THESIS I

3 Historical Methods of Calculating Square Roots

Presentation by Yael Green

Computer Science | 2024

Touro University

- Studied since 1800 BCE
- Critical in calculations for:
  - Financial growth
  - Volume/positioning of objects
  - Statistical analysis of data
  - Etc.

- Can't be written as a ratio of two integers
- Infinite non-repeating sequence of numbers
- Impossible to ever completely define irrational number
- Common irrational number: pi
- Typically approximate pi to 22/7, which is rational (3.142857142857...)

- Goal then is to find a close approximation of irrational number
- Can use methods to keep calculating further decimal places
- Method is efficient when it accurately computes large number of decimals with each iteration

- Records of approximations of the square root of 2 dating between 1800 and 1600 BCE
- Documentation from ancient Egyptian, Indian, Greek, Persian, and Chinese civilizations over the next several thousand years

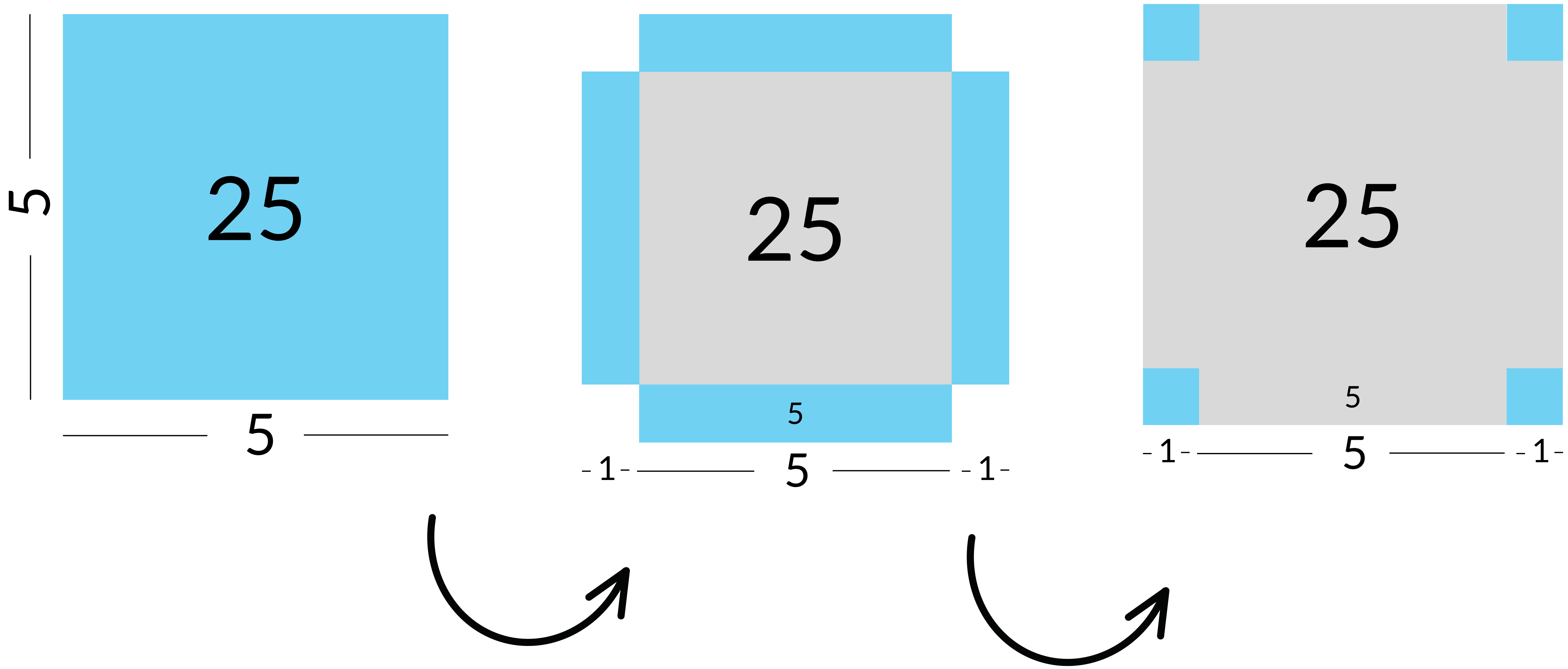
Three methods:

Heron/Newton's method
60 CE / 1687 The Principia

Talmudic method
300 - 600 CE

Bakhshali method
224 - 993 CE

Square root of 50:



Iteration #	Beginning remainder	Divide remainder by 5	Width of current square	Find length of remainder along width of current estimate	Add length to each side	Remainder
1	25	5	5	1	7	1
2	1	0.2	7	0.0285714	7.057143	0.196735
3	0.196735	0.039347	7.057143	0.0055755	7.068294	0.039223
4	0.039223	0.007845	7.068294	0.0011098	7.070513	0.00784
5	0.00784	0.001568	7.070513	0.0002218	7.070957	0.001568

Actual square root: 7.0710678...

Successive approximations: $x = \frac{1}{2}(x + \frac{S}{x})$

Using approximation from above (8*8 = 64):

Iteration #	Beginning approximation	Divide original number by approximation	Add two previous	Divide by 2
1	8	6.25	14.25	7.125
2	7.125	7.01754386	14.14254	7.07127193
3	7.07127193	7.0708637	14.14214	7.071067815
4	7.07106781	7.07106781	14.14214	7.071067812
5	7.07106781	7.07106781	14.14214	7.071067812

[1:] In the case of a non-square [number], subtract the nearest square number; divide the remainder by twice [the root of that number]. [2:] Half the square of that [that is, the fraction just obtained] is divided by the sum of the root and the fraction and subtract [from the sum]. [3:] [The non-square number is] less [than the square of the approximation] by the square [of the last term].

$$a_n = \frac{q - x_n^2}{2x_n} \quad \text{(sentence \#1 above)}$$

$$x_{n+1} = x_n + a_n - \frac{a_n^2}{2(x_n + a_n)} \quad \text{(sentence \#2 above)}$$

x = approximation of sqrt	8
p = 50 - x squared	-14
a = p / 2x	-0.875
ax = a + x	7.125
as = a squared	0.765625
ad = as / 2ax	0.05372807
ax - ad	7.07127193

Heron vs. Bakhshali

Iteration #	Beginning approximation	Divide original number by approximation	Add two previous	Divide by 2
1	8	6.25	14.25	7.125
2	7.125	7.01754386	14.14254	7.07127193
3	7.07127193	7.0708637	14.14214	7.071067815
4	7.07106781	7.07106781	14.14214	7.071067812
5	7.07106781	7.07106781	14.14214	7.071067812

x = approximation of sqrt	8
p = 50 - x squared	-14
a = p / 2x	-0.875
ax = a + x	7.125
as = a squared	0.765625
ad = as / 2ax	0.05372807
ax - ad	7.07127193

as = a squared	4.16629E-08
ad = as / 2ax	2.94601E-09
ax - ad	7.071067812

```java
public double accuracyInN(int num, int iter) {
    double dec = getDiffSqrt(num, iter);
    int ind = 0;
    double ret = 0;

    if(dec != 0) {
        String decStr = getStrAfterDec(dec);

        int ch = decStr.charAt(ind);

        while(ch == '0') {
            ind++;
            ch = decStr.charAt(ind);
        }
        ind--;
        ret = 1.0 / pow( base: 10, ind);
    }

    return ret;
}
```

Finds how many digits of accuracy in N iterations

INTRODUCTION

IRRATIONAL #S

HISTORY

TALMUDIC

NEWTON/HERON

BAKHSHALI

EFFICIENCY

CONCLUSION

Finds how many iterations to achieve N degrees of accuracy

```java
public int iterationsToN(int num, double degree) {
    double curr = findClosestSqrtBelow(num);
    double next = feedNextApprox(num, curr);
    int iter = 1;

    while(Math.abs(curr - next) >= degree) {
        curr = next;
        next = feedNextApprox(num, curr);
        iter++;
    }

    return iter;
}
```

Finding square root of 65	Talmudic	Heron's
In 2 iterations	0.1	0.0000001
In 3 iterations	0.01	Absolute
Iterations to accuracy .1	1	1
Iterations to accuracy .001	4	2

- Bakhshali -> Heron -> Talmudic
- Advantages to each

THANK YOU

any questions?

REFERENCES

- Talmud Bavli, Eiruvin 23b
- Newton, Isaac, and Percival Frost. Principia. Macmillan and Co., 1863.
- David H. Bailey, and Jonathan M. Borwein. “Ancient Indian Square Roots: An Exercise in Forensic Paleo-Mathematics.” The American Mathematical Monthly, vol. 119, no. 8, 2012, pp. 646–57. JSTOR, https://doi.org/10.4169/amer.math.monthly.119.08.646. Accessed 18 Apr. 2024.