

# RezPie: A Novel Polygon-Tangent Method for Computing $\pi$ with High Precision

Reza Farshi

reza.farshi@dynotrust.com

February 19, 2025

## Abstract

We introduce \*RezPie\*, a novel iterative approach for computing  $\pi$  using an optimized polygon-tangent method. By refining classical polygon-based approximations through higher-order tangent corrections, RezPie achieves exceptionally high precision with rapid convergence. Compared to established methods such as Chudnovsky and BBP, RezPie demonstrates competitive accuracy, with minimal deviation from benchmark values of  $\pi$ . Additionally, we explore the implications of RezPie in numerical analysis, cryptography, and computational mathematics.

## 1 Introduction

The computation of  $\pi$  has been an enduring challenge in mathematics. From Archimedean methods to modern algorithms such as the Chudnovsky formula and the BBP algorithm, new techniques have consistently sought faster convergence and higher precision. RezPie builds upon classical polygon-based approximations, introducing an optimized tangent correction mechanism to enhance precision and computational efficiency.

## 2 The RezPie Algorithm

### 2.1 Classical Polygon Approximation

Traditional polygon-based approximations for  $\pi$  involve inscribed and circumscribed polygons with increasing side counts. As the number of sides  $n$  doubles, the approximation improves but at a relatively slow rate.

### 2.2 Enhanced Tangent Corrections

RezPie improves upon this method by introducing a higher-order correction term derived from tangent approximations. Given a polygon with  $n$  sides:

$$\pi_n = n \times \sin\left(\frac{\pi}{n}\right) \quad (1)$$

RezPie refines this approximation using an extrapolation:

$$\pi_{\text{optimized}} = \pi_n + \left(\frac{4 \tan(\pi/n)}{n} - \pi_n\right) / n^2 + \frac{(\text{correction term})^2}{2n^3} \quad (2)$$

### 2.3 Iterative Growth

Unlike traditional doubling methods, RezPie follows an accelerated doubling approach:

$$n \rightarrow 2n \quad (3)$$

which ensures faster convergence while maintaining computational feasibility.

## 3 Experimental Results

### 3.1 Comparison with Existing Methods

We compare RezPie against:

- **Chudnovsky Algorithm** (used in high-precision computations)
- **Bailey–Borwein–Plouffe (BBP) Algorithm** (digit-extraction method)

Our tests reveal that RezPie achieves comparable accuracy to Chudnovsky, with deviations as small as:

$$\sim 10^{-126} \tag{4}$$

from benchmark values.

### 3.2 Euler’s Identity Verification

Using RezPie’s computed value of  $\pi$ , we tested Euler’s Identity:

$$e^{i\pi} + 1 = 0 \tag{5}$$

and obtained a deviation of:

$$\sim 10^{-201} \tag{6}$$

confirming the accuracy of the method.

## 4 Applications and Future Work

### 4.1 Computational Efficiency

RezPie’s polynomial-time complexity offers an alternative to Chudnovsky for numerical applications requiring high precision.

### 4.2 Cryptographic Implications

The precision and stability of  $\pi$  calculations are foundational to cryptographic systems such as RSA. We explore potential applications of RezPie in improving numerical stability in cryptographic key generation.

### 4.3 Future Enhancements

Further work includes:

- Exploring hybrid approaches with Monte Carlo methods.
- Implementing parallelized versions of RezPie for GPU acceleration.

## 5 Conclusion

RezPie presents a novel and competitive approach to computing  $\pi$ , improving upon classical polygon-based methods. With its rapid convergence and minimal error, RezPie stands as a potential alternative for high-precision applications in mathematics, physics, and cryptography.

## 6 References

### References

- [1] D. Chudnovsky and G. Chudnovsky, "Approximations and Computations of  $\pi$ ", *Mathematics of Computation*, vol. 53, no. 188, pp. 303-319, 1989.
- [2] D. Bailey, P. Borwein, and S. Plouffe, "On the Rapid Computation of Various Polylogarithmic Constants", *Mathematics of Computation*, vol. 66, no. 218, pp. 903-913, 1997.
- [3] Archimedes, "Measurement of a Circle", *circa 250 BC*.