```
(*Part 1a:*)
(*This function will take an equation
 (in the form of a function) and a tolerance level as inputs,
perform the bisection technique on the equation for the given tolerance and output
 a list with values we need to put in the columns of the table we will create.*)
bisectionTechnique[eq_, ϵ_] := (
  count = 0; a = 1; b = 3;
  While[count ≤ 1000, p = (a + (b - a) / 2);
   If[(eq[p] == 0) || (((b - a) / 2) < ϵ), myRoot = p;
    Break[]];
   count++;
   If[eq[p] * eq[a] > 0, a = p, b = p]];
  computerRoot = r /. FindRoot[{eq[r]}, {r, 2}]; Return[{"y =" eq[x], ϵ, myRoot,
    computerRoot, Abs[myRoot - computerRoot], Ceiling[(Log2[2 / ϵ] - 1)], count}]
 )
```

```
In[•]:= (*These three lines translate the equations into
 functions. We want to rewrite the equations as functions so that
 the bisectionTechnique function can take the equations as inputs.*)
equation1[x_] := x^5 - 2 x^4 - 1
equation2[x_] := x^3 - x - 1
equation3[x_] := x^2 + 2 x - 5
```

```
In[•]:= (*This cell makes a grid for the three
 equations for the four different tolerance levels.*)
Grid[{bisectionTechnique[equation1, 0.1], bisectionTechnique[equation1, 0.05],
  bisectionTechnique[equation1, 0.01], bisectionTechnique[equation1, 0.001],
  bisectionTechnique[equation2, 0.1], bisectionTechnique[equation2, 0.05],
  bisectionTechnique[equation2, 0.01], bisectionTechnique[equation2, 0.001],
  bisectionTechnique[equation3, 0.1], bisectionTechnique[equation3, 0.05],
  bisectionTechnique[equation3, 0.01], bisectionTechnique[equation3, 0.001]}, Frame → All]
```

```
(*This cell aligns everything in the grid.*)
Insert[%82, Alignment → Left, 2]
```

```
(*This cell titles the columns.*)
ReplacePart[%83, 1 → Prepend[First[%83], {"Polynomial", "Error Tolerance (ϵ)",
    "Root from Bisection Technique", "Actual Root", "Actual Error",
    "Maximum Steps Required", "Number of Steps Bisection Used in Technique "}]]
```

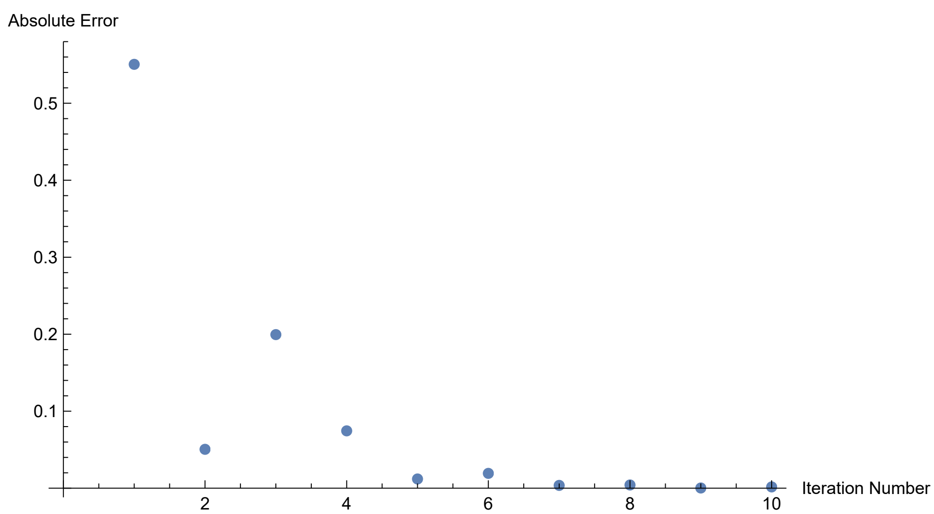
Out[∘]=

Polynomial	Error Tolerance (ϵ)	Root from Bisection Technique	Actual Root	Actual Error	Maximum Steps Required	Number of Steps Bisection Used in Technique
$y = (-1 - 2x^4 + x^5)$	0.1	$\frac{33}{16}$	2.05597	0.0065326	4	4
$y = (-1 - 2x^4 + x^5)$	0.05	$\frac{65}{32}$	2.05597	0.0247174	5	5
$y = (-1 - 2x^4 + x^5)$	0.01	$\frac{263}{128}$	2.05597	0.0012799	7	7
$y = (-1 - 2x^4 + x^5)$	0.001	$\frac{2105}{1024}$	2.05597	0.000303334	10	10
$y = (-1 - x + x^3)$	0.1	$\frac{21}{16}$	1.32472	0.012218	4	4
$y = (-1 - x + x^3)$	0.05	$\frac{43}{32}$	1.32472	0.019032	5	5
$y = (-1 - x + x^3)$	0.01	$\frac{169}{128}$	1.32472	0.00440546	7	7
$y = (-1 - x + x^3)$	0.001	$\frac{1357}{1024}$	1.32472	0.000477355	10	10
$y = (-5 + 2x + x^2)$	0.1	$\frac{23}{16}$	1.44949	0.0119897	4	4
$y = (-5 + 2x + x^2)$	0.05	$\frac{47}{32}$	1.44949	0.0192603	5	5
$y = (-5 + 2x + x^2)$	0.01	$\frac{185}{128}$	1.44949	0.00417724	7	7
$y = (-5 + 2x + x^2)$	0.001	$\frac{1485}{1024}$	1.44949	0.00070557	10	10

In[ ]:=
```
(*Part 1b:*)
(*This function will take an iteration number as an input,
apply that many iterations of the bisection technique to the 3rd equation,
and output the absolute error after that many iterations.*)
error3rdEquation[iteration_] := (
  computerRoot = r /. FindRoot[{equation3[r]}, {r, 2}];
  count = 0; a = 1; b = 3;
  While[count ≤ 1000, p = (a + (b - a) / 2);
   If[count == iteration, Break[]];
   myRoot = p;
   count++;
   If[equation3[p] * equation3[a] > 0, a = p, b = p]];
  Return[Abs[myRoot - computerRoot]]
 )

(*This cell will first create a table with iterations numbers 1 through 10
 and the error3rdEquation output for each of the iteration numbers. Then,
it will plot the table and label each of the axes of the plot. I chose
 iteration numbers 1 through 10 because this will show the phenomenon where
 absolute error increases after an increase in the iteration number.*)
errorTable = Table[{i, error3rdEquation[i]}, {i, 1, 10}];
ListPlot[errorTable, PlotMarkers → {Automatic, 6},
 AxesLabel → {"Iteration Number", "Absolute Error"}]
```

Out[ ]=



Explanation: After certain iterations, the absolute error will increase from the previous iteration. This will happen when we are working inside an interval, in which the actual root is closer to our last guess than the actual root is to the midpoint of that interval. Suppose: a_i = (our last guess), we are working inside the interval $[a_i, b_i]$, the actual root, R, is closer to a_i than R is to the midpoint of $[a_i, b_i]$. Our next guess will throw us to the midpoint of $[a_i, b_i]$ which will take us farther away from R than a_i was.