ANNALS OF COMMUNICATIONS IN MATHEMATICS
Volume 8, Number 3 (2025), 425-430
DOI: 10.62072/acm.2025.080309
ISSN: 2582-0818
https://www.technoskypub.com/journal/acm/

# SUBEXPONENTIAL COMPUTATION OF TRUNCATED THETA SERIES

FRANCESCO SICA

ABSTRACT. We describe an algorithm to compute in $O(e^{c\sqrt{k\log k}})$ binary operations, for some absolute constant $c > 0$, expressions like

$$\sum_{1\leqslant n\leqslant 2^{\alpha}} e^{\frac{2\pi i n^2}{2^k}} n^a$$

and

$$\sum_{\substack{1\leqslant n\leqslant 2^{\alpha}\\ 1\leqslant m\leqslant 2^{\beta}}} e^{\frac{2\pi i n m}{2^k}} n^a m^b$$

where $\alpha, \beta = O(k)$ and $a, b$ are fixed (small) nonnegative integers. The error terms in these computations are $O(e^{-ck})$.

**Keywords.** Theta series, partial sums, integer factorisation.

## 1. INTRODUCTION

The problem of factoring large integers is central in cryptography and computational number theory. The current state of the art in factoring large integers $N$ is the Number Field Sieve algorithm [2, 3], a continuation of the efforts started with the Quadratic Sieve [8] and Continued Fraction [6] algorithms. We should also mention the Elliptic Curve Method (ECM) by H. Lenstra [4], which is particularly useful when $N$ has a small prime factor $p$. They are all probabilistic factoring algorithms. These algorithms have *heuristic* running times $O\big(\exp(c(\log N)^{1/3}(\log\log N)^{2/3})\big)$, $O\big(\exp(c(\log N)^{1/2}(\log\log N)^{1/2})\big)$ and $O\big(\exp(c(\log p)^{1/2}(\log\log p)^{1/2})\big)$ respectively, for some constant $c$ (not always the same). The first two strive to find nontrivial arithmetical relations of the form $x^2 \equiv y^2 \pmod N$ (which lead to a nontrivial factor by computing $\gcd(N, x + y)$), whereas the third is a generalisation of Pollard's $p-1$ method [7], involving computations in some elliptic curve group instead of $\mathbb{Z}/N$. We should note, however, that there exist probabilistic algorithms with proved running time $O\big(\exp((1 + o(1))(\log N)^{1/2}(\log\log N)^{1/2})\big)$ [5]. As far as the author is aware, no such rigorous bound exists in the form $O\big(\exp\left((\log N)^c\right)\big)$ for $c < 1/2$.

---

2020 *Mathematics Subject Classification.* 11M06, 94A60.

*Key words and phrases.* Theta series; Partial sums; Integer factorisation.

Received: September 18, 2025. Accepted: September 28, 2025. Published: September 30, 2025.

In contrast, deterministic factoring algorithms are still exponential, with the best result requiring $O\big(N^{1/5}(\log N)^{16/5}/(\log\log N)^{3/5}\big)$ bit operations [1]. The deterministic approach introduced in [9] is different. It essentially tries to compute the sum of divisors function $\sigma(n) = \sum_{d|n} d$ through successive averages (i.e., Cesaro means) via computations of series arising from the Riemann zeta function. In our researches, a crucial role is played by the evaluation of double series like

$$\sum_{\substack{1\leqslant n_1\leqslant x\\ 1\leqslant n_2\leqslant y}} e^{2\pi i p(n_1,n_2)} n_1^{\lambda} n_2^{\mu} \tag{1.1}$$

to a fixed precision[1] $O(\max(x,y)^{-c_1})$ in subexponential time, that is, performing $O(\max(x,y)^{\epsilon})$ binary computations for an arbitrary $\epsilon > 0$. Here, $p \in \mathbb{R}[X,Y]$ and $\lambda, \mu \geqslant 0$ are fixed (small) integers. If $p(X,Y) = a_1X + a_2Y + a_3$, then (1.1) splits into the product

$$e^{2\pi i a_3} \sum_{1\leqslant n_1\leqslant x} e^{2\pi i a_1 n_1} n_1^{\lambda} \sum_{1\leqslant n_2\leqslant y} e^{2\pi i a_2 n_2} n_2^{\mu} \ .$$

Using

$$\sum_{n\leqslant x} e^{2\pi i u n} n^k = \frac{1}{(2\pi i)^k}\frac{d^k}{du^k}\sum_{n\leqslant x} e^{2\pi i u n} = \frac{1}{(2\pi i)^k}\frac{d^k}{du^k}\left(\frac{e^{2\pi i u[x]}-1}{e^{2\pi i u}-1}\right) \ ,$$

one can then achieve the computation to the required precision (say $O(\max(x,y)^{-c_1})$) in subexponential time. The present work can be viewed as a first nontrivial follow-up.

1.1. **Remarks on notation.** We are concerned with (1.1), with $p(X,Y) = X^2/2^k, Y^2/2^k$ or $XY/2^k$. In the following, $\mathbf{e}_k(x)$ is a shorthand for $e^{\frac{2\pi i x}{2^k}}$ and $\mathbf{e}(x) = \mathbf{e}_0(x)$, so that (1.1) becomes

$$\sum_{\substack{1\leqslant n_1\leqslant x\\ 1\leqslant n_2\leqslant y}} \mathbf{e}\big(p(n_1,n_2)\big) n_1^{\lambda} n_2^{\mu} \ .$$

We will also write a finite sum in $j$

$$\sum_{\substack{1\leqslant n\leqslant x\\ 1\leqslant m\leqslant y}} \mathbf{e}\big(p(n,m)\big) n^{\lambda} m^{\mu}$$
$$\widehat{=} \sum_{j} \sum_{\substack{1\leqslant n_1\leqslant x_j\\ 1\leqslant n_2\leqslant y_j}} \mathbf{e}\big(p(n_1,n_2)\big) n_1^{\alpha_j} n_2^{\beta_j} \sum_{\substack{1\leqslant m_1\leqslant x_j'\\ 1\leqslant m_2\leqslant y_j'}} \mathbf{e}\big(p(m_1,m_2)\big) m_1^{\gamma_j} m_2^{\delta_j}$$

to denote

$$\sum_{\substack{1\leqslant n\leqslant x\\ 1\leqslant m\leqslant y}} \mathbf{e}\big(p(n,m)\big) n^{\lambda} m^{\mu}$$
$$= \sum_{j} C_j \sum_{\substack{1\leqslant n_1\leqslant x_j\\ 1\leqslant n_2\leqslant y_j}} \mathbf{e}\big(p(n_1,n_2)\big) n_1^{\alpha_j} n_2^{\beta_j} \sum_{\substack{1\leqslant m_1\leqslant x_j'\\ 1\leqslant m_2\leqslant y_j'}} \mathbf{e}\big(p(m_1,m_2)\big) m_1^{\gamma_j} m_2^{\delta_j}$$

and

$$\sum_{j} C_j \sum_{\substack{1\leqslant n_1\leqslant x_j\\ 1\leqslant n_2\leqslant y_j}} n_1^{\alpha_j} n_2^{\beta_j} \sum_{\substack{1\leqslant m_1\leqslant x_j'\\ 1\leqslant m_2\leqslant y_j'}} m_1^{\gamma_j} m_2^{\delta_j} \leqslant c_2 \sum_{\substack{1\leqslant n\leqslant x\\ 1\leqslant m\leqslant y}} n^{\lambda} m^{\mu} \ ,$$

[1]Positive absolute constants – independent of $x, y, u, v, k$ – will be denoted $c_1, c_2, \ldots$.

with the involved constant being absolute.

## 2. Genesis of the problem

In [9], it is mentioned that the bottleneck in the analytic factorization approach is the computation in $O(x^{-c_3})$ of such series as

$$\sum_{n_1,n_2\geqslant 1}\frac{\mathbf{e}(2\sqrt{xn_1n_2})}{n_1^{3/2}n_2^2}\ , \tag{2.1}$$

where $x$ is close to the integer to be factored. The first step in calculating (2.1) within the required precision is to consider the sum of terms

$$(n_1,n_2)\in[2^{k_1},2^{k_1+1})\times[2^{k_2},2^{k_2+1}) \tag{2.2}$$

over boxes of size $2^{k_1}\times 2^{k_2}$, for $\max(2^{k_1},2^{k_2})\leqslant x^{c_4}$. There are $O(\log^2 x)$ such boxes. Each box is then subdivided into $x^{2\epsilon}$ boxes by equal subdivision of its sides into $x^{\epsilon}$ intervals. We are thus reduced to considering sums over a subexponential number of boxes of type

$$(n_1,n_2)\in[y_1,y_1+2^u)\times[y_2,y_2+2^v)\ , \tag{2.3}$$

where $2^{k_1}\leqslant y_1<2^{k_1+1}, 2^{k_2}\leqslant y_2<2^{k_2+1}$ are integers and $\max(2^u/y_1,2^v/y_2)<x^{-\epsilon}$. Summing over these boxes allows to develop $n_1^{-3/2}$ and $n_2^{-2}$, as well as $\sqrt{n_1n_2}$ in the exponential, into Taylor series truncated after a finite number of terms. This will work as long as the box vertices in (2.2) have coordinates at least $x^{\epsilon}$. If this is not the case, it suffices to sum over the coordinates $<x^{\epsilon}$ trivially and reason in the way described if the other coordinate is $>x^{\epsilon}$.

For example, we can write, for $0\leqslant r<2^u$ and $0\leqslant s<2^v$,

$$\begin{aligned}2\sqrt{x(y_1+r)(y_2+s)}&=2\sqrt{xy_1y_2}+\sqrt{\frac{xy_2}{y_1}}\,r+\sqrt{\frac{xy_1}{y_2}}\,s\\&\quad-\frac{1}{2}\sqrt{\frac{xy_2}{y_1^3}}\,r^2-\frac{1}{2}\sqrt{\frac{xy_1}{y_2^3}}\,s^2+\frac{1}{2}\sqrt{\frac{x}{y_1y_2}}rs+\cdots\end{aligned} \tag{2.4}$$

with the error becoming smaller and smaller in absolute value, a finite number of terms sufficing to reduce it below $O(x^{-c_5})$ if $y_1,y_2$ are larger than $x^{\epsilon}$ (but in any case less than $x^{c_1}$). Similar expansions are derived for $(y_1+r)^{-3/2}(y_2+s)^{-2}$. Since $e^{\epsilon}-1=O(\epsilon)$, we can approximate (2.1) in a box (2.3) by considering only truncated Taylor expansions in $r,s$, to arrive to sums of type

$$\sum_{0\leqslant r<2^u}\sum_{0\leqslant s<2^v}\mathbf{e}\big(p(r,s)\big)r^as^b \tag{2.5}$$

for $a,b=O(1)$ and where $p\in\mathbb{R}[X,Y]$ is a polynomial with coefficients in $[0,1]$, using periodicity. Another approximation of a diophantine nature is then performed by approximating $p(X,Y)$ with $\mathfrak{p}(X,Y)$ coefficient-wise to the nearest rational coefficient with denominator $2^k$ for $k$ large enough that $|p(r,s)-\mathfrak{p}(r,s)|<x^{-\epsilon}$ for all $(r,s)\in[0,2^u)\times[0,2^v)$. We then obtain a final approximation

$$\sum_{0\leqslant r<2^u}\sum_{0\leqslant s<2^v}\mathbf{e}_k\big(f(r,s)\big)r^as^b\ , \tag{2.6}$$

where $f\in\mathbb{Z}[X,Y]$ and again $a,b=O(1)$, albeit with a larger constant involved. It is these expressions that we will show how to compute recursively. In the following, we will suppose that $a,b=O(\log x)$ and will focus on the simplest nontrivial case when $f(X,Y)=X^2,Y^2$ or $XY$.

## 3. THE CASE OF SECOND-DEGREE WITH SMALL COEFFICIENTS

We consider the following sums (for $\max(2^u, 2^v) = O(x^{c_6})$ and natural numbers $a, b = O(\log x)$):

$$\sum_{0\leqslant r<2^u} \sum_{0\leqslant s<2^v} \mathbf{e}_k(rs) r^a s^b = \sum_{0\leqslant r<2^u} \sum_{0\leqslant s<2^v} e^{\frac{2\pi i rs}{2^k}} r^a s^b \tag{3.1}$$

Note that in (3.1), one can suppose that $\max(u, v) \leqslant k$, since, after integer division by $2^k$, we have $r = \rho 2^k + r'$ with $0 \leqslant \rho < 2^{u-k}$ and $s = \sigma 2^k + s'$ with $0 \leqslant \sigma < 2^{v-k}$, and the previous equation becomes

$$\sum_{0\leqslant r'<\min(2^u,2^k)} \sum_{0\leqslant s'<\min(2^v,2^k)} \mathbf{e}_k(r's') \sum_{0\leqslant\rho<2^{u-k}} \sum_{0\leqslant\sigma<2^{v-k}} (\rho 2^k + r')^a (\sigma 2^k + s')^b$$

and the inner double sums on $\rho$ and $\sigma$ can be calculated explicitly after expanding the products, thereby reducing the computation of (3.1) to the evaluation of similar sums for smaller values of $a, b$ and $u, v \leqslant k$. Note also that, in the trivial case when $u + v \leqslant k$, a Maclaurin expansion of $\mathbf{e}_k(\cdot)$ with $O(\log x)$ terms will reduce the sum to a computation of Bernoulli polynomials of degree bounded by $O(\log x)$. In particular, in the nontrivial case we have $2^k = O(x^{c_7})$.

Let now $k_1 = \lceil k/2 \rceil$ and perform integer divisions by $2^{k_1}$ to write in (3.1)

$$\begin{cases} r = r_0 2^{k_1} + r_1 & (0 \leqslant r_1 < \min(2^u, 2^{k_1}), (0 \leqslant r_0 < 2^{u-k_1}) \ , \\ s = s_0 2^{k_1} + s_1 & (0 \leqslant s_1 < \min(2^v, 2^{k_1}), (0 \leqslant s_0 < 2^{v-k_1}) \ . \end{cases} \tag{3.2}$$

Then, after noticing that $e_k(2^{2k_1} r_0 s_0) = 1$, we obtain

$$\sum_{0\leqslant r<2^u} \sum_{0\leqslant s<2^v} \mathbf{e}_k(rs) r^a s^b = \sum_{\substack{0\leqslant r_0<2^{u-k_1} \\ 0\leqslant r_1<\min(2^u,2^{k_1})}} \sum_{\substack{0\leqslant s_0<2^{v-k_1} \\ 0\leqslant s_1<\min(2^v,2^{k_1})}} \mathbf{e}_{k-k_1}(r_0 s_1) \mathbf{e}_{k-k_1}(r_1 s_0) \mathbf{e}_k(r_1 s_1) (r_0 2^{k_1} + r_1)^a (s_0 2^{k_1} + s_1)^b \ .$$

As mentioned previously, since $r_1 s_1 = O(2^k)$, we can develop $\mathbf{e}_k(r_1 s_1)$ into a Maclaurin series using

$$\mathbf{e}_k(r_1 s_1) = \sum_{\kappa_1 \leqslant \log x} \frac{1}{\kappa_1!} \left( \frac{2\pi i r_1 s_1}{2^k} \right)^{\kappa_1} + O(x^{-c_8 \log\log x})$$

to get

$$\sum_{0\leqslant r<2^u} \sum_{0\leqslant s<2^v} \mathbf{e}_k(rs) r^a s^b$$
$$\triangleq \sum_{1\leqslant j\leqslant O(\log^3 x)} \sum_{\substack{0\leqslant r_0<2^{u-k_1} \\ 0\leqslant s_1<\min(2^v,2^{k_1})}} \mathbf{e}_{k-k_1}(r_0 s_1) r_0^{\alpha_j} s_1^{\beta_j} \sum_{\substack{0\leqslant s_0<2^{v-k_1} \\ 0\leqslant r_1<\min(2^u,2^{k_1})}} \mathbf{e}_{k-k_1}(r_1 s_0) r_1^{\gamma_j} s_0^{\delta_j}$$

with $\max_j(\alpha_j, \beta_j, \gamma_j, \delta_j) = O(\log x)$ (the constant in this upper bound is a priori larger than the constant involved in the upper bound of $a, b = O(\log x)$, although with a minimum of work it can be made practically the same). It should be noted at this point that each term in $j$ factors into the product of two *independent* sums, which allows to compute the product by computing the factors individually and multiplying the results together. The procedure can be iterated for each of the factors: since $\max(2^{u-k_1}, 2^{v-k_1}) \leqslant 2^{k-k_1} \leqslant 2^{k_1}$, defining $k_2 = \lceil k_1/2 \rceil$ and integer-dividing each variable by $2^{k_2}$, we are reduced to the same computation as above, with $k$ replaced by $k_1$, $k_1$ replaced by $k_2$, $u$ by $u - k_1$ and $v$

by $\min(v,k_1)$ for the first factor – resp. $u$ by $v-k_1$ and $v$ by $\min(u,k_1)$ for the second factor. Note also that $k-k_1=k_1$ or $k_1-1$, so that, for $X\in\mathbb{R}$,

$$\mathbf{e}_{k-k_1}(X)=\mathbf{e}_{k_1}(\varepsilon_1 X)\ ,$$

where $\varepsilon_1=1,2$. In particular, for each of

$$\sum_{\substack{0\leqslant r_0<2^{u-k_1}=X_{1,1}\\0\leqslant s_1<\min(2^v,2^{k_1})=Y_{1,1}}}\mathbf{e}_{k_1}(\varepsilon_1 r_0 s_1) r_0^{\alpha_j} s_1^{\beta_j}$$

and

$$\sum_{\substack{0\leqslant s_0<2^{v-k_1}=X_{1,2}\\0\leqslant r_1<\min(2^u,2^{k_1})=Y_{1,2}}}\mathbf{e}_{k_1}(\varepsilon_1 r_1 s_0) r_1^{\gamma_j} s_0^{\delta_j}\ ,$$

when splitting the variables as in (3.2) by integer division by $2^{k_2}$, we introduce $O(\log^3 x)$ new terms, from the Maclaurin expansion of the exponential of the product of the remainders which needs, by definition of $\triangleq$, to be expanded to the same $\log x$ terms.

When the two expansions are multiplied together, we will have a total of $O(\log^6 x)$ terms, which needs to be multiplied by the number of $j$-terms to find a grand total of $O(\log^{3+6} x)$ terms. Each of these terms will be a product of $4=2^2$ sums of type

$$\sum_{\substack{1\leqslant n_1<X_{2,m}\\1\leqslant n_2<Y_{2,m}}}\mathbf{e}_{k_2}(\varepsilon_1\varepsilon_2 n_1 n_2) n_1^{\lambda} n_2^{\mu}\ ,$$

where $\varepsilon_i\in\{1,2\}$ for $i=1,2$, and $X_{2,m},Y_{2,m}\leqslant 2^{k_2}$ for $m=1,2,3,4$.

In general, define $k_\omega=\lceil k_{\omega-1}/2\rceil$, for $\omega\geqslant 3$. Then

$$\sum_{0\leqslant r<2^u}\sum_{0\leqslant s<2^v}\mathbf{e}_k(rs)r^a s^b$$
$$\triangleq\sum_{1\leqslant j\leqslant O(\log^{3\cdot 2^\omega-3}x)}\prod_{1\leqslant m\leqslant 2^\omega}\sum_{\substack{0\leqslant r_{(m)}<X_{\omega,m}\\0\leqslant s_{(m)}<Y_{\omega,m}}}\mathbf{e}_{k_\omega}(\varepsilon_1\cdots\varepsilon_\omega r_{(m)}s_{(m)}) r_{(m)}^{\alpha_{j,m}} s_{(m)}^{\beta_{j,m}}\ ,\qquad (3.3)$$

where $X_{\omega,m},Y_{\omega,m}\leqslant 2^{k_\omega}$ and $\varepsilon_i\in\{1,2\}$. By induction, $k_\omega=k/2^\omega+\epsilon(\omega)$, where $0\leqslant\epsilon(\omega)<2$. At the $\omega$-th step, the computation of

$$\sum_{\substack{0\leqslant r_{(m)}<X_{\omega,m}\\0\leqslant s_{(m)}<Y_{\omega,m}}}\mathbf{e}_{k_\omega}(\varepsilon_1\cdots\varepsilon_\omega r_{(m)}s_{(m)}) r_{(m)}^{\alpha_{j,m}} s_{(m)}^{\beta_{j,m}}$$

trivially takes $O(2^{2k_\omega})$ operations; after finitely many $\omega$ steps this becomes $O(x^\epsilon)$. At that point, the overall computation time of (3.3) will be bounded by $O(x^\epsilon\log^{3\cdot 2^\omega-3}x)$.

In fact, in the previous analysis, the optimal choice of $\omega$ is such that $2^\omega=c_9\sqrt{k/\log k}$. In that case, the number of $j$-summands increases to and the factors' computation in (3.3) can be reduced to the same order of magnitude $O\big(e^{c_{10}\sqrt{k\log k}}\big)=O\big(e^{c_{11}\sqrt{\log x\log\log x}}\big)$.

Finally, notice that a truncated theta expression like

$$\sum_{1\leqslant n\leqslant 2^\alpha} e^{\frac{2\pi i n^2}{2^\kappa}} n^a$$

can be transformed into the case just considered by writing the integer division $n=r2^k+s$ as before, where $k=\lceil\kappa/2\rceil$, and using a Maclaurin expansion for $\mathbf{e}_\kappa(s^2)$.

## 4. Conclusion

Unfortunately, the previous method is very specialized and cannot readily be generalized to polynomials in more than two variables or degree higher than two. For instance, considering, for even $k$,

$$\sum_{0 \leqslant r < 2^k} \mathbf{e}_k(r^3)$$

and writing $r = r_0 2^{k_1} + r_1 = r_0 2^{k/2} + r_1$ as above, the previous expression becomes

$$\sum_{0 \leqslant r_0, r_1 < 2^{k_1}} \mathbf{e}_{k_1}(3 r_0 r_1^2) \mathbf{e}_k(r_1^3)$$

and doesn't split into a nontrivial product. There may be other splittings that could lead to a subexponential algorithm, but none seem as natural as what we described in this work. Being able to generalize this approach to a polynomial exponential sum in two variables of arbitrary degree would lead to a deterministic subexponential factoring algorithm.

## References

[1] D. Harvey and M. Hittmeir. A log-log speedup for exponent one-fifth deterministic integer factorisation. *Math. Comp.*, 91:1367–1379, 2022.
[2] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, and J. M. Pollard. The Number Field Sieve. In *ACM Symposium on Theory of Computing*, pages 564–572, 1990.
[3] A. K. Lenstra and H. W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1993.
[4] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126(3):649–673, 1987.
[5] H. W. Lenstra, Jr. and Carl Pomerance. A rigorous time bound for factoring integers. *J. Amer. Math. Soc.*, 5(3):483–516, 1992.
[6] M. A. Morrison and J. Brillhart. A method of factoring and the factorization of $F_7$. *Math. Comp.*, 29:183–205, 1975. Collection of articles dedicated to Derrick Henry Lehmer on the occasion of his seventieth birthday.
[7] J. M. Pollard. Theorems on factorization and primality testing. *Proc. Cambridge Philos. Soc.*, 76:521–528, 1974.
[8] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In *Computational methods in number theory, Part I*, volume 154 of *Math. Centre Tracts*, pages 89–139. Math. Centrum, Amsterdam, 1982.
[9] F. Sica. Factoring with hints. *J. Math. Cryptol.*, 15(1):123–130, 2021.

Francesco Sica
Department of Mathematics and Statistics, Florida Atlantic University, 777 Glades Rd, Boca Raton, FL, 33431 USA.
*ORCID: 0000-0002-6027-2548*
*Email address*: sicaf@fau.edu