



EFFICIENT ALGORITHM FOR KRONECKER SYMBOL

Anton Iliev^{1,2}, Nikolay Kyurkchiev^{1,2}, Asen Rahnev¹

¹Faculty of Mathematics and Informatics

University of Plovdiv Paisii Hilendarski

24, Tzar Asen Str., 4000 Plovdiv, BULGARIA

²Institute of Mathematics and Informatics

Bulgarian Academy of Sciences

Acad. G. Bonchev Str., Bl. 8, 1113 Sofia, BULGARIA

Abstract: We develop new efficient realization of Kronecker symbol calculation. Numerical experiments show that our approach leads to more than 10 times faster algorithm in comparison even to commercial CAS. We will point out that presented here algorithm extends the results by Bach and Shallit [48] for computation the Jacobi symbol in terms of the simple continued fraction of a rational number a/b because now we give the exact solution in such terms for more general Kronecker symbol.

AMS Subject Classification: 11A05, 68W01

Key Words: Kronecker symbol, reduced number of operations

1. Introduction

For every integer numbers a and b , the algorithm for calculating Kronecker symbol is given in [50]. This task generalize the task of calculation of Jacobi symbol, which is described in [48], [49], [37] and [38]. In series of papers and books we demonstrate how the task for efficient calculation of processes similar to Euclidean algorithm [1]–[6] and [41]–[50] can be optimized [10]–[40] for iterative and more over recursive realizations.

Received: January 11, 2021

Revised: March 19, 2021

Published: March 25, 2021

© 2021 Academic Publications, Ltd.

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

For testing purposes for new algorithm 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.

The calculation of Kronecker symbol without "undefined" cases is given in [50].

2. Main Results

Here we will give optimized realizations of algorithms in both recursive and iterative calculations of Kronecker symbol without "undefined" cases.

Algorithm 1.

```

if (((a & 1) == 0 && (b & 1) == 0) ||
    (b == 0 && Math.Abs(a) != 1)) Kronecker = 0;
else if (b == 0) Kronecker = 1;
else
{
a0 = a;
j = 1;
if (a < 0) if (b < 0) { b = -b; a = -a; j = -j; }
else a = -a;
else if (b < 0) b = -b;
while ((b & 1) == 0)
{
b >>= 1; r = a & 7;
if ((r == 3) || (r == 5)) j = -j;
}
if (a0 < 0 && (b & 3) == 3) j = -j;
do
{
if (a > 0)
{
while ((a & 1) == 0)
{
a >>= 1; r = b & 7;
if ((r == 3) || (r == 5)) j = -j;
}
if (a == 1) { b = 1; break; }
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
b %= a;
}
}

```

```

else break;
if (b > 0)
{
while ((b & 1) == 0)
{
b >>= 1; r = a & 7;
if ((r == 3) || (r == 5)) j = -j;
}
if (b == 1) break;
if ((b & 3) == 3 && (a & 3) == 3) j = -j;
a %= b;
}
else { b = a; break; }
} while (true);
if (b == 1) Kronecker = j; else Kronecker = 0;
}

```

For organizing recursion of Kronecker symbol we will use the optimized recursive realization for Jacobi symbol algorithm [37]:

Algorithm 2.

```

static long Jacobi(long a, long b, ref int j)
{
if (a == 0) return b;
else
{
if ((a & 1) == 0)
{
long r = b & 7;
if ((r == 3) || (r == 5)) j = -j;
return Jacobi(a >> 1, b, ref j);
}
if (a == 1) return 1;
}
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
b %= a;
if (b == 0) return a;
if ((b & 1) == 0)
{
long r = a & 7;
if ((r == 3) || (r == 5)) j = -j;
return Jacobi(b >> 1, a, ref j);
}
}
}

```

```

}
if ((b & 3) == 3 && (a & 3) == 3) j = -j;
a %= b;
if (a == 0) return b;
return Jacobi(a, b, ref j);
}

```

The recursive Algorithm 2 for Kronecker symbol should be called by:

```

if (((a & 1) == 0 && (b & 1) == 0) ||
(b == 0 && Math.Abs(a) != 1)) Kronecker = 0;
else if (b == 0) Kronecker = 1;
else
{
a0 = a;
j = 1;
if (a < 0) if (b < 0) { b = -b; a = -a; j = -j; }
else a = -a;
else if (b < 0) b = -b;
while ((b & 1) == 0)
{
b >>= 1; r = a & 7;
if ((r == 3) || (r == 5)) j = -j;
}
if (a0 < 0 && (b & 3) == 3) j = -j;
b = Jacobi(a, b, ref j);
if (b == 1) Kronecker = j; else Kronecker = 0;
}

```

Numerical Example

We will test the proposed Algorithms 1 and 2 for the following example:

```

long a, a0, b, r, d = 0;
int j, Kronecker;
for (int i = -50000000; i < 50000001; i++)
{ a = i; b = 25000001 - i;
//Here are placed Algorithm 1 and
//calling of recursive Algorithm 2.
d += Kronecker; }
Console.WriteLine(d);

```

CPU time of Algorithm 1 is: **36.203 seconds.**

CPU time of Algorithm 2 is: **62.277 seconds.**

Conclusion

We present efficient realizations of algorithms (iterative and recursive) for finding Kronecker symbol. The presented by us algorithms demonstrate high computational speed.

Acknowledgments

The work of the first author has been accomplished with the financial support of the MES by the Grant No. D01-387/18.12.2020 for NCDSC part of the Bulgarian National Roadmap on RIs.

The work of the second and third authors is supported by the Project FP21-FMI-002 "Intelligent Innovative ICT in Research in Mathematics, Informatics and Pedagogy of Education" of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

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, (2021), preprint.
- [40] 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.
- [41] D. Knuth, *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*, 3rd ed., Addison-Wesley, Boston (1998).
- [42] Hr. Krushkov, A. Iliev, *Practical programming guide in Pascal, Parts I and II*, Koala press, Plovdiv (2002). (in Bulgarian)

- [43] P. Nakov, P. Dobrikov, *Programming=++Algorithms*, 5th ed., Sofia (2015). (in Bulgarian)
- [44] A. Rahnev, K. Garov, O. Gavrailov, *Textbook for extracurricular work using BASIC*, MNP Press, Sofia (1985). (in Bulgarian)
- [45] A. Rahnev, K. Garov, O. Gavrailov, *BASIC in examples and tasks*, Government Press "Narodna prosveta", Sofia (1990). (in Bulgarian)
- [46] N. Kasakliev, *C# Programming Guide*, University Press "Paisii Hilendarski", Plovdiv (2016). (in Bulgarian)
- [47] A. Rahnev, N. Pavlov, N. Valchanov, T. Terzieva, *Object Oriented Programming*, Lightning Source UK Ltd., London (2014).
- [48] E. Bach, J. Shallit, *Algorithmic Number Theory, Vol. 1: Efficient Algorithms*, MA: MIT Press, Cambridge (1996), 343–344.
- [49] S. Eikenberry, J. Sorenson, Efficient Algorithms for Computing the Jacobi Symbol, *J. Symbolic Computation*, 26 (1998), 509–523.
- [50] E. Weisstein, Kronecker Symbol, From MathWorld—A Wolfram Web Resource, <https://mathworld.wolfram.com/KroneckerSymbol.html>

