(x+$b)}\)
$PAR \(= \frac{$bma}{(x+$b)^2}\)
END_SOLUTION
Context()->normalStrings;

ENDDOCUMENT();
```

Definition of a Derivative 2

I found this question in a University of Toronto problem set that I thought offered an interesting solution to make sure the student was using the definition of the derivative

```
# This question is from U of T MAT 137Y 2008 term test solutions
DOCUMENT();

loadMacros(
  "PGstandard.pl”,
  "MathObjects.pl",
  "problemPanic.pl",

);

$showPartialCorrectAnswers = 1;

Context("Numeric");

Context()->texStrings;
BEGIN_TEXT
Suppose \(f\) is a function which satisfies the properties:

$PAR

\(f(x+y) = f(x) + f(y) + x^2y +xy^2\)

$PAR  for all real numbers \(x\) and \(y\), and

$PAR\(\displaystyle \lim_{x \to 0} \frac{f(x)}{x} = 1 \).

$PAR \(f'(x) = \) \{ans_rule(40)\}

$PAR

END_TEXT
Context()->normalStrings;

ANS(Formula("1+x^2")->cmp() );

Context()->texStrings;
BEGIN_TEXT
  \{Panic::Button(label => "HINT:", penalty => .25)\}
    (you will lose 25% of your points if you do)
END_TEXT
Context()->normalStrings;


 if ($panicked) {
     BEGIN_TEXT
     $PAR $BBOLD Hint: $EBOLD Use the definition of a derivative: \(f'(x) = \displaystyle
\lim_{h \to 0} \frac{f(x+h)-f(x)}{h} \).  Notice how you can rearrange the formula for
\(f(x+y)\) as: \(f(x+y)-f(x) = f(y) + x^2y+xy^2\).
    END_TEXT
  }

# this allows the student to get part marks and see which answers they got right
install_problem_grader(~~&avg_problem_grader);

Panic::GradeWithPenalty;

BEGIN_SOLUTION
$PAR SOLUTION $PAR
Use the definition of a derivative: \(f'(x) = \displaystyle \lim_{h \to 0} \frac{f(x+h)-
f(x)}{h} \) and the formula for \(f(x+y)\).

$PAR \(f'(x) = \displaystyle \lim_{h \to 0} \frac{f(x+h)-f(x)}{h} \)
$PAR \( = \displaystyle \lim_{h \to 0} \frac{f(x)+f(h)+x^2h+xh^2-f(x)}{h} \)
$PAR \( = \displaystyle \lim_{h \to 0} \frac{f(h)+x^2h+xh^2}{h} \)
$PAR \( = \displaystyle \lim_{h \to 0} \left(\frac{f(h)}{h}+x^2+xh\right) \)
$PAR \( = 1+x^2\)

$PAR Therefore \(f'(x) = 1+x^2\)

END_SOLUTION

ENDDOCUMENT();
```

Index of Economic Questions

The economic questions, except the first two on the list, are optimization questions. I have included the math techniques needed to answer them. There are more economic questions than are on this list, but they are already classified by topic

Economic Topic	Math Topics	Source File
Elasticity	Exponential differentiation Product/chain rules	setElasticity/computating.pg
Elasticity	Exponential differentiation Product/chain rules	setElasticity/compute2.pg
Marginal Utility	Intro to rates of change: pre-differentiation	setEconomics/tsimpleUtility.pg
Marginal Costs	Intro to rates of change: pre-differentiation	setEconomics/tsimplecosts.pg
Labour Demand Curve	Logarithmic Differentiation	setEconomics/tLabourDemand.pg
Laffer Curve	Product/ quotient/ chain rules	setEconomics/tlaffer.pg
Elasticity	Product/ quotient/ chain rules	setElasticity/constantDemand.pg
Elasticity	Implicit differentiation	setElasticity/constantElasticity
Elasticity	Implicit differentiation	setElasticity/optimization.pg
Elasticity	Implicit differentiation	setElasticity/implicityElasticity.pg
Indifference Curves	Implicit differentiation	setEconomics/tindifference1.pg setEconomics/tindifference2.pg
Production Function	Implicit differentiation	setEconomics/tproductionFunction.pg
Optimising utility Interpretation of LM	Lagrenge Multipliers (LM)	setmath105/LMindifference.pg
Optimising utility	LM	setmath105/workPlay.pg
Optimising utility Interpretation of LM	LM	setmath105/LMInterpretation.pg

Helpful Links

WeBWorK Wiki:	http://webwork.maa.org/wiki/Main_Page
Sample Questions:	http://webwork.maa.org/wiki/Category:Sample_Problems
Intro to Perl	http://perldoc.perl.org/perlintro.html
Math Objects	http://webwork.maa.org/wiki/Introduction_to_MathObjects
Problem Techniques	http://webwork.maa.org/wiki/Category:Problem_Techniques
WeBWorK Forum	http://webwork.maa.org/moodle/login/index.php
Basic Training	http://webwork.maa.org/wiki/TrainingAuthors