



ONE NOTE ON THE ERHO-OKRINYA ALGORITHM FOR GREATEST COMMON DIVISOR

Anton Iliev¹, Nikolay Kyurkchiev¹, Asen Rahnev¹,
 Todorka Terzieva¹, Viktor Matanski¹

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

Abstract: In this article, we continue our investigations on so-called “Algorithm 3a” by [1]. This algorithm is a slightly optimized version of “Algorithm 3” [1], which we have already investigated [2] as a key alternative to the classical organization of the Euclidean algorithm [33]. The algorithm presented here is mainly applicable for numbers of ordinary length.

AMS Subject Classification: 11A05, 68W01

Key Words: Euclidean algorithm, greatest common divisor, Erho-Okrinya algorithm, computational process design

1. Introduction

For every integer numbers a and b , the algorithmic realization of calculating greatest common divisor has been discussed in many sources [3]–[37]. New niche of these algorithmic processes can be revealed in sources [38]–[54].

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.

Received: July 2, 2026

Revised: August 3, 2026

Published: September 22, 2026

© 2026 Math Team /MT/ – 1991, Ltd.

url: <https://www.e.ijpam.eu>

We have presented a faster-to-compute iterative version of Harris' algorithm [12], [25]:

Algorithm 1.

```

int g = 0;
if ((a & 1) == 0 && (b & 1) == 0)
    do { a >>= 1; b >>= 1; g++; }
    while ((a & 1) == 0 && (b & 1) == 0);
while ((a & 1) == 0) a >>= 1;
while ((b & 1) == 0) b >>= 1;
if (a > b) do
{
    a %= b;
    if (a < 1) { gcd = b << g; break; }
    if ((a & 1) == 0)
    {
        do a >>= 1; while ((a & 1) == 0);
        if (a == 1) { gcd = a << g; break; }
    }
    else
    {
        a = b - a;
        do a >>= 1; while ((a & 1) == 0);
        if (a == 1) { gcd = a << g; break; }
    }
    b %= a;
    if (b < 1) { gcd = a << g; break; }
    if ((b & 1) == 0)
    {
        do b >>= 1; while ((b & 1) == 0);
        if (b == 1) { gcd = b << g; break; }
    }
    else
    {
        b = a - b;
        do b >>= 1; while ((b & 1) == 0);
        if (b == 1) { gcd = b << g; break; }
    }
} while (true);
else do
{
    b %= a;

```

```

    if (b < 1) { gcd = a << g; break; }
    if ((b & 1) == 0)
    {
        do b >>= 1; while ((b & 1) == 0);
        if (b == 1) { gcd = b << g; break; }
    }
    else
    {
        b = a - b;
        do b >>= 1; while ((b & 1) == 0);
        if (b == 1) { gcd = b << g; break; }
    }
    a %= b;
    if (a < 1) { gcd = b << g; break; }
    if ((a & 1) == 0)
    {
        do a >>= 1; while ((a & 1) == 0);
        if (a == 1) { gcd = a << g; break; }
    }
    else
    {
        a = b - a;
        do a >>= 1; while ((a & 1) == 0);
        if (a == 1) { gcd = a << g; break; }
    }
} while (true);

```

as well as its recursive description as

Algorithm 2.

```

static long Harris(long a, long b, int g)
{
    if (a > b)
    {
        a %= b;
        if (a < 1) return b << g;
        if ((a & 1) == 0)
            return Harris(a >> 1, b, g);
        else
        {
            if (a == 1) return a << g;

```

```

        a = b - a;
        if ((a & 1) == 0)
            return Harris(a >> 1, b, g);
    }
    else
    {
        b %= a;
        if (b < 1) return a << g;
        if ((b & 1) == 0)
            return Harris(a, b >> 1, g);
        else
        {
            if (b == 1) return b << g;
            b = a - b;
            if ((b & 1) == 0)
                return Harris(a, b >> 1, g);
        }
    }
    return Harris(a, b, g);
}

```

2. Main Results

We obtain the faster calculation process of Erho-Okrinya “Algorithm 3a” [1]:

Algorithm 3.

```

if (a > b)
{
    D = a - b;
    while (D > 0)
    {
        if (D < b)
        {
            a = (b - 1) % D + 1;
            b = (D - 1) % a + 1;
        }
        else
        {

```

```

        a = (a - 1) % b + 1;
        b = (b - 1) % a + 1;
    }
    D = a - b;
}
gcd = a;
}
else
{
    D = b - a;
    while (D > 0)
    {
        if (D < a)
        {
            b = (a - 1) % D + 1;
            a = (D - 1) % b + 1;
        }
        else
        {
            b = (b - 1) % a + 1;
            a = (a - 1) % b + 1;
        }
        D = b - a;
    }
    gcd = b;
}

```

as well as its recursive presentation –

Algorithm 4.

```

static long Erho_Okrinya_Alg_3a_rec(long a, long b)
{
    long D;
    D = a - b;
    if (D > 0)
    {
        if (D < b)
        {
            a = (b - 1) % D + 1;
            b = (D - 1) % a + 1;
        }
    }
}

```

```

        else
        {
            a = (a - 1) % b + 1;
            b = (b - 1) % a + 1;
        }
        return Erho_Okrinya_Alg_3a_rec(a, b);
    }
    else return a;
}

```

The Algorithms 2. and 4. can be activated by:

```

int g = 0;
if ((a & 1) == 0 && (b & 1) == 0)
    do { a >>= 1; b >>= 1; g++; }
    while ((a & 1) == 0 && (b & 1) == 0);
while ((a & 1) == 0) a >>= 1;
while ((b & 1) == 0) b >>= 1;
gcd = Harris(a, b, g);

and

if (a > b) gcd = Erho_Okrinya_Alg_3a_rec(a, b);
else gcd = Erho_Okrinya_Alg_3a_rec(b, a);

```

respectively.

3. Numerical Example

We will using the introduced above new Algorithms 3. and 4. as well as Algorithms 1. and 2. with the numerical example below:

```

long a, b, gcd = 0, d = 0, D;
for (int i = 1; i < 100000001; i++)
{
    a = i; b = 200000002 - i;
    //here is placed the code of Algorithms 1. and 3.,
    //and calling of Algorithms 2. and 4.
    d += gcd;
}
Console.WriteLine(d);

```

CPU time of Algorithm 1 is: **21.443 seconds**.

CPU time of Algorithm 2 is: **29.144 seconds**.

CPU time of Algorithm 3 is: **18.473 seconds**.

CPU time of Algorithm 4 is: **21.279 seconds**.

4. Conclusion

We recall that the iterative and recursive Algorithms 3. and 4. (which optimize Algorithm 3 from [1]) from [2] give **19.912 seconds** and **22.229 seconds**, respectively. Algorithms 1. and 2. from [2] also are slower and their time is **22.501 seconds** and **23.988 seconds**, respectively. The results of the numerical calculations give us confidence that the new algorithms 3. and 4. have the necessary capabilities for their practical application, as they take less time.

Acknowledgments

This study is financed by the project No FP25-FMI-010 “Innovative Interdisciplinary Research in Informatics, Mathematics, and Pedagogy of Education” of the Scientific Fund of the Paisii Hilendarski University of Plovdiv, Bulgaria.

References

- [1] J. Erho, A. Okrinya, Novel Approach to Shortening Convergence of Euclidean’s Greatest Common Divisor Algorithm, *Mathematics Open*, 2026, **05**, 2550022.
- [2] 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.
- [3] W. Su, X. Wu, Y. Zhao, NL2ACSL: Interactively Translating Natural Language to ANSI C Specification Language with Large Language Models, *Knowledge-Based Systems*, 2025.
- [4] 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).

- [5] A. Golev, *Textbook on algorithms and programs in C#*, University Press “Paisii Hilendarski”, Plovdiv, 2012, (in Bulgarian).
- [6] T. Terzieva, *Introduction to web programming*, University Press “Paisii Hilendarski”, Plovdiv, 2021, ISBN: 978-619-202-623-3, (in Bulgarian).
- [7] T. Terzieva, *Development of algorithmic thinking in the Informatics Education*, University Press “Paisii Hilendarski”, Plovdiv, 2021, ISBN: 978-619-202-622-6, (in Bulgarian).
- [8] T. Terzieva, *Educational tools for teaching in digital environment*, University Press “Paisii Hilendarski”, Plovdiv, 2021, (in Bulgarian).
- [9] S. Enkov, *Programming in Arduino Environment*, University Press “Paisii Hilendarski”, Plovdiv (2017), (in Bulgarian).
- [10] A. Iliev, N. Kyurkchiev, A. Rahnev, New Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 2020, 28, No. 1, 69–74.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] A. Iliev, N. Kyurkchiev, *New Trends in Practical Algorithms: Some Computational and Approximation Aspects*, LAP LAMBERT Academic Publishing, Beau Bassin, 2018.

- [18] A. Iliev, N. Kyurkchiev, The faster extended Euclidean algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, 2019, 21–26.
- [19] 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.
- [20] A. Iliev, N. Kyurkchiev, A. Rahnev, *Nontrivial Practical Algorithms: Part 2*, LAP LAMBERT Academic Publishing, Beau Bassin, 2019.
- [21] 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).
- [22] 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.
- [23] 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.
- [24] A. Iliev, N. Kyurkchiev, A. Rahnev, A Refinement of the Extended Euclidean Algorithm, (2021), *International Electronic Journal of Pure and Applied Mathematics*, 2021, 15, No. 1, 33–44.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.

- [30] 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.
- [31] 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.
- [32] 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.
- [33] D. Knuth, *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*, 3rd ed., Addison-Wesley, Boston, 1998.
- [34] A. Rahnev, K. Garov, O. Gavrilov, *Textbook for extracurricular work using BASIC*, MNP Press, Sofia, 1985, (in Bulgarian).
- [35] A. Rahnev, K. Garov, O. Gavrilov, *BASIC in examples and tasks*, Government Press “Narodna prosveta”, Sofia, 1990, (in Bulgarian).
- [36] N. Kasakliev, *C# Programming Guide*, University Press “Paisii Hilendarski”, Plovdiv, 2016, (in Bulgarian).
- [37] A. Rahnev, N. Pavlov, N. Valchanov, T. Terzieva, *Object Oriented Programming*, Lightning Source UK Ltd., London, 2014.
- [38] 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.
- [39] C. Falcon Rodriguez, M. Cruz, C. Falcon, Full Euclidean Algorithm by Means of a Steady Walk, *Applied Mathematics*, 2021, 12, 269–279.
- [40] 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.
- [41] H. Gyulyustan, A Note on Euclidean Sequencing Algorithm, *Proceedings of the Scientific Conference “Innovative ICT for Digital Research Space in Mathematics, Informatics and Educational Pedagogy”*, Pamporovo, 7–8 November 2019, *Plovdiv University Press*, 2020, 57–63, ISBN: 978-619-202-572-4.

- [42] 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.
- [43] 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.
- [44] 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.
- [45] 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.
- [46] 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.
- [47] 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.
- [48] 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.
- [49] 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.
- [50] 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.
- [51] Z. Ibran, E. Aljatlawi, A. Awin, On Continued Fractions and Their Applications, *Journal of Applied Mathematics and Physics*, 2022, **10**, 142–159.
- [52] N. Boubaker, Comprehensive Insights into Matrix Continued Fractions and Their Uses, *North African Journal of Scientific Publishing*, 2025, **3**, 53–61.

- [53] 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.
- [54] 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.