

A NOTE ON THE EXTENDED ERHO-OKRINYA ALGORITHM FOR GREATEST COMMON DIVISOR USING SGN FUNCTION

Anton Iliev^{1,2}, Nikolay Kyurkchiev^{1,2},
Asen Rahnev^{1,2}, Angel Golev^{1,2}, Pavel Kyurkchiev^{1,2}

¹ Faculty of Mathematics and Informatics,
University of Plovdiv Paisii Hilendarski,
24, Tzar Asen Str., 4000 Plovdiv, Bulgaria

² Centre of Excellence in Informatics and
Information and Communication Technologies,
1113 Sofia, Bulgaria

ABSTRACT: In this research, we elaborate new extended algorithm which generalize “Algorithm 3a” from [2, 3]. Also we wider and optimize the received results concern this iteration process in studies [5, 6]. This algorithm also modify regular extended Euclidean algorithm [39]. The presented calculations are suitable for numbers, which are not very long.

AMS Subject Classification: 11A05, 68W01

Key Words: extended Euclidean algorithm, greatest common divisor, extended Erho-Okrinya algorithm, extended Tembhurne-Sathe algorithm, sgn function, computational process design

Received: June 29, 2026

Revised: August 25, 2026

Published: September 21, 2026

doi: 10.12732/ijdea.v25i1.4

Math Team /MT/ – 1991, Ltd.

<https://www.acadpubl.eu/ap/>

1. INTRODUCTION

For every integer numbers a and b we will depict extended algorithm for finding such integer numbers x and y for which Bezout’s identity [1] is correct, i.e. $x * a + y * b = \text{Greatest Common Divisor (gcd) of } a \text{ and } b$. The intriguing question of searching greatest common divisor is widely spread research topic [7]–[43].

Other forms of usage of these and modified algorithms are well presented in [44]–[60].

For testing purposes we will use the following PC configuration:

- Processor – Intel(R) Core(TM) i7-8565U CPU 1.80GHz, 1992 Mhz, 4 Core(s), 8 Logical Processor(s);
- Installed Physical Memory (RAM) – 8.00 GB;
- System Manufacturer – Acer;
- System Model – Swift SF314-55G;
- System Type – x64-based PC;
- OS Name – Microsoft Windows 11 Enterprise x64;
- Software Environment – Microsoft Visual Studio Community 2026 x64.

In our previous study we obtain [5] extended algorithm via “Algorithm 3a” from [2]. In [6] through “Algorithm 3” from [2] we receive new extended algorithm using sgn function. Now we will use this extended algorithm [5] to obtain more general form of this iteration process. The purpose of this aim is because the computational procedures from [3], [5] are faster than these in [8], [4]. For testing purposes here we will use also the extended Tembhurne-Sathe algorithm through sgn function [33].

2. MAIN RESULTS

Using sgn function we describe the new Extended Algorithms 1. and 2. placed below:

Algorithm 1.

```

if (Math.Abs(a) > Math.Abs(b))
{
    x2 = 0; y1 = 0;
    if (a > 0) x1 = 1; else if (a < 0) { x1 = -1; a = -a; }
    if (b > 0) y2 = 1; else if (b < 0) { y2 = -1; b = -b; }
    D = a - b;
    while (D > 0)
    {

```

```
if (D < b)
{
    x3 = x1;
    r = x2;
    x1 = y1;
    x2 = y2;
    y1 = x3 - y1;
    y2 = r - y2;
    r = (b - 1) / D;
    x1 -= r * y1;
    x2 -= r * y2;
    a = (b - 1) % D + 1;
    r = (D - 1) / a;
    b = (D - 1) % a + 1;
}
else
{
    r = (a - 1) / b;
    x1 -= r * y1;
    x2 -= r * y2;
    a = (a - 1) % b + 1;
    r = (b - 1) / a;
    b = (b - 1) % a + 1;
}
D = a - b;
y1 -= r * x1;
y2 -= r * x2;
}
gcd = a;
x = x1;
y = x2;
}
else
{
    x1 = 0; y2 = 0;
    if (a > 0) y1 = 1; else if (a < 0) { y1 = -1; a = -a; }
```

```
if (b > 0) x2 = 1; else if (b < 0) { x2 = -1; b = -b; }
D = b - a;
while (D > 0)
{
    if (D < a)
    {
        x3 = x1;
        r = x2;
        x1 = y1;
        x2 = y2;
        y1 = x3 - y1;
        y2 = r - y2;
        r = (a - 1) / D;
        x1 -= r * y1;
        x2 -= r * y2;
        b = (a - 1) % D + 1;
        r = (D - 1) / b;
        a = (D - 1) % b + 1;
    }
    else
    {
        r = (b - 1) / a;
        x1 -= r * y1;
        x2 -= r * y2;
        b = (b - 1) % a + 1;
        r = (a - 1) / b;
        a = (a - 1) % b + 1;
    }
    D = b - a;
    y1 -= r * x1;
    y2 -= r * x2;
}
gcd = b;
x = x1;
y = x2;
}
```

as well as its recursive implementation as

Algorithm 2.

```
static long Extended_Erho_Okrinya_Alg_3a_rec_sgn(long a, long b,
ref long x, ref long y, long x1, long x2, long y1, long y2)
{
    long D, x3, r;
    D = a - b;
    if (D > 0)
    {
        if (D < b)
        {
            x3 = x1;
            r = x2;
            x1 = y1;
            x2 = y2;
            y1 = x3 - y1;
            y2 = r - y2;
            r = (b - 1) / D;
            x1 -= r * y1;
            x2 -= r * y2;
            a = (b - 1) % D + 1;
            r = (D - 1) / a;
            b = (D - 1) % a + 1;
        }
        else
        {
            r = (a - 1) / b;
            x1 -= r * y1;
            x2 -= r * y2;
            a = (a - 1) % b + 1;
            r = (b - 1) / a;
            b = (b - 1) % a + 1;
        }
        y1 -= r * x1;
        y2 -= r * x2;
    }
    return
```

```

    Extended_Erho_Okrinya_Algorithm_3a_rec_sgn(a, b, ref x,
    ref y, x1, x2, y1, y2);
}
else { x = x1; y = x2; return a; }
}

```

The recursive Algorithm 2. can be started by:

```

if (Math.Abs(a) > Math.Abs(b))
{
    x2 = 0; y1 = 0;
    if (a > 0) x1 = 1; else if (a < 0) { x1 = -1; a = -a; }
    if (b > 0) y2 = 1; else if (b < 0) { y2 = -1; b = -b; }
    gcd = Extended_Erho_Okrinya_Algorithm_3a_rec_sgn(a, b, ref x,
        ref y, x1, x2, y1, y2);
}
else
{
    x1 = 0; y2 = 0;
    if (a > 0) y1 = 1; else if (a < 0) { y1 = -1; a = -a; }
    if (b > 0) x2 = 1; else if (b < 0) { x2 = -1; b = -b; }
    gcd = Extended_Erho_Okrinya_Algorithm_3a_rec_sgn(b, a, ref x,
        ref y, x1, x2, y1, y2);
}

```

3. NUMERICAL EXAMPLE

We will make experiment with new Algorithms 1. and 2. as well as Algorithms 1. and 2. from [33] with the following example:

```

long a, b, d = 0, r, gcd = 0;
long x1 = 0, x2 = 0, x3, y1 = 0, y2 = 0;
long D, b2, a2, x = 0, y = 0, q, u, v, sng;
for (int i = 1; i < 1000000001; i++)
{
    a = i; b = 2000000002 - i;
    //Here are placed the source code of
    //new Algorithm 1. and Algorithm 1. from [33]
}

```

```
//as well as calling of new recursive Algorithm 2.
//as well as Algorithm 2. from [33].
d += gcd;
}
Console.WriteLine(d);
```

CPU time of new Algorithm 1 is: **26.570 seconds**.

CPU time of new Algorithm 2 is: **29.773 seconds**.

4. CONCLUSION

The numerical results confirm that the new Algorithm 1. is faster than Algorithm 1. (**33.992 seconds**) from [33] but also it is quicker than Algorithms 1. and 3. from [6] (which finish time was 1. – **33.286 seconds** and 3. – **28.779 seconds**). It also can be noted that new Algorithm 2. take shorter time than Algorithm 4. (**36.072 seconds**) from [33] and it is faster than Algorithms 2. and 4. from [6] (which finish time was 2. – **44.719 seconds** and 4. – **33.145 seconds**).

ACKNOWLEDGMENT

This study is supported by the Centre of Excellence in Informatics and ICT under the Grant No. BG16RFPR002-1.014-0018-C01, financed by the Research, Innovation and Digitalization for Smart Transformation Programme 2021–2027 and co-financed by the European Union.

REFERENCES

- [1] E. Bezout, *Theorie generale des equations algebriques*, Paris, France: Ph.-D. Pierres, 1779.
- [2] J. Erho, A. Okrinya, Novel Approach to Shortening Convergence of Euclidean's Greatest Common Divisor Algorithm, *Mathematics Open*, 2026, **05**, 2550022.
- [3] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, V. Matanski, One Note on the Erho-Okrinya Algorithm for Greatest Common Divisor, *International Electronic Journal of Pure and Applied Mathematics*, 2026, 20, No. 1, (in print).

- [4] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, V. Kyurkchiev, A New Extended Erho-Okrinya Algorithm for Calculating the Greatest Common Divisor, *International Journal of Differential Equations and Applications*, 2026, 25, No. 1, 17–32.
- [5] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, V. Matanski, A Note on the Extended Erho-Okrinya Algorithm for Greatest Common Divisor, *International Electronic Journal of Pure and Applied Mathematics*, 2026, 20, No. 1, (in print).
- [6] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, V. Kyurkchiev, Extended Erho-Okrinya Algorithm for Finding the Greatest Common Divisor using SGN Function, *International Electronic Journal of Pure and Applied Mathematics*, 2026, 20, No. 1, 11–26.
- [7] W. Su, X. Wu, Y. Zhao, NL2ACSL: Interactively Translating Natural Language to ANSI C Specification Language with Large Language Models, *Knowledge-Based Systems*, 2025.
- [8] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, P. Kyurkchiev, A More Efficient Description of the Erho-Okrinya Algorithm for Calculating the Greatest Common Divisor, *International Electronic Journal of Pure and Applied Mathematics*, 2026, 20, No. 1, 1–10.
- [9] K. Garov, A. Rahnev, *Textbook-notes on programming in BASIC for facultative training in mathematics for 9.–10. Grade of ESPU*, Sofia, 1986, (in Bulgarian).
- [10] A. Golev, *Textbook on algorithms and programs in C#*, University Press “Paisii Hilendarski”, Plovdiv, 2012, (in Bulgarian).
- [11] T. Terzieva, *Introduction to web programming*, University Press “Paisii Hilendarski”, Plovdiv, 2021, ISBN: 978-619-202-623-3, (in Bulgarian).
- [12] T. Terzieva, *Development of algorithmic thinking in the Informatics Education*, University Press “Paisii Hilendarski”, Plovdiv, 2021, ISBN: 978-619-202-622-6, (in Bulgarian).
- [13] T. Terzieva, *Educational tools for teaching in digital environment*, University Press “Paisii Hilendarski”, Plovdiv, 2021, (in Bulgarian).
- [14] S. Enkov, *Programming in Arduino Environment*, University Press “Paisii Hilendarski”, Plovdiv, 2017, (in Bulgarian).
- [15] A. Iliev, N. Kyurkchiev, A. Rahnev, New Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 2020, 28, No. 1, 69–74.
- [16] A. Iliev, N. Kyurkchiev, A Note on Knuth’s Implementation of Euclid’s Greatest

- Common Divisor Algorithm, *International Journal of Pure and Applied Mathematics*, 2017, **117**, 603–608.
- [17] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Harris-Stein Modification of Euclidean Algorithm for Greatest Common Divisor. II, *International Journal of Pure and Applied Mathematics*, 2018, 120, No. 3, 379–388.
- [18] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of TembhurneSathe Modification of Euclidean Algorithm for Greatest Common Divisor. IV, *Dynamic Systems and Applications*, 2019, 28, No. 1, 143–152.
- [19] A. Iliev, N. Kyurkchiev, A. Golev, A Note on Knuth’s Implementation of Extended Euclidean Greatest Common Divisor Algorithm, *International Journal of Pure and Applied Mathematics*, 2018, **118**, 31–37.
- [20] A. Iliev, N. Kyurkchiev, A Note on Euclidean and Extended Euclidean Algorithms for Greatest Common Divisor for Polynomials, *International Journal of Pure and Applied Mathematics*, 2018, **118**, 713–721.
- [21] A. Iliev, N. Kyurkchiev, A Note on Knuth’s Algorithm for Computing Extended Greatest Common Divisor using SGN Function, *International Journal of Scientific Engineering and Applied Science*, 2018, **4**, No. 3, 26–29.
- [22] A. Iliev, N. Kyurkchiev, *New Trends in Practical Algorithms: Some Computational and Approximation Aspects*, LAP LAMBERT Academic Publishing, Beau Bassin, 2018.
- [23] A. Iliev, N. Kyurkchiev, The faster extended Euclidean algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, 2019, 21–26.
- [24] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Least Absolute Remainder Algorithm for Greatest Common Divisor. III, *Neural, Parallel, and Scientific Computations*, 2019, **27**, No. 1, 1–9.
- [25] A. Iliev, N. Kyurkchiev, A. Rahnev, *Nontrivial Practical Algorithms: Part 2*, LAP LAMBERT Academic Publishing, Beau Bassin, 2019.
- [26] A. Iliev, N. Valchanov, T. Terzieva, Generalization and Optimization of Some Algorithms, *Collection of scientific works of National Conference “Education in Information Society”*, Plovdiv, ADIS, 12–13 May 2009, 2009, 52–58, (in Bulgarian).
- [27] A. Iliev, N. Kyurkchiev, A. Rahnev, New Extended Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 2020, 28, No. 1,

89–95.

- [28] A. Iliev, N. Kyurkchiev, A. Rahnev, V. Kyurkchiev, New Extended Based on Generalization of Harris Algorithm, *Communications in Applied Analysis*, 2022, **26**, No. 1, 61–73.
- [29] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, A Note on Extended Based On Generalization of Harris Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 2023, 17, No. 1, 53–64.
- [30] A. Iliev, N. Kyurkchiev, A. Rahnev, A Refinement of the Extended Euclidean Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 2021, 15, No. 1, 33–44.
- [31] A. Iliev, N. Kyurkchiev, A. Rahnev, S. Enkov, An Effective Realization of Daykin-Harris Algorithm, *International Journal of Differential Equations and Applications*, 2024, 23, No. 1, 29–40.
- [32] A. Iliev, N. Kyurkchiev, A. Rahnev, S. Enkov, Extended Based On Generalized Daykin-Harris Algorithm, *International Journal of Differential Equations and Applications*, 2024, 23, No. 1, 41–49.
- [33] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, Extended Based on Generalized Tembhurne-Sathe Algorithm Using Sgn Function, *International Journal of Differential Equations and Applications*, 2023, 22, No. 1, 47–56.
- [34] A. Iliev, N. Kyurkchiev, A. Rahnev, A Note on Adaptation of the Knuth's Extended Euclidean Algorithm for Computing Multiplicative Inverse, *International Journal of Pure and Applied Mathematics*, 2018, 118, No. 2, 281–290.
- [35] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, New Extended Based on Generalization of Tembhurne-Sathe Algorithm, *International Journal of Differential Equations and Applications*, 2022, 21, No. 2, 89–100.
- [36] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, Efficient Binary Extended Algorithm Using SGN Function, *International Journal of Differential Equations and Applications*, 2021, 20, No. 2, 179–186.
- [37] A. Iliev, N. Kyurkchiev, A Note on Least Absolute Remainder Euclidean Algorithm for Greatest Common Divisor, *International Journal of Scientific Engineering and Applied Science*, 2018, 4, No. 3, 31–34.
- [38] A. Iliev, N. Kyurkchiev, A. Rahnev, New Extended Algorithm Using Least Absolute Remainder, *International Journal of Differential Equations and Applications*, 2022, 21, No. 1, 85–92.

- [39] D. Knuth, *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*, 3rd ed., Addison-Wesley, Boston, 1998.
- [40] A. Rahnev, K. Garov, O. Gavrilov, *Textbook for extracurricular work using BASIC*, MNP Press, Sofia, 1985, (in Bulgarian).
- [41] A. Rahnev, K. Garov, O. Gavrilov, *BASIC in examples and tasks*, Government Press “Narodna prosveta”, Sofia, 1990, (in Bulgarian).
- [42] N. Kasakliev, *C# Programming Guide*, University Press “Paisii Hilendarski”, Plovdiv, 2016, (in Bulgarian).
- [43] A. Rahnev, N. Pavlov, N. Valchanov, T. Terzieva, *Object Oriented Programming*, Lightning Source UK Ltd., London, 2014.
- [44] J. Erho, J. Consul, B. Japheth, Juggling Versus Three-Way-Reversal Sequence Rotation Performance Across Four Data Types, *International Journal of Scientific Research in Computer Science and Engineering*, 2019, **7**, No. 6, 10–18.
- [45] C. Falcon Rodriguez, M. Cruz, C. Falcon, Full Euclidean Algorithm by Means of a Steady Walk, *Applied Mathematics*, 2021, **12**, 269–279.
- [46] V. Matanski, An Efficient Binary Algorithm for Solving Equation $GCD * 2^{|J-K|} = X * A0 + Y * B0$, *Proc. of Anniversary International Scientific Conference “Computer Technologies and Applications”*, 15–17 September 2021, Pamporovo, Bulgaria, *Plovdiv University Press*, 79–86, ISBN: 978-619-202-702-5.
- [47] H. Gyulyustan, A Note on Euclidean Sequencing Algorithm, *Proc. of the Scientific Conference “Innovative ICT for Digital Research Space in Mathematics, Informatics and Educational Pedagogy”*, Pamporovo, 7–8 November 2019, *Plovdiv University Press*, 57–63, ISBN: 978-619-202-572-4.
- [48] P. Kyurkchiev, V. Matanski, The Faster Euclidean Algorithm for Computing Polynomial Multiplicative Inverse, *Proc. of the Scientific Conference “Innovative ICT in Research and Education: Mathematics, Informatics and Information Technologies”*, Pamporovo, 29–30 November 2018, (2019), 43–48, ISBN: 978-619-202-439-0.
- [49] V. Matanski, P. Kyurkchiev, The Faster Lehmer’s Greatest Common Divisor Algorithm, *Proc. of the Scientific Conference Innovative “ICT in Research and Education: Mathematics, Informatics and Information Technologies”*, Pamporovo, 29–30 November 2018, (2019), 37–42, ISBN: 978-619-202-439-0.
- [50] Z. Ashraf, A. Sohail, M. Iqbal, Design and Performance Evaluation of an Authentic End-to-End Communication Model on Large-Scale Hybrid IPv4-IPv6 Virtual Networks to Detect MITM Attacks, *Cryptography*, 2024, **8**, No. 4, 49.

- [51] L. Babenko, V. Starodubcev, Estimation of The Execution Time of Encryption, Decryption, and Homomorphic Calculations Using The Domingo-Ferrer Cryptosystem, *Izvestiya SFedU. Engineering Sciences*, 2024, No 5, 6–15.
- [52] A. Ramamoorthy, A. Dutta, M. Lakshmi, Enhanced Encryption of Lunar Hyperspectral Data using Blinded Multi Prime RSA Cryptosystem, *J Indian Soc Remote Sens*, 2025, 16 pp.
- [53] H. Li, L. Deng, Y. Gu, N. Wang, Y. Chen, S. Li, A secure data sharing scheme based on searchable public key encryption for authorized multi-receiver, *Journal of Systems Architecture*, 2025, **167**, 103489.
- [54] K. Song, J. Wei, H. Hu, X. Hou, L. Yan, S. Zang, Design and Implementation of a Parallel Stein Algorithm Based on a Ternary Optical Computer, *Appl. Opt.*, 2025, **64**, 6888–6896.
- [55] H. Zhang, H. Zhang, C. Xia, Z. Huang, Y. Chen, A. Barnard, ROCKET: An RNS-Based Photonic Accelerator for High-Precision and Energy-Efficient DNN Training, *Proc. of the 2025 International Conference on Supercomputing (ICS'25)*, Salt Lake City, UT, USA, 8–11 June 2025, 14 pages.
- [56] L. Babenko, V. Starodubcev, N. Yelchaninova, Estimation of the Search Time for Key Components in a Known Plaintext Attack on the Domingo-Ferrer Cryptosystem, *Izvestiya SFedU. Engineering Sciences*, 2025, No. 3, 110–118.
- [57] Z. Ibran, E. Aljatlawi, A. Awin, On Continued Fractions and Their Applications, *Journal of Applied Mathematics and Physics*, 2022, **10**, 142–159.
- [58] N. Boubaker, Comprehensive Insights into Matrix Continued Fractions and Their Uses, *North African Journal of Scientific Publishing*, 2025, **3**, 53–61.
- [59] S. Ghosal, B. Sau, Reducing Interception Latency in High-Security Facilities via AO* Optimization Algorithm, *International Journal of Innovative Science and Research Technology*, 26 No. AUG221, 2026, 701–749.
- [60] P. Dumka, R. Chauhan, T. Verma, D. Mishra, Understanding Kuttaka (Pulverizer) method by Bhaskara II with the help of Microsoft excel, *Journal of Mathematical Problems, Equations and Statistics*, 2025, **6**, No. 1, 28–31.