

A CERTIFIED LOWER BOUND FOR THE FIRST AUTOCORRELATION CONSTANT

TEJASVI SINGH TOMAR

ABSTRACT. Let C_1 be the largest constant such that $\sup_{|t| \leq 1/2} \int f(t-x)f(x) dx \geq C_1 \left(\int_{-1/4}^{1/4} f \right)^2$ for all nonnegative functions f on the real line. This constant governs the size of large generalized Sidon sets, and it is Problem 6.2 in the list of Georgiev, Gómez-Serrano, Tao and Wagner. The previous lower bounds include 1.28 by Cloninger and Steinerberger and 1.2937 by Kim and Pilanci. We prove $C_1 \geq 1.32$. Consequently, for all large g , the largest g -Sidon sets in $\{1, \dots, N\}$ have fewer than $1.2310\sqrt{gN}$ elements once N is large. The proof starts from a reduction of Cloninger and Steinerberger: for each n , C_1 is at least a finite min-max V_n of quadratic forms in the masses of f on n equal bins. We bound V_{30} from below by a spatial branch-and-bound over block masses. A linear relaxation with McCormick inequalities and square cuts closes the regions of the search, and every leaf carries an explicit dual vector. A checker written in Python evaluates the 24,935,952 leaf bounds in exact rational arithmetic. A second checker, written and proved sound in the Lean 4 proof assistant, accepts the same certificate by compiled evaluation. The bound $C_1 \geq 1.31$ also has a proof checked by the Lean kernel alone.

1. INTRODUCTION

For $f: \mathbb{R} \rightarrow [0, \infty]$ write $(f * f)(t) = \int_{\mathbb{R}} f(t-x)f(x) dx \in [0, \infty]$. Following [1], we call the largest constant C such that

$$(1.1) \quad \sup_{-1/2 \leq t \leq 1/2} (f * f)(t) \geq C \left(\int_{-1/4}^{1/4} f(x) dx \right)^2$$

for all measurable $f: \mathbb{R} \rightarrow [0, \infty]$ the *first autocorrelation constant* C_1 . It is Problem 6.2 of [5]. Restricting f to $[-1/4, 1/4]$ does not increase the left side, so C_1 is also the infimum of $\sup(f * f) / (\int f)^2$ over nonnegative f supported in $[-1/4, 1/4]$. Our proof also works with the essential supremum in place of the supremum. Schinzel and Schmidt [16] studied this quantity and conjectured that $C_1 = \pi/2$.

The constant comes from additive combinatorics. A set $A \subseteq \{1, \dots, N\}$ is a g -Sidon set if $|\{(a, b) \in A \times A : a + b = m\}| \leq g$ for all m . Cilleruelo, Ruzsa and Vinuesa [2] proved that the largest g -Sidon set in $\{1, \dots, N\}$ has between $\underline{\sigma}(g)\sqrt{gN}(1 - o(1))$ and $\bar{\sigma}(g)\sqrt{gN}(1 + o(1))$ elements as $N \rightarrow \infty$, where $\underline{\sigma}(g)$ and $\bar{\sigma}(g)$ both tend to a constant σ as $g \rightarrow \infty$. They also expressed σ through the continuous problem, and in our normalization $\sigma^2 = 2/C_1$ [3, Section 1.2].

Explicit step functions give the best upper bounds for C_1 . Matolcsi and Vinuesa [12] found $C_1 \leq 1.5098$, which disproved the conjecture of Schinzel and Schmidt. Computer searches [1, 5] and later programs have lowered the bound below 1.5029 [4]. Lower bounds, however, have moved slowly. Cloninger and Steinerberger list the earlier ones [3, Section 1.2], which end with the bound $C_1 \geq 1.2748$ of Matolcsi and Vinuesa [12] (quoted as 1.2749 in [3]). They reduced

2020 *Mathematics Subject Classification.* 42A85, 11B83; 90C26, 90C57, 68V05.

Key words and phrases. Autoconvolution, autocorrelation inequality, Sidon sets, branch and bound, linear programming duality, computer-assisted proof, Lean.

the problem to a finite min-max over mass vectors, replaced the simplex of mass vectors by a lattice, and proved $C_1 \geq 1.28$ by a large enumeration. Ivanisvili and Xie [7] sharpened an error term of that computation to obtain 1.2802. Kim and Pilanci [8] proved $C_1 \geq 1.2937$ with a semidefinite relaxation, and they checked its dual solution in interval arithmetic.

Theorem 1.1. $C_1 \geq \frac{33}{25}$. *That is, (1.1) holds with $C = 1.32$ for every measurable $f: \mathbb{R} \rightarrow [0, \infty]$.*

Corollary 1.2. *In the notation above, $\sigma \leq \sqrt{50/33} < 1.2310$.*

The method. Cloninger and Steinerberger observed that the masses $p_i = \int_{B_i} f$ of f on n equal bins B_i already force $f * f$ to be large somewhere. If $c_1 \leq i + j \leq c_2$, then $B_i + B_j$ lies in an interval J of length $(c_2 - c_1 + 2)/(2n)$, so $\sup_J f * f$ is at least the averaged mass $q_{c_1 c_2}(p)$ of (2.1). Hence $C_1 \geq V_n$, where V_n is the minimum over the simplex of the largest of these averages (Lemma 2.1). This inequality has no discretization error, and since we work on the continuous simplex, the lattice error that limited [3] does not arise. We prove $V_{30} \geq \frac{33}{25}$ (Theorem 2.2).

The function $p \mapsto \max_{c_1 \leq c_2} q_{c_1 c_2}(p)$ is a maximum of indefinite quadratic forms, but convex relaxations of the whole minimization are weak. The uniform mixture of the n vertices of Δ_n has second moment matrix $\frac{1}{n}I$, and for this matrix all averaged forms are at most 1. A relaxation that contains the convex hull of the points (p, pp^T) with $p \in \Delta_n$ therefore has value at most 1. We split the simplex into regions and relax on the regions separately. Following the multiscale idea of [3], the regions are boxes for the block masses m_D of a binary tree of bins (Section 3), so that coarse masses are resolved before fine ones. On a region we replace the products $m_D m_E$ by variables bounded by the four McCormick inequalities [11], and the window averages become linear. We tighten this relaxation by valid inequalities $(v_0 + v_D m_D + v_E m_E)^2 \geq 0$ in the spirit of Sherali and Fraticelli [15]. A linear program then bounds the maximum of the averages on the region from below, and its dual vector certifies the bound (Lemma 3.1).

Verification. The certificate (Table 1) consists of the search tree, one dual vector per leaf, and the square inequalities used in each subtree. We check it with a Python checker of about 700 lines, which needs only the standard library. It rebuilds the constraints from their definitions, recomputes the regions from the split decisions, evaluates the dual bounds in exact rational arithmetic, and checks that the leaves cover the region (2.2). The Python checker imports nothing from the search program, but the search builds its linear programs with the constraint definitions of the checker, so the two share these definitions.

We formalized the reduction of Section 2, the reversal symmetry and the soundness of a second checker in Lean 4 [9] with Mathlib [10]. The Lean checker re-implements the constraints of Section 3 and proves them valid. The formal theorem states (1.1) for measurable $f: \mathbb{R} \rightarrow [0, \infty]$ with lower Lebesgue integrals. We also derive in Lean the stronger statement with the essential supremum, and a version for measurable $f: \mathbb{R} \rightarrow \mathbb{R}$ that is nonnegative and vanishes outside $[-1/4, 1/4]$, stated with Mathlib’s convolution and the L^∞ norm on $[-1/2, 1/2]$. An exporter translates the job logs into Lean source. Since the formal theorem does not mention the certificate, the exporter need not be trusted. The Lean checker accepts the certificate through `native_decide`, which lets Lean accept the result of compiled code as a proof step, so the trusted base includes the compiler. The final theorem depends on Lean’s three standard axioms and on 290,309 further axioms that record the results of compiled evaluations, one per job (a subtree of the search, Section 4) and one for the root region. We also wrote a third checker for evaluation by the Lean kernel alone. It has checked the certificates of $V_{20} \geq 32/25$ and $V_{28} \geq 131/100$ of Section 5, with the duals rounded to multiples of 2^{-64} . Hence $C_1 \geq 1.31$ has a formal proof that depends only on Lean’s three standard axioms.

2. REDUCTION TO A FINITE PROBLEM

Fix $n \geq 1$ and put $h = 1/(2n)$. For $0 \leq i < n$ let $B_i = [-\frac{1}{4} + ih, -\frac{1}{4} + (i+1)h)$, and let

$$\Delta_n = \left\{ p \in \mathbb{R}^n : p_i \geq 0, \sum_i p_i = 1 \right\}.$$

For $0 \leq c_1 \leq c_2 \leq 2n - 2$ and $p \in \Delta_n$ define

$$(2.1) \quad q_{c_1 c_2}(p) = \frac{2n}{c_2 - c_1 + 2} \sum_{c_1 \leq i+j \leq c_2} p_i p_j,$$

where the sum runs over ordered pairs $(i, j) \in \{0, \dots, n-1\}^2$. Let $F_n(p) = \max_{c_1 \leq c_2} q_{c_1 c_2}(p)$ and $V_n = \min_{p \in \Delta_n} F_n(p)$.

Lemma 2.1 (Cloninger–Steinerberger). $C_1 \geq V_n$ for every $n \geq 1$.

Proof. Let f be measurable and nonnegative, and let $g = f \mathbf{1}_{[-1/4, 1/4)}$. Then $g * g \leq f * f$ pointwise and $\int g = \int_{-1/4}^{1/4} f$. If this integral is zero there is nothing to prove. If it is infinite, then $\int_{-1/2}^{1/2} g * g = (\int g)^2 = \infty$ by Tonelli's theorem, so the left side of (1.1) is infinite. Otherwise we scale so that $\int g = 1$. We write $g = \sum_i g_i$ with $g_i = g \mathbf{1}_{B_i}$ and put $p_i = \int g_i$, so that $p \in \Delta_n$. All terms of $g * g = \sum_{i,j} g_i * g_j$ are nonnegative, and $g_i * g_j$ vanishes outside $B_i + B_j \subseteq [-\frac{1}{2} + (i+j)h, -\frac{1}{2} + (i+j+2)h]$. For $c_1 \leq c_2$ let $J = [-\frac{1}{2} + c_1 h, -\frac{1}{2} + (c_2 + 2)h] \subseteq [-\frac{1}{2}, \frac{1}{2}]$. If $c_1 \leq i+j \leq c_2$, then $B_i + B_j \subseteq J$, and $\int_J g_i * g_j = p_i p_j$ by Tonelli's theorem. Hence

$$|J| \sup_J g * g \geq \int_J g * g \geq \sum_{c_1 \leq i+j \leq c_2} p_i p_j,$$

and $\sup_{|t| \leq 1/2} g * g \geq q_{c_1 c_2}(p)$ because $|J| = (c_2 - c_1 + 2)h$. We take the maximum over $c_1 \leq c_2$ and obtain $\sup g * g \geq F_n(p) \geq V_n$. The same argument works with the essential supremum on J , which also bounds $|J|^{-1} \int_J g * g$ from above. $\square$

The reversal $p_i \mapsto p_{n-1-i}$ maps the pair (c_1, c_2) to $(2n-2-c_2, 2n-2-c_1)$ and preserves $c_2 - c_1$, so it preserves F_n . One of p and its reversal satisfies $\sum_i (2i - n + 1)p_i \leq 0$. Therefore

$$(2.2) \quad V_n = \min \left\{ F_n(p) : p \in \Delta_n, \sum_i (2i - n + 1)p_i \leq 0 \right\}.$$

Theorem 2.2. $V_{30} \geq \frac{33}{25}$.

Theorem 1.1 follows from Lemma 2.1 and Theorem 2.2, and Corollary 1.2 follows from Theorem 1.1 and $\sigma^2 = 2/C_1$. Since our proof also gives (1.1) with the essential supremum, the corollary holds in either normalization. We prove Theorem 2.2 in Section 4.

3. REGIONS AND THEIR RELAXATIONS

3.1. Regions. Let $\mathcal{T}$ be the balanced binary tree of bin intervals. Its root is $\{0, \dots, n-1\}$, an interval with more than one bin has its two halves as children (the left half has $\lfloor \cdot/2 \rfloor$ bins), and its leaves are the single bins. For a block $D \in \mathcal{T}$ put $m_D = \sum_{i \in D} p_i$. A *region* is a set

$$R = \left\{ p \in \Delta_n : L_D \leq m_D \leq U_D \ (D \in \mathcal{T}), \sum_i (2i - n + 1)p_i \leq 0 \right\}$$

with rational bounds. The root region has $L_D = 0$ and $U_D = 1$. A region is split at a block D and a rational s with $L_D < s < U_D$ into the regions with U_D replaced by s and with L_D replaced by s . After a split the bounds are tightened by the rules $m_D = m_{D_1} + m_{D_2}$ for the children D_1, D_2 of D and $m_{\text{root}} = 1$, applied until they no longer change. A region whose tightened bounds cross is empty.

3.2. The relaxation. For p in a region R we consider the point $x(p)$ with the following coordinates: m_D for $D \in \mathcal{T}$; $Z_{ij} = p_i p_j$ for $i \leq j$, where Z_{ji} stands for Z_{ij} ; $Z_{DE} = m_D m_E$ for pairs of non-root blocks that are not both single bins (for two single bins, Z_{DE} stands for the corresponding Z_{ij}); the band masses $P_c = \sum_{i+j=c} p_i p_j$ over ordered pairs and their partial sums $G_c = P_0 + \dots + P_c$ for $0 \leq c \leq 2n - 2$; and $t = F_n(p)$. This point satisfies the following constraints.

- (i) Window constraints $\frac{2n}{c_2 - c_1 + 2}(G_{c_2} - G_{c_1 - 1}) - t \leq 0$ for $0 \leq c_1 \leq c_2 \leq 2n - 2$, with $G_{-1} = 0$.
- (ii) Band, tree and product constraints: the definitions of P_c and G_c , the identities $m_D = m_{D_1} + m_{D_2}$, and the identities $Z_{DE} = Z_{D_1 E} + Z_{D_2 E}$ (or $Z_{DD} = Z_{D_1 D_1} + 2Z_{D_1 D_2} + Z_{D_2 D_2}$).
- (iii) Simplex constraints $\sum_j Z_{ij} = p_i$ for each i , which hold because $\sum_j p_j = 1$. These are reformulation-linearization constraints in the sense of Sherali and Adams [14].
- (iv) The symmetry constraint $\sum_i (2i - n + 1)p_i \leq 0$.
- (v) McCormick constraints: for each pair of non-root blocks D, E , the four inequalities that state $(m_D - L_D)(m_E - L_E) \geq 0$, $(U_D - m_D)(U_E - m_E) \geq 0$, $(m_D - L_D)(U_E - m_E) \geq 0$ and $(U_D - m_D)(m_E - L_E) \geq 0$ after $m_D m_E$ is replaced by Z_{DE} .
- (vi) Square constraints: for integers v_0, v_D, v_E and distinct non-root blocks D, E , the inequality obtained from $(v_0 + v_D m_D + v_E m_E)^2 \geq 0$ by expanding the square and replacing the products of block masses by the corresponding variables Z .
- (vii) The bounds $L_D \leq m_D \leq U_D$, $0 \leq Z, P, G \leq 1$ and $0 \leq t \leq 3$, the last one whenever $F_n(p) \leq 3$.

Lemma 3.1. *Write finitely many of these constraints as $\ell_r \leq a_r \cdot x \leq u_r$ with $\ell_r, u_r \in \mathbb{Q} \cup \{\pm\infty\}$, and the bounds as $b^- \leq x \leq b^+$. Let y be a rational vector with $y_r \leq 0$ whenever $\ell_r = -\infty$ and $y_r \geq 0$ whenever $u_r = \infty$, and put $\rho = e_t - \sum_r y_r a_r$. If*

$$\text{LB}(y) = \sum_{y_r > 0} y_r \ell_r + \sum_{y_r < 0} y_r u_r + \sum_c \min(\rho_c b_c^-, \rho_c b_c^+) \geq \lambda$$

for some $\lambda \leq 3$, then $F_n(p) \geq \lambda$ for every $p \in R$.

Proof. Suppose $p \in R$ and $F_n(p) < \lambda$. Then $x = x(p)$ satisfies all constraints and bounds, and $t = \rho \cdot x + \sum_r y_r (a_r \cdot x) \geq \text{LB}(y) \geq \lambda$, a contradiction. $\square$

The quantity $\text{LB}(y)$ is the safe dual bound of Neumaier and Shcherbina [13], valid for any y with the stated signs, optimal or not.

4. THE COMPUTATION

4.1. The search. We solved the linear programs with the open-source solver HiGHS [6] in double precision. The solver need not be trusted, because a leaf enters the certificate only with its dual vector and the checkers evaluate $\text{LB}(y)$ exactly. At $n = 30$ the relaxation without square constraints has 10,038 rows and 1,889 columns. Window and McCormick constraints enter the linear programs lazily, when the current solution violates them. We obtain square constraints from an eigenvector of the matrix $\begin{pmatrix} 1 & m_D & m_E \\ m_D & Z_{DD} & Z_{DE} \\ m_E & Z_{DE} & Z_{EE} \end{pmatrix}$ for its smallest eigenvalue at the current solution, scaled and rounded to integers. When the linear program at a node does not close it, we give every block D the score $\sum_E |Z_{DE} - m_D m_E| (1 + 20 w_{DE} / \max w)$, where w_{DE} is the sum of the absolute duals of the pair's McCormick constraints. We split the block with the largest score at its mass in the optimal solution, clipped to the middle half of its interval. Splits are made at multiples of 2^{-30} , so that the search propagates bounds exactly in floating point, in agreement with the rational propagation in the checkers.

We distributed the tree over 290,308 subtrees, which we call *jobs*. One worker processes a job depth-first and delegates the unexplored nodes as new jobs when the job exceeds a node budget. Each job also lists the square constraints it uses. For all but 23,144 of the leaves, the record stores the window, tree, symmetry, simplex and McCormick duals as integers divided by a power of two, and the square-constraint duals as double-precision numbers. The checkers then choose the band and product duals by solving the triangular linear system that gives the P , G and Z_{DE} columns zero reduced cost. Since any values on equality constraints are admissible in Lemma 3.1, this choice affects only the size of the certificate. The remaining leaves store all duals as double-precision numbers, and the checkers read all stored numbers exactly.

Proof of Theorem 2.2. We check the jobs one at a time: for each job we recompute its root region from the splits on its path, follow its depth-first record, and recompute the child regions. A child whose tightened bounds cross would be empty and need no leaf, but no such child occurs. At a leaf we evaluate $\text{LB}(y)$ exactly and find $\text{LB}(y) \geq \frac{33}{25}$, so Lemma 3.1 gives $F_{30} \geq \frac{33}{25}$ on the leaf region. A final pass confirms three facts: every delegated subtree is the root of exactly one job, all jobs are reachable from the root job, and the delegation graph has no cycles. Because a delegated region and the root region of its job are recomputed from the same path, they coincide. The two children of a split cover their parent, so the leaf regions cover the region (2.2), and (2.2) gives $V_{30} \geq \frac{33}{25}$. The Python checker performs these steps on the archived certificate, and the Lean checker performs them again. $\square$

TABLE 1. The certificate of Theorem 2.2 and the cost of the computation.

Bins n and bound λ	30, $\frac{33}{25}$
Nodes of the search tree	49,871,903
Leaves	24,935,952
Jobs (subtrees)	290,308
Smallest exact leaf bound	1.3200000010...
LP rows (no squares); columns	10,038; 1,889
Archive	48,732,549,120 bytes
SHA-256	0aa4923334b4c089a7be957bc6297db4cd30d9f45b67a43fafb987eb0381d761
Search	54 min on 32 servers with 384 threads each
Python check	39 min, 375 processes on a 384-thread server
Lean check (<code>native_decide</code>)	67,740 modules, about 136 CPU-hours

5. REMARKS

Limits of the reduction. We list in Table 2 upper bounds for V_n , each from an explicit mass vector found by a local search from random starting points and evaluated exactly. Independent runs agree on all digits shown. We release such vectors for $2 \leq n \leq 24$ and for $n = 26, 28, \dots, 36$. For $2 \leq n \leq 19$ the value is below 1.2937, and $V_1 = 1$, so the reduction needs at least 20 bins to improve on [8]. Since $V_{30} \leq 1.331221$, Theorem 2.2 is within 0.0113 of the best bound available at $n = 30$.

Smaller certificates. Without square constraints we also certified $V_{20} \geq 32/25$ with 113,761 leaves, $V_{24} \geq 13/10$ with 2,801,076 leaves and $V_{28} \geq 131/100$ with 15,250,196 leaves. The Python checker for that format and the Lean checker, through `native_decide`, accepted all three. For V_{28} we ran the Python checker on the jobs in parallel on 32 machines and combined the per-job results, a step that includes the partition check. The certificate for $V_{20} \geq 32/25$ gives an

TABLE 2. Upper bounds for V_n from explicit mass vectors, rounded up.

n	$V_n \leq$	n	$V_n \leq$	n	$V_n \leq$	n	$V_n \leq$
12	1.243365	19	1.292579	23	1.308665	30	1.331221
16	1.271793	20	1.294443	24	1.312492	32	1.335788
17	1.281014	21	1.299660	26	1.320130	34	1.338779
18	1.283145	22	1.304949	28	1.325596	36	1.342202

independent proof of the bound $C_1 \geq 1.28$ of [3]. With the same branching rule, the square constraints reduced the number of leaves by a factor of 2.5 at $n = 20$ and 2.3 at $n = 24$.

DATA AVAILABILITY

The certificate, the search program, the checkers and the Lean formalization are archived at doi:10.5281/zenodo.23480605. The certificate is split there into 78 parts, which concatenate to the archive of Table 1. The certificate of $V_{20} \geq 32/25$ is archived there as well. The certificates of $V_{24} \geq 13/10$ and $V_{28} \geq 131/100$, of 11 and 81 gigabytes, are not archived. The programs and the Lean sources are also available at <https://github.com/tejasvi/autoconvolution-lower-bound>.

USE OF AI TOOLS

Generative AI tools were used to find and prove the results, to write the programs and the Lean formalization, and to draft the text. The author reviewed the arguments, the programs and the formal proofs, and takes responsibility for the content.

REFERENCES

- [1] A. Novikov et al., *AlphaEvolve: A coding agent for scientific and algorithmic discovery*, arXiv:2506.13131, 2025.
- [2] J. Cilleruelo, I. Z. Ruzsa and C. Vinuesa, *Generalized Sidon sets*, Adv. Math. **225** (2010), 2786–2807.
- [3] A. Cloninger and S. Steinerberger, *On suprema of autoconvolutions with an application to Sidon sets*, Proc. Amer. Math. Soc. **145** (2017), 3191–3200.
- [4] Together AI, *EinsteinArena state-of-the-art results*, <https://github.com/togethercomputer/EinsteinArena-new-SOTA>, 2026.
- [5] B. Georgiev, J. Gómez-Serrano, T. Tao and A. Z. Wagner, *Mathematical exploration and discovery at scale*, arXiv:2511.02864, 2025.
- [6] Q. Huangfu and J. A. J. Hall, *Parallelizing the dual revised simplex method*, Math. Program. Comput. **10** (2018), 119–142.
- [7] P. Ivanisvili and X. Xie, *Grokability in five inequalities*, arXiv:2605.05193, 2026.
- [8] S. Kim and M. Pilanci, *AI-assisted discovery of convex relaxations via dual agents*, Proc. 43rd Int. Conf. Machine Learning, PMLR **306**, 2026; arXiv:2606.31182.
- [9] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction – CADE 28, Lecture Notes in Comput. Sci. **12699**, Springer, 2021, 625–635.
- [10] The mathlib Community, *The Lean mathematical library*, in: Proc. 9th ACM SIGPLAN Int. Conf. Certified Programs and Proofs (CPP 2020), 2020, 367–381.
- [11] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs. I. Convex underestimating problems*, Math. Program. **10** (1976), 147–175.
- [12] M. Matolcsi and C. Vinuesa, *Improved bounds on the supremum of autoconvolutions*, J. Math. Anal. Appl. **372** (2010), 439–447.
- [13] A. Neumaier and O. Shcherbina, *Safe bounds in linear and mixed-integer linear programming*, Math. Program. **99** (2004), 283–296.
- [14] H. D. Sherali and W. P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM J. Discrete Math. **3** (1990), 411–430.

- [15] H. D. Sherali and B. M. P. Fraticelli, *Enhancing RLT relaxations via a new class of semidefinite cuts*, J. Global Optim. **22** (2002), 233–261.
- [16] A. Schinzel and W. M. Schmidt, *Comparison of L^1 - and L^∞ -norms of squares of polynomials*, Acta Arith. **104** (2002), 283–296.

ORCID: <https://orcid.org/0000-0003-4668-3639>