Babylonia Approximations of Square Roots

**Historical Context:**
- When: ca. 1700 B.C.
- Where: Babylon
- Who: Mathematicians
- Mathematics focus: Demonstrate use of iteration to approximate square roots.

**Suggested Readings:**
- History of square root algorithms:
- Babylonian calculation of $\sqrt{2}$:
  [http://www.math.ubc.ca/people/faculty/cass/Euclid/ybc/analysis.html](http://www.math.ubc.ca/people/faculty/cass/Euclid/ybc/analysis.html)
  or
- The Babylonian’s use of sexagesimal notation:
  [http://www.spirasolaris.ca/sbb1sup1.html](http://www.spirasolaris.ca/sbb1sup1.html)
- NCTM’s *Historical Topics for the Mathematics Classroom* (1969):
  “Incommensurables and irrational numbers” (pp. 70-72) and “Radical symbol” (pp. 147-148)
- Key search words/phrases: Babylon mathematics, square roots, iteration, approximation, Heron’s Method, sequence, convergence

**Problem to Explore:**
Find the square root of a positive number $m$.

**Why This Problem is Important:**
- The Babylonians iterative approach to extracting approximations to square roots is visual, intuitive, and powerful.
- The Babylonians extraction of square roots is a “particular case” of Newton’s method for approximating zeroes of a function.
Problem Solving Experiences:
The Babylonians often are credited with suggesting the first technique for extracting approximations of square roots, which was needed for their solution of quadratic equations. Their assumed technique relies on iterations involving “dividing and averaging.”

The Babylonian algorithm: Suppose you want to find $\sqrt{m}$, where $m$ is a positive number. Let $a_1$ be a first approximation to $\sqrt{m}$. For $n = 1, 2, 3, \ldots$, calculate $a_{n+1} = \frac{1}{2}(a_n + \frac{m}{a_n})$, producing a sequence $a_1, a_2, a_3, \ldots$ of approximations to $\sqrt{m}$.

1. Starting with $a_1 = 1$, use the Babylonian algorithm to find the $\sqrt{2}$, keeping track of both $a_n$ and $2/a_n$ for each step in the iterative algorithm.
2. A number line can be built to visually represent your calculations in Problem #1. Carefully graph $a_1 = 1$, $2/a_1 = 2$, and $\sqrt{2} = 1.414213562\ldots$, then add the values for $a_2$ and $2/a_2$, etc. First, what is always true about the values $a_n$ and $2/a_n$ for each value of $n$? Any idea why this occurs? Second, what is happening with each new insertion of the values $a_n$ and $2/a_n$ relative to the actual value of $\sqrt{2}$? Again, any idea why this occurs?

To explore the Babylonian algorithm further, assume that $a_1$ is a first approximation but not equal to $\sqrt{m}$. If $a_1$ is multiplied by $m/a_1$, the result is $m$. Since both $a_1$ and $m/a_1$ are approximations for $\sqrt{m}$, one would expect intuitively that their arithmetic mean [i.e. $a_2 = \frac{1}{2}(a_1 + \frac{m}{a_1})$] should be an even better approximation.

3. Test the Babylonian algorithm further by implementing it via a spreadsheet, computer program, or calculator program. Find the $\sqrt{15}$ by trying a wide range of first non-zero approximations (e.g. $a_1 = 1, a_1 = 16, a_1 = 1000, a_1 = 0.0001, a_1 = -5$, etc.). What observations can you make from your results?

Extension and Reflection Questions:

Extension 1: Reconsider the Babylonian algorithm. If $a_n$ is an approximation for $\sqrt{m}$, prove that the arithmetic mean $(1/2)(a_n + m/a_n)$ is an even better approximation. Hint: After building a visual model of the relative locations of $\sqrt{m}$, $a_n$, $m/a_n$, $a_{n+1} = (1/2)(a_n + m/a_n)$, and $m/a_{n+1} = m/[(1/2)(a_n + m/a_n)]$, try to show that (1) $\sqrt{m}$ is always between the approximation values of $a_n$ and $m/a_n$, and (2) the distance between iteration values $a_n$ and $m/a_n$ is always greater than the distance between the subsequent iteration values of $a_{n+1} = (1/2)(n+m/n)$ and $m/a_{n+1} = m/[(1/2)(n+m/n)]$.

Extension 2: Use the information in Extension 2 to first prove that $a_n$ and $m/a_n$ converge to a common limit value $A$ and then prove that $A = \sqrt{m}$.
Extension 3: What is the rate of convergence of the Babylonian algorithm in its extraction of $\sqrt[n]{m}$?

Open-ended Exploration: The Babylonian algorithm only extracted square roots for positive numbers. Can you make adjustments that also will extract cube roots of positive numbers? Nth roots of positive numbers?