A Quadratically Convergent Method for Computing Euler's Number

EMERSON MILLER, DOUGLAS SALANE
John Jay College
The City University of New York
524 West 59th Street, NY, NY, 10019
USA

Abstract: - We describe the use of Newton's method to compute a high precision approximation to e. This work provides a brief introduction to the the history of computing elementary transcendental functions and numbers, presents a novel method for computing e, examines the quadratic convergence of Newton's method in this application, and makes use of multi-precision arithmetic available in Python to compute e to any desired precision.

Key-words: - computing Euler's number, Newton's method, quadratic convergence

Received: April 6, 2024. Revised: August 7, 2024. Accepted: September 13, 2024. Published: October 31, 2024.

1. Introduction

EULER'S number e is one of the most important irrational constants in mathematics. Used to describe growth when the growth rate is proportional to population size, most are familiar with its application in modelling population growth and radioactive decay. An excellent history of the early uses and development of e appears in [1].

Although associated with Euler, Jacob Bernoulli stumbled upon the familiar sequence approximation to e while investigating the growth of savings where interest is compounded n times per year.

$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$$

Some decades later Euler in investigating series approximations to functions showed e was the sum of the infinite series

$$e = \sum_{i}^{n}\frac{1}{n!}$$

There is a rich history of methods to compute e and as well as other transcendental numbers and functions. The goal is to compute high precision approximations with the least amount of computation. For e, one approach is to use binary splitting to avoid the costly number of divisions in summing the series approximation to e [8]. This technique also requires fast multiplication methods for large integers in order to gain an advantage over the straightforward evaluation of the series expansion. Another approach are the arithmetic-Geometric Mean(AGM)Methods [2]. These date back to Gauss and Legendre and have been used by authors to produce rapidly converging approximations to π and e and transcendental functions.[3] The extended abstract [4] provides a concise overview of techniques used for high precision evaluation of transcendental functions and numbers.

Once one transcendental function can be evaluated, other elementary functions and constants can be calculated from it by composition or inversion. The work we have just discussed addresses the difficult problem - finding a rapidly convergent method to approximate just one transcendental function [2] Here we assume the transcendental function $ln(x)$ is available and develop a quadratically convergent method for computing e.

The goal is to develop an iterative method for approximating e that converges quadratically so there is a doubling of the number of correct digits with each iteration. With the transcendental function $ln(x)$ available, we are able to take advantage of the quadratic convergence of Newton's method to approximate e. The use of Newton's method is often limited for computing transcendental numbers or functions since if it is applied to an algebraic function the method will converge to an algebraic result.

The motive here is instructional. We apply various techniques which will allow the reader to appreciate fundamentals such as Newton's method, quadratic convergence, techniques for computing transcendental numbers and the use of high precision arithmetic now readily available in computer languages such as Python. The method we present requires only rudimentary knowledge of calculus and allows the reader to appreciate methods for computing transcendental numbers. For other examples of computation of transcendental numbers for pedagogical purposes see [5].

The technique used here takes advantage of an interesting property of the natural log and exponential functions. If one considers the unique value x where

$$ln(x) = \frac{1}{x}$$

then $e = x^x$.

In the next section, we derive this relation. In the third section we use Newton's method to obtain an approximation to the solution and provide a quadratically convergent algorithm for computing e. In the following section we discuss the convergence properties of Newton's method in this application. The last section presents numerical results that demonstrate the method converges quadratically. In addition, we use the variable precision arithmetic package available for Python so the algorithm can compute e to any number of digits, limited only by machine memory and computation time.

2. An Expression for e

It is straightforward to show the preceding equation has a unique solution $\overline{x}$ and that $e = \overline{x}^{\overline{x}}$. Consider the function

$$f(x) = \ln(x) - \frac{1}{x}$$

The function $f(x)$ is a continuous and strictly increasing function if $x > 0$. Since $f(1)$ is less than zero and $f(x)$ is positive for x sufficiently large, for example, for $x = 3$, the function has a unique root, denoted as $\overline{x}$, in the interval $(1, 3)$. The result now follows from the definition of $f(x)$ and the laws of exponents.

Since $f(x)$ has a unique zero, denoted by $\overline{x}$, we write

$$\ln(\overline{x}) = \frac{1}{\overline{x}}$$

Now simple algebraic manipulations, the laws of exponents and the definition of $\ln(x)$ yield the following:

$$\overline{x}\ln(\overline{x}) = 1$$
$$e^{\overline{x}\ln(\overline{x})} = e$$
$$\left(e^{\ln(\overline{x})}\right)^{\overline{x}} = e$$
$$\overline{x}^{\overline{x}} = e$$

An approximation to e can now be computed by using Newton's method to approximate $\overline{x}$.

It is interesting to note that the value $\overline{x}$ is the point on the x axis where the area under the curve $y = \frac{1}{x}$ from 1 to x is given by $\frac{1}{x}$. In general, for $x > 1$, the area is given by the $\ln(x)$, which is the way the natural $\ln(x)$was derived almost a century before Euler computed the base to be e. In the next section we describe our application of Newton's method to find $\overline{x}$, the root of f.

3. Use of Newton's Method to Approximate e

The function $f(x)$ provides a quadratically convergent method for approximating e. Basically, we just apply Newton's method to find the unique root of the function $f(x)$. Since $f(1) < 0$ and $f(3) > 0$ and f is strictly increasing, the unique root must lie in interval. Moreover, since the derivative is positive, it is possible to show the root is unique and Newton's method will converge to it if started from x=1.

Recall Newton's method for finding a zero of a function has the form

$$x_{n+1} = x_n - \left(f'(x_n)\right)^{-1} f(x_n)$$

Starting the iteration sufficiently close to the root, we expect each iteration to double the number of correct digits with each approximation of the zero of $f(x)$. This is verified by the numerical results and convergence analysis in the Numerical Results section.

Newton's method to find a root of the function f can be written as

$$x_{n+1} = x_n - \left(\frac{x_n^2}{x_n + 1}\right)\left(\ln(x_n) - \frac{1}{x_n}\right)$$

In our implementation we rewrite the correction term in the above iteration so that only one division is required. Of course the preceding iteration only approximates $\overline{x}$, the root of $f(x)$. Once an approximation to $\overline{x}$ is obtained, the approximation to e can be computed. A sketch of an algorithm for computing e is as follows:

Algorithm 1 Newton's Method to Compute e

```
x ← 1
eps ← 1.0E − 500
correction ← 1
while correction > eps do
    correction ← (x^2 ln(x) − x)/(x + 1)
    x ← x − correction
    eapproximation ← x^x
end while
```

It is not necessary to compute the approximation for e at each iteration of Newton's method. When the correction is less than stopping tolerance, *eps*, we could terminate the iteration and compute the approximation for e. We compute the e approximation after each iteration simply to show how the approximation to e improves with each iteration.

Finally, if the approximate solution to $\ln(x) - \frac{1}{x} = 0$ is accurate to k digits, the approximation to e is accurate to approximately k-1 digits. This is easily shown by considering$f(x) = \epsilon$, where ϵ is the error. Also, since the Newton correction allows us to determine the accuracy of both x and the approximation to e, it is reasonable to control the Newton iteration by monitoring the correction. We discuss this in detail in the Numerical Results section.

4. Convergence Properties

Under appropriate conditions Newton's method exhibits quadratic convergence. Formally, this means that if $\overline{x}$ is a root of f, then for successive iterates of the method we have

$$| x_{n+1} - \overline{x} | < C \, | x_n - \overline{x} |^2$$

where C is some constant. The preceding inequality implies that if the iterates are sufficiently close to the root, each step of Newton's method typically will result in a doubling of the number of significant digits in the approximation. This is certainly what is observed after the second iteration in numerical results presented in the next section.

In this application of Newton's method, the function $f(x)$ satisfies the necessary conditions for quadratic convergence in a neighborhood sufficiently close to the root. The function and its first two derivatives are continuous on $[1, 3]$, the interval containing the root. In addition, $f^{'}$ is positive on the interval so$f^{'}(\overline{x}) \neq 0$, a condition necessary for quadratic convergence.

Determining a neighborhood of the root where Newton's method converges is often very difficult. Formal statements of the convergence theorems only specify that in a small enough neighborhood of the root the method converges quadratically. In this case, however, the behavior of f' and f'' provide some insight. The first derivative is positive and strictly decreasing on [1,3]. In addition, the second derivative is negative and strictly increasing on [1,3]. Thus we can determine bounds on the magnitudes of the derivatives over the interval by evaluating them at the endpoints.

Quadratic Convergence of Newton's method depends on finding a delta neighborhood of the root r such that if

$$c(\delta) = \frac{1}{2} \frac{\max\limits_{|x-r|\leq\delta} |f''(x)|}{\min\limits_{|x-r|\leq\delta} |f'(x|}$$

$\delta c(\delta) < 1$ [6]. Starting the iteration from a point within such a neighborhood of the root guarantees the iterates remain in the neighborhood and convergence is quadratic. The continuity of the derivatives guarantee such a neighborhood exists.

The root lies in the interval [1,3]. Evaluating the derivatives at these endpoints, however, shows they are too far from the solution to be useful in a convergence analysis. Nonetheless, after one step of Newton's method and an examination of the next correction we are led to consider the interval [1.5,2]. Note that $f(1.5) < 0$ and $f(2) > 0$ so the root lies in [1.5 ,2]. The first two Newton corrections indicate the root is close to 1.73. The ratio of the maximum of f'' and the minimum of f' , computed from the interval endpoints, is less than 1 on [1.5,2]. Choosing $\delta < .2$ guarantees the neighborhood of the root is in the interval and $\delta c(\delta) < 1$. Thus, after the first step of Newton's method we are assured the iterates converge quadratically, precisely what is observed in the Numerical Results section.

It is well known that Newton's method tends to be self-correcting in the sense that if started at a point where quadratic convergence is not observed in the first few iterations, it often enters a region where convergence is quadratic. In this case the first step of Newton's method results in a relatively large step that gets close to the root. The preceding analysis and numerical results in the next section show the algorithm after the first step is close enough to the root to converge quadratically.

5. Numerical Results

We implemented the method in Python using the multi-precision math package [7]. The table Convergence Results shows just how quickly the algorithm converges and finds an approximation for e. The algorithm computed e to 1004 digits in just 11 iterations. Here we compare the computed approximate value for e, i.e., x^x, to a value provided by MPMath, which was correct to 1500 digits. The results in the last two columns show that if the method produces an approximation x for $\overline{x}$ and $f(x) = 0$ to k digits, the approximation to e will be correct to k-1 digits.

Convergence Results			
step	corr	$ln(x) - \frac{1}{x}$	$e - x^x$
1	-5.0E-01	-2.6E-01	-8.8E-01
2	-2.4E-01	-2.5E-02	-1.2E-01
3	-2.8E-02	-2.7E-04	-1.3E-03
4	-3.1E-04	-3.2E-08	-1.6E-07
5	-3.7E-08	-4.6E-16	-2.2E-15
6	-5.1E-16	-9.1E-32	-4.4E-31
7	-1.0E-31	-3.6E-63	-1.7E-62
8	-4.0E-63	-5.6E-126	-2.7E-125
9	-6.3E-126	-1.4E-251	-6.5E-251
10	-1.5E-251	-8.1E-503	-3.8E-502
11	-9.1E-503	-2.9E-1005	-1.3E-1004

In addition, the table shows that with each iteration after the second iteration, the magnitude of the correction decreases quadratically as well. The same holds for the values of f and the error in the approximate values for e shown in the last column.

Note that the exponent of the error in the approximation to e, the last column, is roughly double the exponent of the Newton correction. The method presented computes a correction, updates x, the approximation to $\overline{x}$, and then updates the approximation to e. Since the Newton correction also converges quadratically to zero, the subsequent correction exponent will be double the previous correction exponent, as is seen in the correction column in the table. Thus, after a computed correction, the approximate value of x is actually correct to double the number of the current correction digits. Since as noted the error in x^x or the e approximation is about the same as the error in x, we stop the iteration when the correction exponent is half the desired exponent for the error. For example, iteration 11 shows the correction exponent is -503, while the error approximation exponent is -1004. To obtain 1000 correct digits for e, eps was set to 1.E-500, which terminated the iteration.

6. Concluding Remarks

The paper demonstrates a novel method for rapid, high precision computation of e that takes advantage of the fact that a transcendental function is available. Newton's method is employed to compute a transcendental number by finding the root of a function which is not algebraic . The paper provides a brief overview of methods for high precision elementary function evaluation, properties of exponential and the natural log functions, Newton's method and its convergence properties, and the use of extended precision arithmetic.

References

[1] M. Aghaeiboorkheili and J. Giuina, "The History of the Derivation of Euler's Number," *Journal of Applied Mathematics and Physics*, 10(9): DOI: 10.4236/jamp.2022.109185.

[2] J.M. Borwein and P.B. Borwein, "The Arithmetic-Geometric Mean and Fast Computation of Elementary Functions," *SIAM Review*, 26(3):351-356.DOI:https://doi.org/10.1137/1026073, 1984.

[3] R.P. Brent, "Fast Multiple-Precision Evaluation of Elementary Functions," *Journal of the ACM*, 23(2):242-251.DOI:https://doi.org/10.1137/1026073, 1976.

[4] R.P. Brent, "Fast Algorithms for High-Precision Computation of Elementary Functions(Extended Abstract)," https://maths-people.anu.edu.au/ brent/pd/RNC7-Brent.pdf, 2006.

[5] H. Brothers, "Improving the Convergence of Newton's Series Approximation for e," *The College Mathematics Journal*, 36(1):34-39, 2004.

[6] W. Cheny and D. Kincaid, "Improving the Convergence of Newton's Series Approximation for e," *Numerical Mathematics and Computing: 7th Edition*, Boston, MA: Cengage Learning, Inc., 2013.

[7] MPMATH Development Team, " MPMath: a Python library for arbitrary-precision floating-point arithmetic (version 1.3.0)." https://mpmath.org/, 2023.

[8] Numbers, Constants and Computation,"The Binary Splitting Method." http://numbers.computation.free.fr, 2001.