Communications in Applied Analysis, **25**, No. 1 (2021), 39-51 ISSN: 1083-2564

A REFINEMENT OF THE EXTENDED EUCLIDEAN ALGORITHM USING SGN FUNCTION

Anton Iliev[1,2], Nikolay Kyurkchiev[1,2],
Asen Rahnev[1], Todorka Terzieva[1]

[1]Faculty of Mathematics and Informatics
University of Plovdiv Paisii Hilendarski
24, Tzar Asen Str., 4000 Plovdiv, BULGARIA
[2]Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., Bl. 8, 1113 Sofia, BULGARIA

ABSTRACT: We present new extended algorithms, which work for every integer numbers a and b. These algorithms show high computational effectiveness for regular and long numbers.

AMS Subject Classification: 11A05, 68W01
Key Words: extended algorithm, greatest common divisor, reduced number of operations

Received: March 13, 2021 ; **Accepted:** June 1, 2021 ;
Published: June 6, 2021. **doi:** 10.12732/caa.v25i1.4
Dynamic Publishers, Inc., Acad. Publishers, Ltd. http://www.acadsol.eu/caa

1. INTRODUCTION

For two integer numbers a and b such that $a^2 + b^2 > 0$ we search the integer numbers x and y for which $x*a+y*b = gcd$, where gcd is the greatest common divisor. In the case when $a = b = 0$ the greatest common divisor is not defined. We give nontraditional ways for performing such iteration processes [10]–[44].

Some practical applications of these algorithms can be explored in [52]–[57]. Different approach for this thematic can be found in [1]–[6] and [45]–[51].

For testing purposes we will use the following computer: processor – Intel(R) Core(TM) i7-6700HQ CPU 2.60GHz, 2592 Mhz, 4 Core(s), 8 Logical Processor(s), RAM 16 GB, Microsoft Windows 10 Enterprise x64, Microsoft Visual C# 2017 x64.

Using classical manner in [15] we provide the following:

Algorithm 1.

```
t = 0; u = 0;
if (a > 0) s = 1; else if (a < 0) s = -1;
if (b > 0) v = 1; else if (b < 0) v = -1;
b = Math.Abs(b); a = Math.Abs(a);
do
{
if (a < 1) { x = u; y = v; gcd = b; break; }
if (b < 1) { x = s; y = t; gcd = a; break; }
if (a > b)
{
q = a / b; a %= b;
s -= q * u; t -= q * v;
}
else
{
q = b / a; b %= a;
u -= q * s; v -= q * t;
}
}
while (true);
```

its recursive implementation as

Algorithm 2.

```
static long Euclid(long a, long b, ref long x,
ref long y, long s, long t, long u, long v)
{ if (a < 1) { x = u; y = v; return b; }
if (b < 1) { x = s; y = t; return a; }
if (a > b)
{ long q = a / b;
return Euclid(a % b, b, ref x, ref y,
s - q * u, t - q * v, u, v);
}
else
{ long q = b / a;
return Euclid(a, b % a, ref x, ref y,
s, t, u - q * s, v - q * t);
}
}
```

and its calling

```
t = 0; u = 0;
if (a > 0) s = 1; else if (a < 0) s = -1;
if (b > 0) v = 1; else if (b < 0) v = -1;
b = Math.Abs(b); a = Math.Abs(a);
if (a > b) gcd = Euclid(a, b, ref x, ref y, s, t, u, v);
else gcd = Euclid(b, a, ref y, ref x, s, t, u, v);
```

In [15] using the optimization of computational process we produce the following:

Algorithm 3.

```
t = 0; u = 0;
if (a > 0) s = 1; else if (a < 0) s = -1;
if (b > 0) v = 1; else if (b < 0) v = -1;
b = Math.Abs(b); a = Math.Abs(a);
if (a > b) do
{
```

```
q = a / b; a %= b;
s -= q * u; t -= q * v;
if (a < 1) { x = u; y = v; gcd = b; break; }
q = b / a; b %= a;
u -= q * s; v -= q * t;
if (b < 1) { x = s; y = t; gcd = a; break; }
}
while (true);
else do
{
q = b / a; b %= a;
u -= q * s; v -= q * t;
if (b < 1) { x = s; y = t; gcd = a; break; }
q = a / b; a %= b;
s -= q * u; t -= q * v;
if (a < 1) { x = u; y = v; gcd = b; break; }
}
while (true);
```

its recursive analog as

Algorithm 4.

```
static long Euclid(long a, long b, ref long x,
ref long y, long s, long t, long u, long v)
{
if (b < 1) { x = s; y = t; return a; }
long q = a / b; a %= b; s -= q * u; t -= q * v;
if (a < 1) { x = u; y = v; return b; }
q = b / a;
return Euclid(a, b % a, ref x, ref y,
s, t, u - q * s, v - q * t);
}
```

and its calling

```
t = 0; u = 0;
```

```
if (a > 0) s = 1; else if (a < 0) s = -1;
if (b > 0) v = 1; else if (b < 0) v = -1;
b = Math.Abs(b); a = Math.Abs(a);
if (a > b) gcd = Euclid(a, b, ref x, ref y, s, t, u, v);
else gcd = Euclid(b, a, ref y, ref x, s, t, u, v);
```

2. Main Results.

Using an approach generated by us [10]–[44] we present the following:

Algorithm 5.

```
t = 0; u = 0;
if (a > 0) s = 1; else if (a < 0) s = -1;
if (b > 0) v = 1; else if (b < 0) v = -1;
b = Math.Abs(b); a = Math.Abs(a);
do
if (a > b)
{
q = a / b; a %= b;
s -= q * u; t -= q * v;
if (a < 1) { x = u; y = v; gcd = b; break; }
b -= a; u -= s; v -= t;
if (a == b) { x = s; y = t; gcd = a; break; }
}
else
{
q = b / a; b %= a;
u -= q * s; v -= q * t;
if (b < 1) { x = s; y = t; gcd = a; break; }
a -= b; s -= u; t -= v;
if (a == b) { x = u; y = v; gcd = b; break; }
}
while (true);
```

its recursive analog as

Algorithm 6.

```
static long Euclid(long a, long b, ref long x,
ref long y, long s, long t, long u, long v)
{
long q = a / b; a %= b; s -= q * u; t -= q * v;
if (a < 1) { x = u; y = v; return b; }
b -= a; u -= s; v -= t;
if (a == b) { x = s; y = t; return a; }
if (b > a) return Euclid(b, a, ref x, ref y, u, v, s, t);
else return Euclid(a, b, ref x, ref y, s, t, u, v);
}
```

and its calling

```
t = 0; u = 0;
if (a > 0) s = 1; else if (a < 0) s = -1;
if (b > 0) v = 1; else if (b < 0) v = -1;
b = Math.Abs(b); a = Math.Abs(a);
if (a > b) gcd = Euclid(a, b, ref x, ref y, s, t, u, v);
else gcd = Euclid(b, a, ref y, ref x, s, t, u, v);
```

Numerical Example.

We will compare the new algorithms 5. and 6. with algorithms 1., 3. and 2., 4. implementations.

```
long a, b, x = 0, y = 0, u, s = 0, t, v = 0, q, gcd, d = 0;
for (int i = 1; i < 100000001; i++)
{ a = i; b = 200000002 - i;
//here can be placed the source code of
//every one of Algorithms 1, 3 and 5
//as well the calling of
//recursive Algorithms 2, 4 and 6
d += gcd; }
Console.WriteLine(d);
```

CPU time of Algorithm 1 is: **38.565 seconds.**

CPU time of Algorithm 2 is: **70.927 seconds.**

CPU time of Algorithm 3 is: **33.551 seconds.**

CPU time of Algorithm 4 is: **52.192 seconds.**

CPU time of Algorithm 5 is: **33.884 seconds.**

CPU time of Algorithm 6 is: **55.461 seconds.**

3. Conclusion

The Strassen [58] approach is in the basis of such iteration processes and the especially in hybrid Algorithms 5 and 6 which combine as main operations "remainder" and "difference".

ACKNOWLEDGEMENTS

This work has been accomplished with the financial support by the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014-2020) and co-financed by the European Union through the European structural and Investment funds.

REFERENCES

[1] A. Akritas, A new method for computing polynomial greatest common divisors and polynomial remainder sequences, *Numerische Mathematik*, **52** (1988), 119–127.

[2] S. Enkov, *Programming in Arduino Environment*, University Press "Paisii Hilendarski", Plovdiv (2017). (in Bulgarian)

[3] F. Chang, Factoring a Polynomial with Multiple-Roots, *World Academy of Science, Engineering and Technology*, **47** (2008), 492–495.

[4] Th. Cormen, Ch. Leiserson, R. Rivest, Cl. Stein, *Introduction to Algorithms*, 3rd ed., The MIT Press, Cambridge (2009).

[5] 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)

[6] A. Golev, *Textbook on algorithms and programs in C#*, University Press "Paisii Hilendarski", Plovdiv (2012). (in Bulgarian)

[7] T. Terzieva, *Introduction to web programming*, University Press "Paisii Hilendarski", Plovdiv (2021), ISBN 978-619-202-623-3. (in Bulgarian)

[8] T. Terzieva, *Development of algorithmic thinking in the Informatics Education*, University Press "Paisii Hilendarski", Plovdiv (2021), ISBN 978-619-202-622-6. (in Bulgarian)

[9] T. Terzieva, *Educational tools for teaching in digital environment*, University Press "Paisii Hilendarski", Plovdiv (2021). (in Bulgarian)

[10] A. Iliev, N. Kyurkchiev, A Note on Knuth's Implementation of Euclid's Greatest Common Divisor Algorithm, *International Journal of Pure and Applied Mathematics*, **117** (2017), 603–608.

[11] 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*, **118** (2018), 31–37.

[12] 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*, **118** (2018), 281–290.

[13] 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*, **118** (2018), 713–721.

[14] A. Iliev, N. Kyurkchiev, A Note on Least Absolute Remainder Euclidean Algorithm for Greatest Common Divisor, *International Journal of Scientific Engineering and Applied Science*, **4** No. 3 (2018), 31–34.

[15] 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*, **4** No. 3 (2018), 26–29.

[16] A. Iliev, N. Kyurkchiev, *New Trends in Practical Algorithms: Some Computational and Approximation Aspects*, LAP LAMBERT Academic Publishing, Beau Bassin (2018).

[17] A. Iliev, N. Kyurkchiev, 80th Anniversary of the birth of Prof. Donald Knuth, *Biomath Communications*, **5** (2018), 7 pp.

[18] A. Iliev, N. Kyurkchiev, New Realization of the Euclidean Algorithm, *Collection of scientific works of Eleventh National Conference with International Participation Education and Research in the Information Society*, Plovdiv, ADIS, June 1–2, (2018), 180–185. (in Bulgarian)

[19] A. Iliev, N. Kyurkchiev, New Organizing of the Euclid's Algorithm and one of its Applications to the Continued Fractions, *Collection of scientific works from conference "Mathematics. Informatics. Information Technologies. Application in Education"*, Pamporovo, Bulgaria, 10–12 October 2018, (2019), 199–207.

[20] A. Iliev, N. Kyurkchiev, The faster Euclidean algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 15–20.

[21] A. Iliev, N. Kyurkchiev, The faster extended Euclidean algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 21–26.

[22] P. Kyurkchiev, V. Matanski, The faster Euclidean algorithm for computing polynomial multiplicative inverse, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 43–48.

[23] V. Matanski, P. Kyurkchiev, The faster Lehmer's greatest common divisor algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 37–42.

[24] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement Euclidean Algorithm for Greatest Common Divisor. I, *Neural, Parallel, and Scientific Computations*, **26** No. 3 (2018), 355–362.

[25] 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*, **120** No. 3 (2018), 379–388.

[26] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Least Absolute Remainder Algorithm for Greatest Common Divisor. III, *Neural, Parallel, and Scientific Computations*, **27** No. 1 (2019), 1–9.

[27] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Tembhurne–Sathe Modification of Euclidean Algorithm for Greatest Common Divisor. IV, *Dynamic Systems and Applications*, **28** No. 1 (2019), 143–152.

[28] A. Iliev, N. Kyurkchiev, A. Rahnev, *Nontrivial Practical Algorithms: Part 2*, LAP LAMBERT Academic Publishing, Beau Bassin (2019).

[29] 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)

[30] H. Gyulyustan, A Note on Euclidean Sequencing Algorithm, *Proceedings of the Scientific Conference "Innovative ICT for Digital Research Area in Mathematics, Informatics and Pedagogy of Education"*, Pamporovo, 7–8 November 2019, Plovdiv University Press, (2020), 57–64.

[31] A. Iliev, N. Kyurkchiev, A. Rahnev, New Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 69–74.

[32] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Stein's Binary Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 75–80.

[33] A. Iliev, N. Kyurkchiev, A. Rahnev, New Algorithms for Finding Modular Multiplicative Inverse, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 81–88.

[34] A. Iliev, N. Kyurkchiev, A. Rahnev, New Extended Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 89–95.

[35] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Modular Multiplicative Inverse Binary Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 14 No. 1 (2020), 37–44.

[36] A. Iliev, N. Kyurkchiev, A. Rahnev, Recursive Extended Stein's Binary Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 14 No. 1 (2020), 31–36.

[37] A. Iliev, N. Kyurkchiev, A. Rahnev, A new improvement of Jacobi symbol algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 15 No. 1 (2020), 13–22.

[38] A. Iliev, N. Kyurkchiev, A. Rahnev, A new improvement of Jacobi symbol binary algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 15 No. 1 (2020), 1–11.

[39] A. Iliev, N. Kyurkchiev, A. Rahnev, Efficient Binary Algorithm for Kronecker Symbol, *Communications in Applied Analysis*, 25 No. 1 (2021), 11–21.

[40] A. Iliev, N. Kyurkchiev, A. Rahnev, Efficient Algorithm for Kronecker Symbol, *International Electronic Journal of Pure and Applied Mathematics*, 15 No. 1 (2021), 23–30.

[41] A. Iliev, N. Kyurkchiev, A. Rahnev, A Refinement of the Knuth's Extended Euclidean Algorithm for Computing Modular Multiplicative Inverse, (2021), *Communications in Applied Analysis*, 25 No. 1 (2021), 23–37.

[42] A. Iliev, N. Kyurkchiev, A. Rahnev, A Refinement of the Extended Euclidean Algorithm, (2021), *International Electronic Journal of Pure and Applied Mathematics*, 15 No. 1 (2021), 33–44.

[43] A. Iliev, N. Kyurkchiev, A. Rahnev, T. Terzieva, A Refinement of the Böh's Algorithm for Computing Modular Multiplicative Inverse, (2021), preprint.

[44] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Extended Stein's Binary Algorithm, *Proceedings of the Anniversary International Scientific Conference "Synergetics and Reflection in Mathematics Education"*, Pamporovo, 16–18 October 2020, Plovdiv University Press, (2020), 259–264.

[45] D. Knuth, *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*, 3rd ed., Addison-Wesley, Boston (1998).

[46] Hr. Krushkov, A. Iliev, *Practical programming guide in Pascal, Parts I and II*, Koala press, Plovdiv (2002). (in Bulgarian)

[47] P. Nakov, P. Dobrikov, *Programming=++Algorithms*, 5th ed., Sofia (2015). (in Bulgarian)

[48] A. Rahnev, K. Garov, O. Gavrailov, *Textbook for extracurricular work using BASIC*, MNP Press, Sofia (1985). (in Bulgarian)

[49] A. Rahnev, K. Garov, O. Gavrailov, *BASIC in examples and tasks*, Government Press "Narodna prosveta", Sofia (1990). (in Bulgarian)

[50] N. Kasakliev, *C# Programming Guide*, University Press "Paisii Hilendarski", Plovdiv (2016). (in Bulgarian)

[51] A. Rahnev, N. Pavlov, N. Valchanov, T. Terzieva, *Object Oriented Programming*, Lightning Source UK Ltd., London (2014).

[52] D. Rachmawati, M. Budiman, On Using The First Variant of Dependent RSA Encryption Scheme to Secure Text: A Tutorial, *J. Phys.: Conf. Ser.*, (2020), 1542 012024.

[53] J. A. Erho, J. I. Consul, B. R. Japheth, Juggling Versus Three-Way-Reversal Sequence Rotation Performance Across Four Data Types, *International Journal of Scientific Research in Computer Science and Engineering*, **7** No. 6 (2019), 10–18.

[54] J. L. Butar-butar, F. Sinuhaji, Faktorisasi Polinomial Square-Free dan bukan Square-Free atas Lapangan Hingga Zp, *Jurnal Teori dan Aplikasi Matematika*, **3** No. 2 (2019), 132–142.

[55] L. Akcay, B. Ors, Comparison of RISC-V and transport triggered architectures for a post-quantum cryptography application, *Turk J Elec Eng & Comp Sci*, **29**, (2021), 321–333.

[56] C. Falcon Rodriguez, M. Cruz, C. Falcon, Full Euclidean Algorithm by Means of a Steady Walk, *Applied Mathematics*, **12** (2021), 269–279.

[57] Y. Fan, G. Chen, M. Cui, Formalization of Finite Field GF(2^n) Based on COQ, *Computer Science*, **47** No. 12 (2020), 311–318.

[58] V. Strassen, Gaussian Elimination is not Optimal, *Numer. Math.* **13**, (1969), 354–356.