WeBWorK Accuracy and Answer Checking

1. The instructors set the tolerances for checking the answers. This is usually set to an absolution error of 0.001. This means that if x is the correct answer and y is the student's answer, then the answer is counted correct if $|x-y|<0.001$.

   Now, for a bit more detail and discussion of the pros and cons of various error measures. You may continue to learn about the two options and why the instructor selected the absolute error measure. Read a bit further to learn about some of the interesting problems they encountered in creating your problems.

2. There are two type of error tolerances: Let x be the correct answer and y be the student's answer.
   a) absolute error of e means the answer will be counted correct if $|x-y|<e$
   b) relative error of e means the answer will be counted correct if $|\frac{x-y}{x}|<e$
3. For a value of e, the absolute error remains the same regardless of the value of x. Examples: for $e=0.001$, the answer must always be within $x\pm 0.001$. If x is very large, e.g., $x=10000$ then the correct answer would be within $9999.999<y<10000.001$. If x is very small e.g., $x=0.001$ then the correct answer would be within $0<y<0.002$. These extreme cases illustrate the problems with always using absolute error.
4. For a value of e, the relative error changes with the value of x. Examples: for $e=0.001$, and $x=1$, the answer must be within $x\pm 0.001$, just like the absolute error. However, if x is very large, e.g., $x=10000$ then the correct answer would be within $9990<y<10010$. If x is very small e.g., $x=0.001$ then the correct answer would be within $0.000999<y<0.001001$. The relative error seems more appropriate for most answers where we know the range of the answers.
5. For problems where the range of the answers are uncertain it is usually better to use absolute error. If the value x is computed, the error tolerance must include the computational error. If we ask for $x=\cos(\pi/4)$, the correct answer is $x=\frac{\sqrt{2}}{2}$. If a calculator is allowed, we may accept values within 0.001. We know the range of the answer and relative and absolute errors are about the same. However, if we randomly choose angles in $(-\pi,\pi]$ and let you use a calculator, a random angle that yields a value near zero, will require many more decimal places to be counted correct, if we use relative error.
6. Checking functions: Let x(t) be the correct answer and y(t) be the student's answer. Webwork checks the equivalence of the two functions by sampling each of them at random test points, or sometimes test points defined by the instructor. If the error for each evaluation is within the tolerance, the answer is counted correct. Since our functions often include sinusoids that might be sampled to give a value near zero, we usually7 specify an absolute tolerance of 0.001.

Some interesting points and examples for the really curious.

7. The default interval for generating is [0,1). This is reasonable enough "in general." For problems that are valid for both positive and negative input values, we need to be sure to ask change the test interval to include negative numbers. For cases, where it in important for the student to specify the range of the answer, e.g., include the unit step, u(t), to indicate the answer is good only for non-negative values, it is important for us to include test points in the negative range.
8. For many realistic problem with differential equations, the solution is a decaying sinusoid. The rate of decay may be in the millisecond or microsecond range. Testing the functions in the range [0,1) would result in getting values near zero and would usually show wrong answers as correct since they produced small values. In these cases, it is important for the problem designer to consider the random values and compute a reasonable range for the random test points.
9. When creating problems in Webwork, we check the problem by a) solving it and typing the correct answer, b) repeating the check using a different random number seed, c) typing in an obviously incorrect answer. However, while there may be only one correct answer, there are an infinite number of incorrect ones. Sometimes creative wrong answers lead the problem designer to develop more rigorous methods of generating test points.

   An example of this occurred this semester for the aliasing problem, HW1, Problem4. The designer computed the aliased signals and specified a range for random test points. However, students reported having answers counted correct that were in error. On close examination, we found that Webworks random sampling scheme was basically a sampling at a lower frequency that sampled the high frequency wrong answer at points that aliased to the frequencies of the correct lower frequency aliased signal. We had been done in by the very effect we had been teaching. The problem was corrected by forcing Webwork to sample at much higher frequencies than the range expected in the wrong answers.