New Lower Bounds for Rows of d-Disjunct Matrices via Recursive Potentials

Xiaopeng Zhao*

***Abstract*—In nonadaptive combinatorial group testing, given n items with at most d positives, the goal is to identify them using as few pooled tests as possible. A $t \times n$ binary matrix represents the design, where rows are tests and columns are items. The matrix is d-disjunct if no column is contained in the Boolean union of any d others. Let $T(d)$ be the minimum t for which such a matrix exists with $n > t$. Shangguan and Ge proved $T(d) \geq \frac{15+\sqrt{33}}{24} d^2$ by counting private pairs (IEEE Trans. Inf. Theory, 62(12):7518–7521, 2016). In this paper, we strengthen their argument by introducing a column–deletion recurrence in which the light–heavy threshold varies with the recursive state $z = (n-t)/d^2$, rather than remaining fixed. This yields the improved bound $T(d) \geq 0.9283d^2 - O(d)$. The analytic core reduces to a first-order ODE, and a self-contained interval-arithmetic certificate verifies that the solution reaches the required contact point.**

***Index Terms*—Nonadaptive group testing, d-disjunct matrix, cover-free family, private pair, matching, recursive potential.**

I. Introduction

COMBINATORIAL group testing considers the following problem. There are n items, at most d of which are positive. A test may pool any set of items; its outcome only indicates whether the pool contains a positive item. In *nonadaptive group testing*, all tests are fixed before any outcome is observed. The tests can therefore be run in parallel. This is useful when tests are slow, when samples can be processed in parallel, or when an anomalous source must be located quickly.

Typical applications include Dorfman pooling and pathogen screening, DNA library and molecular-pool screening, detection of electrical shorts and chemical leaks, active-user identification in random-access communication, and pattern search in databases [1], [5]–[8]. These applications share the same combinatorial model. Rows correspond to tests, columns correspond to items, and a matrix entry is 1 when the item is included in the test. The same model also arises in network security. For example, clients or traffic sources can be pooled in advance; abnormal group responses can then locate a small number of application-layer denial-of-service attackers [9]. If an eavesdropper can observe test outcomes, the design can also protect the status of each item [10].

Let $M = (m_{xj})$ be a $t \times n$ binary matrix. Rows are indexed by $x \in [t]$, and columns by $j \in [n]$. Define the support of column j as

$$C_j := \mathrm{supp}_M(j) = \{x \in [t] : m_{xj} = 1\} \subseteq [t].$$

*X. Zhao is with the School of Information and Intelligent Science, Donghua University, Shanghai 201620, China. E-mail: zxp@dhu.edu.cn. (Corresponding author.)

We identify a column with its support. Write the ordered family of all columns as $\mathcal{C} = (C_1, \ldots, C_n)$. The matrix is *$d$-disjunct* if, for every i and every $J \subseteq [n] \setminus \{i\}$ with $|J| \leq d$,

$$C_i \nsubseteq \bigcup_{j \in J} C_j, \tag{1}$$

holds. Equivalently, for every target column C_i and every set of at most d other columns, some row has entry 1 in C_i and entry 0 in all the other columns. Hence the Boolean test outcomes uniquely identify every positive set of size at most d without ambiguity.

Define

$$T(d) := \min\big\{t : \text{there is a } t \times n \text{ } d\text{-disjunct matrix for some } n > t\big\}.$$

The condition $n > t$ means that the design requires fewer tests than testing each item individually. The early work of Kautz and Singleton on superimposed codes and the systematic theory of disjunctive codes developed by D'yachkov and Rykov form the coding-theoretic basis of this problem [2]–[4]. The survey of D'yachkov and Rykov records the following classical bound and attributes it to Bassalygo:

$$t(d,n) \geq \min\left\{\binom{d+2}{2}, n\right\}, \tag{2}$$

where $t(d,n)$ is the minimum number of rows in a d-disjunct matrix with n columns. Thus,

$$T(d) \geq \binom{d+2}{2} = \frac{1}{2}d^2 + O(d),$$

making $1/2$ the classical leading coefficient for nontrivial designs. Erdős, Frankl, and Füredi [11] later proved that $\liminf_{d\to\infty} T(d)/d^2 \geq 5/6$ and conjectured that $\lim_{d\to\infty} T(d)/d^2 = 1$. Conversely, affine planes give $T(d) \leq (d+1)^2$ whenever $d+1$ is a prime power. Thus, 1 is the natural limiting constant. Shangguan and Ge [12] used private pairs and the Erdős–Gallai matching theorem to improve the coefficient to

$$T(d) \geq \kappa_{\mathrm{SG}} d^2, \qquad \kappa_{\mathrm{SG}} = \frac{15+\sqrt{33}}{24} = 0.864356\ldots. \tag{3}$$

In this paper, we introduce a state variable into the same combinatorial recurrence. This allows the threshold between the light-column count and heavy-column deletion to depend on the current state. The goal is to improve the quadratic coefficient further. Our main result is presented as follows.

Theorem 1. *There is an absolute constant Γ such that, for every positive integer d,*

$$T(d) \geq \frac{9283}{10000}d^2 - \Gamma d. \tag{4}$$

In particular,

$$\liminf_{d\to\infty} \frac{T(d)}{d^2} \geq 0.9283 > \kappa_{\mathrm{SG}}.$$

II. Preparations

A. Notation

Let $\mathbb{N} = \{1, 2, \ldots\}$. For every nonnegative integer m, define

$$[m] := \{1, 2, \ldots, m\}, \qquad [0] := \varnothing.$$

We write $\binom{X}{2}$ for the set of all two-element subsets of X; consequently, $|\binom{X}{2}| = \binom{|X|}{2}$. The matrix M, the support family $\mathcal{C} = (C_1, \ldots, C_n)$, and the supports $C_j = \mathrm{supp}_M(j) \subseteq [t]$ are as defined above. We use $j \in [n]$ solely as a column index, and $x \in [t]$ as a row index or as an element of a support.

All graphs are finite and simple. For a graph G, let $V(G)$, $E(G)$, $e(G) = |E(G)|$, and $\nu(G)$ denote its vertex set, edge set, number of edges, and matching number, respectively.

Let $I \subseteq \mathbb{R}$ be an interval. A function $f : I \to \mathbb{R}$ is *Lipschitz continuous* if there exists a constant $L \geq 0$ such that

$$|f(x) - f(y)| \leq L|x - y| \qquad (x, y \in I).$$

Such an L is called a Lipschitz constant. The notation $f \in C^1(I)$ means that f is differentiable and f' is continuous; $f \in C^2(I)$ means that f is twice differentiable and f'' is continuous; and $f \in C^{1,1}(I)$ means that $f \in C^1(I)$ and f' is Lipschitz continuous. If I has endpoints, derivatives are understood in the interior and are assumed to extend continuously to the endpoints. For an open set $\mathcal{D} \subseteq \mathbb{R}^2$, the condition $F \in C^1(\mathcal{D})$ means that F and its first partial derivatives are continuous. In particular, F is uniformly Lipschitz in each variable on every compact rectangle contained in $\mathcal{D}$.

We call a pair of functions satisfying the light-column counting condition and the heavy-column deletion condition a *recursive potential pair*. Here $g(z)$ is the potential, and $a(z)$ is the column-weight threshold.

B. Deletion, Private Pairs, and Matchings

This section gives the combinatorial tools used in the global argument. A point $x \in [t]$ is called *private* if it belongs to exactly one column. A column is called *isolated* if it contains a private point. A pair of elements of a column is a *private pair* of that column if no other column contains that pair.

Lemma 1 (Removing isolated columns)**.** *Let M be a $t \times n$ d-disjunct matrix with $\eta = n - t > 0$. Repeatedly delete an isolated column and the row of one of its private points. The process ends with a $t_0 \times n_0$ d-disjunct matrix that has no isolated column and satisfies*

$$n_0 - t_0 = \eta, \qquad t_0 \leq t.$$

Proof. Let C be isolated, and let x be private to C. Delete column C and row x. Row x cannot have separated any remaining target column from any set of at most d other remaining columns, since every remaining column has entry 0 in that row. Hence the new matrix is still d-disjunct. Each deletion decreases both t and n by one, so $n - t = \eta$ remains unchanged. □

Lemma 2 (Minimum weight of a nonisolated column)**.** *Every column C in a d-disjunct matrix with no isolated column satisfies $|C| \geq d + 1$.*

Proof. For each $x \in C$, the point x is not private, so there exists a column $C_x \neq C$ containing x. If $|C| \leq d$, then the family $\{C_x : x \in C\}$ contains at most $|C|$ columns and covers C. Removing duplicates can only reduce its size. This contradicts (1). □

Lemma 3 (Column-deletion recurrence)**.** *Let C be a column of weight w in a $t \times n$ d-disjunct matrix. Delete C and all w rows in its support. The remaining matrix has size $(t-w)\times(n-1)$ and is $(d-1)$-disjunct. If $\eta = n - t > 0$ in the original matrix, then the new matrix satisfies*

$$\eta_1 := (n-1) - (t-w) = \eta + w - 1 > 0.$$

Proof. Suppose that the remaining matrix is not $(d-1)$-disjunct. Then there exist a remaining target column A and at most $d-1$ other remaining columns $A_1, \ldots, A_r$ such that

$$A \setminus C \subseteq \bigcup_{i=1}^{r} (A_i \setminus C).$$

It follows that $A \subseteq C \cup A_1 \cup \cdots \cup A_r$. Thus, at most d other columns cover a column of the original matrix, a contradiction. The formula for η_1 is immediate. Since $\eta \geq 1$ and $w \geq 1$, we also have $\eta_1 > 0$. □

Lemma 4 (Erdős–Gallai matching theorem)**.** *Let G be a simple graph on N vertices. If $\nu(G) \leq \mu$ and $N \geq 2\mu + 1$, then*

$$e(G) \leq \max\left\{ \binom{2\mu+1}{2}, \binom{\mu}{2} + \mu(N-\mu) \right\}. \tag{5}$$

Proof. See [13, Theorem 4.1]. □

Lemma 5 (Lower bound for private pairs)**.** *Suppose that M has no isolated column. Let C be a column with $|C| = d + s$, where $1 \leq s \leq d-1$. Let $P(C)$ be the set of private pairs in C. Then*

$$|P(C)| \geq \min\left\{ \binom{d+1}{2}, \binom{d+s}{2} - \binom{2s-1}{2} \right\}. \tag{6}$$

Moreover, for every threshold $\theta \in [5/3, 2)$, the condition $|C| \leq \theta d$ implies

$$|P(C)| \geq \frac{\pi_2(\theta)}{2}d^2, \qquad \pi_2(\theta) := -3\theta^2 + 8\theta - 4. \tag{7}$$

Proof. Define a graph G_C on vertex set C. A pair xy is an edge precisely when some other column also contains both x and y. Thus, the edges of G_C are exactly the nonprivate pairs in C.

We first show that $\nu(G_C) \leq s - 1$. Suppose that G_C has s disjoint edges. For each edge, choose a column (other than C) that contains both endpoints. This uses at most s columns.

For each of the remaining $d-s$ points of C, choose another column (different from C) that contains it. Such a column exists because no column is isolated. In total, at most $s+(d-s)=d$ other columns cover C. This contradicts d-disjunctness.

Apply Lemma 4 with $N=d+s$ and $\mu=s-1$. We obtain

$$e(G_C) \le \max\left\{ \begin{array}{c} \binom{2s-1}{2}, \\ \binom{s-1}{2}+(s-1)(d+1) \end{array} \right\}.$$

The second term equals

$$\binom{d+s}{2}-\binom{d+1}{2}.$$

Private and nonprivate pairs partition $\binom{C}{2}$. Subtracting the above maximum from $\binom{d+s}{2}$ gives (6).

It remains to prove (7). The function π_2 is strictly decreasing on $[5/3,2]$, and

$$\pi_2(5/3)=1, \qquad \pi_2(2)=0.$$

If $s \le (2d+2)/3$, the smaller lower bound in (6) is $\binom{d+1}{2}$. Hence,

$$|P(C)| \ge \binom{d+1}{2} \ge \frac{d^2}{2} \ge \frac{\pi_2(\theta)d^2}{2}.$$

If $s \ge (2d+2)/3$, set $\vartheta=(d+s)/d \le \theta$. Direct expansion gives

$$\begin{aligned}\binom{d+s}{2}-\binom{2s-1}{2} &= \frac{d^2+2ds-3s^2-d+5s-2}{2} \\ &= \frac{\pi_2(\vartheta)d^2}{2}+\frac{-d+5s-2}{2}.\end{aligned}$$

Here $-d+5s-2 \ge (7d+4)/3 > 0$, and $\pi_2(\vartheta) \ge \pi_2(\theta)$. Thus, the second lower bound is also at least $\pi_2(\theta)d^2/2$. The two cases prove the claim. □

Corollary 1 (Global private-pair count)**.** *Let M be a $t \times n$ matrix with no isolated column. If every column has weight at most θd, where $\theta \in [5/3,2)$, then*

$$t(t-1) \ge n\,\pi_2(\theta)d^2. \tag{8}$$

Proof. Private pairs belonging to different columns are distinct. Therefore,

$$\binom{t}{2} \ge \sum_{i=1}^{n} |P(C_i)| \ge n\frac{\pi_2(\theta)d^2}{2}.$$

Multiplying by 2 gives (8). □

III. A Global Recursive Potential and the 0.9283 Bound

Let the excess number of columns be

$$\eta := n-t > 0, \qquad z := \frac{\eta}{d^2}.$$

We seek a nondecreasing function $g(z)$ such that every such matrix satisfies $t \ge d^2 g(z) - O(d)$. The light-column case is controlled by Corollary 1. In the heavy-column case, Lemma 3 maps (d,η) to $(d-1,\eta+w-1)$. The next lemma combines these two mechanisms.

Proposition 1 (Recursive potential principle)**.** *Let $g : [0,\infty) \to (0,1)$ be nondecreasing, bounded, and in $C^{1,1}$. Let $a : [0,\infty) \to [5/3,2)$ be continuous. Suppose that, for every $z \ge 0$,*

$$\pi_2(a(z)) = \frac{g(z)^2}{g(z)+z}, \tag{9}$$

$$g'(z)\big(a(z)+2z\big) \ge 2g(z)-a(z). \tag{10}$$

Assume also that some $z_* \ge 0$ makes g constant on $[z_*,\infty)$. Choose $L_g \ge 0$ such that $|g'(x)-g'(y)| \le L_g|x-y|$ for all $x,y \ge 0$. Define

$$\Gamma_{\mathrm{rec}} := \|g\|_\infty + (1+z_*)\|g'\|_\infty + 8L_g(1+z_*)^2, \tag{11}$$

where both supremum norms are over $[0,\infty)$. Then every real $\Gamma \ge \Gamma_{\mathrm{rec}}$, every positive integer d, and every $t \times n$ d-disjunct matrix with $\eta = n-t > 0$ satisfy

$$t \ge d^2 g\Big(\frac{\eta}{d^2}\Big) - \Gamma d. \tag{12}$$

In particular, $T(d) \ge g(0)d^2 - O(d)$.

If $g(z) \equiv c$ is constant, then the exact bound

$$T(d) \ge cd^2 \qquad (d \ge 1). \tag{13}$$

holds without a linear error term.

Proof. First apply Lemma 1. The resulting matrix has the same d and the same $\eta = n-t$, and it has no more rows. It is therefore enough to consider matrices with no isolated column.

We use induction on d. If $d=1$, then $\Gamma \ge \Gamma_{\mathrm{rec}} \ge \|g\|_\infty$ gives

$$g(\eta) - \Gamma \le \|g\|_\infty - \Gamma \le 0 \le t,$$

so (12) holds. Now let $d \ge 2$ and assume the result for $d-1$. At the current state, write

$$z = \frac{\eta}{d^2}, \qquad g_z := g(z), \qquad a_z := a(z),$$

and suppose, for a contradiction, that

$$t < d^2 g_z - \Gamma d < d^2 g_z. \tag{14}$$

Light-column case. Suppose that every column has weight at most $a_z d$. The function $x \mapsto x^2/(x+\eta)$ is strictly increasing for $x > 0$, since its derivative is $x(x+2\eta)/(x+\eta)^2 > 0$. Equations (14) and (9) give

$$\frac{t(t-1)}{t+\eta} < \frac{t^2}{t+\eta} < \frac{(d^2 g_z)^2}{d^2 g_z+\eta} = d^2\frac{g_z^2}{g_z+z} = d^2\pi_2(a_z),$$

which contradicts the bound $t(t-1)/(t+\eta) \ge d^2\pi_2(a_z)$ from Corollary 1.

Heavy-column case. There must therefore be a column of weight $w > a_z d$. Delete this column and all rows in its support. By Lemma 3, the new parameters are

$$d_1 = d-1, \qquad \eta_1 = \eta + w - 1, \qquad z_1 = \frac{\eta_1}{(d-1)^2} > z.$$

The induction hypothesis gives

$$t - w \ge (d-1)^2 g(z_1) - \Gamma(d-1). \tag{15}$$

We next prove the one-step error bound

$$w + (d-1)^2 g(z_1) - d^2 g(z) \ge -\Gamma_{\mathrm{rec}}. \tag{16}$$

If $z \ge z_*$, then $g(z_1) = g(z) =: g_*$. Equation (10) and $g'(z) = 0$ imply $a_z \ge 2g_*$. Hence,

$$\begin{aligned} w + (d-1)^2 g_* - d^2 g_* &= w - (2d-1)g_* \\ &> a_z d - (2d-1)g_* \\ &\ge g_* > 0. \end{aligned}$$

Now suppose that $0 \le z < z_*$. If $w \ge 2d$, then $g(z_1) \ge g(z)$ and $g(z) < 1$ yield

$$w + (d-1)^2 g(z_1) - d^2 g(z) \ge w - (2d-1)g(z) > 0.$$

It remains to consider $w < 2d$. Set

$$\Delta := z_1 - z = \frac{w - 1 + (2d-1)z}{(d-1)^2} > 0. \tag{17}$$

Since g' is Lipschitz, Taylor's inequality with a one-sided remainder gives

$$g(z_1) \ge g(z) + g'(z)\Delta - \frac{L_g}{2}\Delta^2.$$

Let $\omega = w/d$. Substitute the last inequality and separate the terms of order d from the constant terms. This gives

$$\begin{aligned} &w + (d-1)^2 g(z_1) - d^2 g(z) \\ &\quad \ge d\big[\omega - 2g_z + g'(z)(\omega + 2z)\big] + g_z - g'(z)(1+z) \\ &\qquad - \frac{L_g}{2}\left(\frac{w - 1 + (2d-1)z}{d-1}\right)^2. \end{aligned} \tag{18}$$

The expression in the first pair of brackets equals

$$(\omega - a_z)(1 + g'(z)) + \big[a_z - 2g_z + g'(z)(a_z + 2z)\big].$$

We have $\omega > a_z$ and $g'(z) \ge 0$. The second pair of brackets is nonnegative by (10). Thus, the term of order d is nonnegative. Moreover, $d \ge 2$, $w < 2d$, and $z < z_*$ imply

$$0 < \frac{w - 1 + (2d-1)z}{d-1} < \frac{2d(1+z_*)}{d-1} \le 4(1+z_*).$$

Therefore, the right-hand side of (18) is at least

$$-(1+z_*)\|g'\|_\infty - 8L_g(1+z_*)^2 \ge -\Gamma_{\mathrm{rec}},$$

so (16) holds in every case.

Finally, add (15) and (16):

$$\begin{aligned} t &\ge w + (d-1)^2 g(z_1) - \Gamma(d-1) \\ &\ge d^2 g(z) - \Gamma d + (\Gamma - \Gamma_{\mathrm{rec}}) \\ &\ge d^2 g(z) - \Gamma d, \end{aligned}$$

contrary to (14). This completes the induction.

It remains to prove the exact statement for constant g. Let $g(z) \equiv c$. Equation (10) gives $a(z) \ge 2c$. We use induction on d again. If $d = 1$, then $t \ge 1 > c$. Assume the statement for $d-1$. By Lemma 1, it is enough to consider a d-disjunct matrix with no isolated column. Let $z = \eta/d^2$. If every column has weight at most $a(z)d$, then Corollary 1 and (9) give

$$\frac{t(t-1)}{t+\eta} \ge d^2 \pi_2(a(z)) = \frac{c^2 d^4}{cd^2 + \eta}.$$

If $t < cd^2$, the strict monotonicity of $x \mapsto x^2/(x+\eta)$ gives

$$\frac{t(t-1)}{t+\eta} < \frac{t^2}{t+\eta} < \frac{c^2 d^4}{cd^2 + \eta},$$

a contradiction. If some column has weight $w > a(z)d$, then $w > 2cd$. After deleting this column and its rows, the induction hypothesis gives

$$t - w \ge c(d-1)^2,$$

and hence

$$t > 2cd + c(d-1)^2 = cd^2 + c > cd^2.$$

Both cases give $t \ge cd^2$. This proves (13). □

The Shangguan–Ge bound [12] corresponds to a constant threshold in the state recurrence. Let

$$\kappa := \kappa_{\mathrm{SG}} = \frac{15 + \sqrt{33}}{24}.$$

Direct substitution gives

$$\pi_2(2\kappa) = \kappa, \qquad 12\kappa^2 - 15\kappa + 4 = 0. \tag{19}$$

This constant also follows directly from the conditions for a recursive potential. Suppose that $g(z) \equiv c$. At $z = 0$, the light-column condition gives

$$a(0) = \frac{4 + \sqrt{4 - 3c}}{3}.$$

Because $g'(0) = 0$, the heavy-column condition reduces to $a(0) \ge 2c$. Define

$$q(c) := \frac{4 + \sqrt{4 - 3c}}{3} - 2c \qquad (0 < c < 1).$$

The first term is strictly decreasing, whereas $2c$ is strictly increasing. Thus, q is strictly decreasing. Moreover, $q(5/6) > 0$ and $q(1) < 0$. Therefore, q has exactly one zero in $(5/6, 1)$, and the heavy-column condition holds on this interval precisely when c does not exceed that zero. Hence, the largest admissible constant is determined by $a(0) = 2c$. This equation is equivalent to

$$12c^2 - 15c + 4 = 0.$$

Its larger root is $c = \kappa = (15 + \sqrt{33})/24$. This root lies in $(5/6, 1)$, so it is the unique zero described above.

Corollary 2 (The Shangguan–Ge theorem [12] as a constant-t-threshold case)**.** *For every positive integer d,*

$$T(d) \ge \kappa_{\mathrm{SG}} d^2.$$

Proof. Define

$$\begin{aligned} &g_{\mathrm{SG}}(z) := \kappa, \\ &a_{\mathrm{SG}}(z) := \frac{4 + \sqrt{4 - 3\kappa^2/(\kappa + z)}}{3}, \qquad z \ge 0. \end{aligned} \tag{20}$$

These formulas explicitly construct the required potential and threshold functions. Substituting (20) into $\pi_2(v) = -3v^2 + 8v - 4$ gives

$$\pi_2(a_{\mathrm{SG}}(z)) = \frac{\kappa^2}{\kappa + z} = \frac{g_{\mathrm{SG}}(z)^2}{g_{\mathrm{SG}}(z) + z},$$

so the light-column condition (9) holds. The function $z \mapsto \kappa^2/(\kappa + z)$ is decreasing. Hence, the expression under the

square root increases with z, and a_{SG} is nondecreasing. Because π_2 is strictly decreasing on $[5/3, 2)$, (19) also gives

$$a_{\mathrm{SG}}(0) = 2\kappa, \qquad a_{\mathrm{SG}}(z) \geq 2\kappa.$$

Since $g'_{\mathrm{SG}}(z) = 0$, the last inequality is exactly the heavy-column condition (10). In addition, g_{SG} is a constant $C^{1,1}$ function, and a_{SG} is continuous with values in $[2\kappa, 2) \subset [5/3, 2)$. The exact constant-potential conclusion (13) of Proposition 1 therefore yields $T(d) \geq \kappa d^2$. □

To raise the leading constant to $c_0 = 0.9283 > \kappa$, a constant potential is no longer sufficient. Indeed, if $u_0 := \pi_2^{-1}(c_0)$, then $u_0 < 2c_0$, so the heavy-column condition fails. We instead let g grow with the state z and impose equality in the heavy-column condition. We do this until $2g = a$, and then keep g constant. We first construct a parametric orbit with independent variable u. A strictly increasing state map then converts this orbit into functions of z.

The function π_2 decreases strictly from 1 to 0 on $[5/3, 2)$. For $0 < r \leq 1$, its inverse is

$$\pi_2^{-1}(r) = \frac{4 + \sqrt{4 - 3r}}{3}. \tag{21}$$

For a general initial value $c \in (0, 1)$, set

$$u_0 := \pi_2^{-1}(c) = \frac{4 + \sqrt{4 - 3c}}{3}, \tag{22}$$

and consider the following differential equation in the independent variable u:

$$y'(u) = F(u, y(u)), \qquad y(u_0) = c, \tag{23}$$

where

$$F(u, v) := \frac{v(4 - 3u)(2v - u)}{\pi_2(u)(v - u)}. \tag{24}$$

Lemma 6. *On the open set*

$$\mathcal{D} := \{(u, v) : 5/3 < u < 2,\ 0 < v < 1,\ v < u\}$$

F is C^1 in (u, v). If $(u_0, c) \in \mathcal{D}$, then (23) *has a unique maximal C^1 solution. Equivalently, this solution satisfies the integral equation*

$$y(u) = c + \int_{u_0}^{u} F(\xi, y(\xi))\ \mathrm{d}\xi \tag{25}$$

and is constructed locally by this equation. It can be continued as long as its trajectory stays in a compact subset of $\mathcal{D}$.

Proof. For $(u, v) \in \mathcal{D}$, we have $\pi_2(u) > 0$ and $v - u < 0$. Thus, neither denominator in (24) vanishes. It follows that $F \in C^1(\mathcal{D})$. In particular, F is locally Lipschitz in its second variable. The Picard–Lindelöf theorem gives local existence, uniqueness, and the integral equation (25). The standard continuation theorem for ordinary differential equations gives the final claim [14, Chap. 1]. Thus, every hypothesis of these theorems follows from the openness of $\mathcal{D}$ and the fact that $F \in C^1(\mathcal{D})$. □

Let y be this solution. On any interval where $2y(u) - u > 0$, define the *state orbit*

$$Z(u) := \frac{y(u)^2}{\pi_2(u)} - y(u). \tag{26}$$

Since $\pi_2'(u) = 2(4 - 3u)$, differentiation followed by substitution from (23) gives

$$\begin{aligned} Z'(u) &= y'(u)\left(\frac{2y(u)}{\pi_2(u)} - 1\right) - \frac{y(u)^2 \pi_2'(u)}{\pi_2(u)^2} \\ &= \frac{u + 2Z(u)}{2y(u) - u}\, y'(u). \end{aligned} \tag{27}$$

The second equality is an algebraic identity. After both sides are multiplied by $2\pi_2(u)^2(y(u) - u)$, their numerators are both equal to

$$y(u)\pi_2'(u)\big[2y(u)^2 - 2y(u)\pi_2(u) + u\pi_2(u)\big].$$

We start from $Z(u_0) = 0$ and consider the orbit before it reaches $2y = u$. On this part of the orbit, $4 - 3u < 0$, $y - u < 0$, and $2y - u > 0$. Hence, $y' > 0$. We also have $\pi_2(u) < \pi_2(u_0) = c = y(u_0) < y(u)$, and therefore $Z(u) > 0$. Equation (27) now gives $Z'(u) > 0$. Thus, Z has an inverse. Set $a(z) := Z^{-1}(z)$ and define

$$g(z) := y(a(z)), \tag{28}$$

Equation (26) immediately gives equality in the light-column condition. The chain rule gives

$$g'(z) = \frac{y'(u)}{Z'(u)} = \frac{2g(z) - a(z)}{a(z) + 2z}. \tag{29}$$

Here the parameter on the right-hand side is $u = a(z)$. Hence, the functions g and a constructed from the parametric ODE satisfy equality in the heavy-column condition.

Proposition 2. *Let*

$$c_0 := \frac{9283}{10000}, \qquad u_0 := \frac{4 + \sqrt{4 - 3c_0}}{3}.$$

The unique solution of (23) *with $y(u_0) = c_0$ extends to $[u_0, 1.992]$. Moreover, for some $u_* \in (u_0, 1.992)$, it satisfies for the first time*

$$2y(u_*) = u_* < 2, \qquad y(u_*) < 0.996 < 1.$$

Proof. First, exact rational arithmetic verifies that

$$\begin{aligned} \pi_2\left(\frac{170077191474675}{10^{14}}\right) &> \frac{9283}{10000} \\ &> \pi_2\left(\frac{170077191474677}{10^{14}}\right). \end{aligned}$$

By the strict monotonicity of π_2,

$$1.70077191474675 < u_0 < 1.70077191474677, \tag{30}$$

and therefore $2c_0 - u_0 > 0.15582808525322 > 0$.

The program in Appendix uses IEEE 754 binary64 arithmetic [15]. After every operation on an endpoint, the program uses `nextafter` to move the lower endpoint to the adjacent floating-point number toward $-\infty$ and the upper endpoint toward $+\infty$. The distance between adjacent floating-point numbers at a given scale is called one unit in the last place (ulp). The input constants are first represented exactly by `Fraction`. Consequently, every interval produced by the program contains the corresponding exact real value.

Use the grid

$$u_0 = u^{(0)} < u^{(1)} < \cdots < u^{(N)} = 1.992,$$
$$h_j := u^{(j+1)} - u^{(j)} \le 10^{-6}.$$

The first step connects u_0 to the first 10^{-6} grid point on its right. Every later step has length 10^{-6}. Suppose that $y(u^{(j)}) \in Y_j$, and set

$$B_j := Y_j + [-h_j, h_j], \qquad R_j := [u^{(j)}, u^{(j+1)}] \times B_j,$$

Interval arithmetic gives $F(R_j) \subseteq I_j \subset (-1, 1)$. At every step, the program also checks that $R_j \subset \mathcal{D}$. Thus, neither denominator contains zero. Suppose, for a contradiction, that the exact solution first leaves B_j at time $\widehat{u}$. Then

$$|y(\widehat{u}) - y(u^{(j)})| < \widehat{u} - u^{(j)} \le h_j.$$

This displacement is too small for the solution to leave $Y_j + [-h_j, h_j]$, a contradiction. Hence, the exact solution remains in R_j throughout the step. The integral equation gives

$$\begin{aligned} y(u^{(j+1)}) &= y(u^{(j)}) + \int_{u^{(j)}}^{u^{(j+1)}} F(\xi, y(\xi))\, \mathrm{d}\xi \\ &\in Y_j + h_j I_j =: Y_{j+1}. \end{aligned} \tag{31}$$

This enclosure bounds the exact integral. It is not an Euler point-value approximation that discards a truncation error. Appendix A gives the implementation details.

The verification program completes 291229 grid steps and returns

$$\begin{aligned} y(1.992) &\in [0.99547122321526016, \\ &\quad\ 0.99561537499907371], \\ 2y(1.992) - 1.992 &\in [-0.0010575535694803408, \\ &\quad\ -0.00076925000185168446]. \end{aligned}$$

The sign is strictly positive at the initial point and strictly negative at the endpoint. The intermediate value theorem therefore gives a zero. Let u_* be the smallest zero. Then $2y(u) - u > 0$ on $[u_0, u_*)$, and $y(u_*) = u_*/2 < 0.996$. □

Fig. 1 shows the solution orbit used in the construction. Its coordinates were obtained by high-precision numerical integration and only illustrate the shape of the solution. The rigorous proof is the interval enclosure above. Numerically, the first contact point is approximately

$$(u_*, y(u_*)) \approx (1.9911410763,\ 0.9955705382).$$

Proposition 3 (Explicit construction of the recursive potential)**.** *Let y and u_* be as in Proposition 2, and define*

$$g_* := y(u_*), \qquad z_* := Z(u_*) = \frac{g_*^2}{\pi_2(u_*)} - g_*.$$

For $0 \le z \le z_*$, define $a(z) = Z^{-1}(z)$ and $g(z) = y(a(z))$ as in (28)*. For $z \ge z_*$, define*

$$g(z) := g_*, \qquad a(z) := \frac{4 + \sqrt{4 - 3g_*^2/(g_* + z)}}{3}. \tag{32}$$

Then g is nondecreasing, bounded, takes values in $(0, 1)$, and belongs to $C^{1,1}$. The function a is continuous and takes values in $[5/3, 2)$. Together, they satisfy (9)*–*(10)*.*

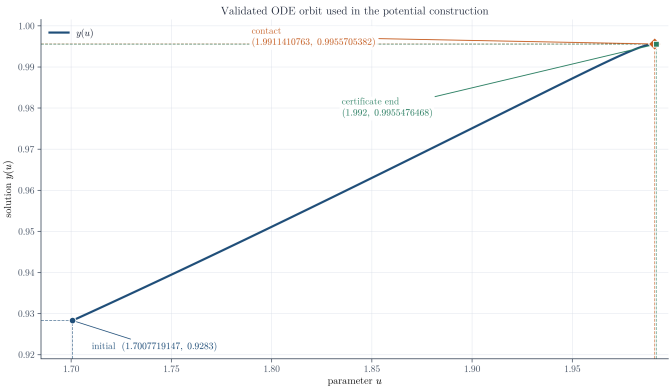

Fig. 1. Numerical plot of the parametric solution $y(u)$ on $[u_0, 1.992]$. The three markers denote the initial point, the first contact point where $2y = u$, and the certificate endpoint. Dashed lines project the marked coordinates onto the two axes.

Proof. We already know that $Z' > 0$ on $[u_0, u_*)$. At the endpoint, use the first line of (27), rather than its $0/0$ form. Since $y'(u_*) = 0$ and $\pi_2'(u_*) < 0$, we have

$$Z'(u_*) = -\frac{g_*^2 \pi_2'(u_*)}{\pi_2(u_*)^2} > 0.$$

Thus, Z is strictly increasing on the closed interval and has a C^1 inverse. The pre-contact construction is therefore well defined. By (29), both boundary conditions hold with equality on this part. Moreover, $\lim_{z \to z_*^-} g'(z) = 0$.

On the constant segment, (32) is the explicit solution of $\pi_2(a) = g_*^2/(g_* + z)$ obtained from (21). Hence, equality in the light-column condition still holds. The resulting $a(z)$ starts from $u_* = 2g_*$, increases continuously, and tends to 2. Thus, $a(z) \ge 2g_*$. Since $g' = 0$ on this segment, the heavy-column inequality also holds.

Finally, y and Z are smooth on the pre-contact segment, and Z' has a positive lower bound on this compact interval. Hence, g is C^2 on $[0, z_*]$ and has a bounded second derivative there. Its left and right derivatives at z_* are both zero, and it is constant to the right of z_*. Therefore, the joined derivative g' is globally Lipschitz, so $g \in C^{1,1}$. The remaining range bounds and monotonicity properties follow directly from the construction and from $g_* < 0.996 < 1$. □

The numerical orbit gives

$$\begin{aligned} z_* &\approx 27.1622012753, \\ g_* &\approx 0.9955705382, \\ a(z_*) = u_* &\approx 1.9911410763. \end{aligned}$$

Figs. 2 and 3 show g and a, respectively. Both are plotted on $0 \le z \le 40$ to display the behavior before and after contact. Their actual domains remain $[0, \infty)$. The decimal coordinates are only illustrative and are not used in the proof.

Proof of Theorem 1. Let (g, a) be the pair constructed in Proposition 3. It satisfies every hypothesis of Proposition 1, and $g(0) = y(u_0) = c_0 = 9283/10000$. Choose any $\Gamma \ge \Gamma_{\mathrm{rec}}$. Every d-disjunct matrix with $n > t$ satisfies

$$t \ge d^2 g\left(\frac{n-t}{d^2}\right) - \Gamma d \ge d^2 g(0) - \Gamma d = \frac{9283}{10000} d^2 - \Gamma d.$$

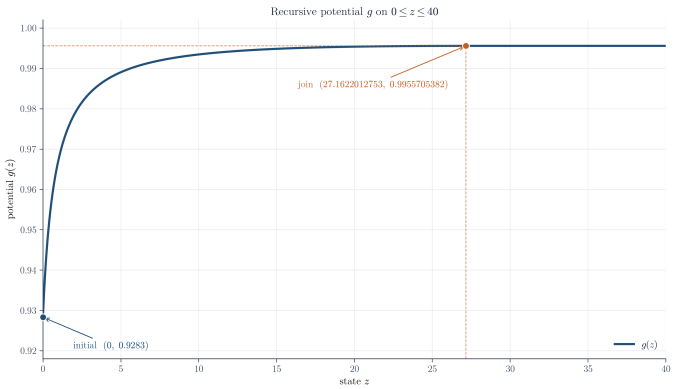

Fig. 2. Numerical plot of the potential $g(z)$. It increases monotonically from $g(0) = 0.9283$ to g_* at $z = z_*$ and remains constant afterward. Dashed lines mark the coordinates of the initial and joining points.

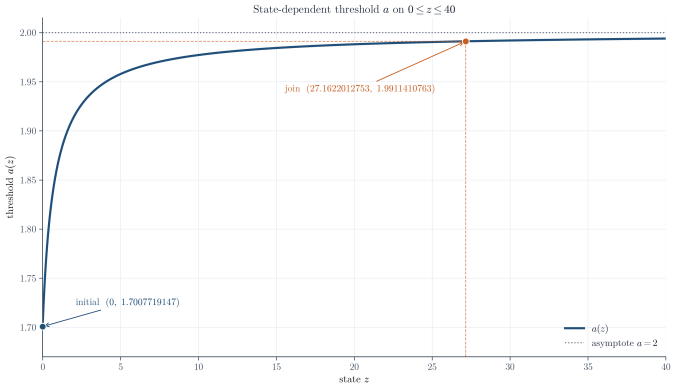

Fig. 3. Numerical plot of the threshold $a(z)$. It increases from $a(0) = u_0 \approx 1.7007719147$ to $a(z_*) = u_*$ and then continues to increase toward the horizontal asymptote $a = 2$. Dashed lines mark the coordinates of the initial and joining points.

Minimizing t over all such matrices gives (4). The corollary in Corollary 2 retains an exact finite-parameter bound, whereas the present theorem improves the asymptotic leading constant by $0.9283 - \kappa_{\mathrm{SG}} = 0.063943\ldots$. □

Remark 1 (An explicit upper bound for Γ). We show as follows that, for the pair constructed in Proposition 3, the main theorem holds with the coarse but explicit upper bound $\Gamma = 2859$. By Proposition 2, we have $g_* < 0.996 = 249/250$ and $u_* < 1.992 = 249/125$. Since π_2 is decreasing on $[5/3, 2)$,

$$z_* < \frac{(249/250)^2}{\pi_2(249/125)} = \frac{62001}{1988} < 32.$$

Before contact, (29) together with $g < 1$, $a \ge 5/3$, and $z \ge 0$ gives

$$0 \le g'(z) = \frac{2g(z) - a(z)}{a(z) + 2z} \le \frac{1}{5};$$

after contact, $g' = 0$. Thus, $\|g'\|_\infty \le 1/5$.

We next bound the Lipschitz constant of g'. Along the parametric orbit, we have $y' \ge 0$, $\pi_2(u) \le y(u)$, and $u \ge u_0 > 17/10$. The first line of (27) gives

$$Z'(u) \ge -\frac{y(u)^2 \pi_2'(u)}{\pi_2(u)^2} \ge -\pi_2'(u) = 6u - 8 \ge \frac{11}{5}.$$

Hence $0 < a'(z) = 1/Z'(u) \le 5/11$ before contact. Differentiating (29) and using $2g - a = g'(a + 2z)$, we obtain

$$g''(z) = -\frac{a'(z)(1 + g'(z))}{a(z) + 2z}, \qquad |g''(z)| \le \frac{18}{55}.$$

The function g' is continuous across the contact point and vanishes after contact. Therefore, we may take $L_g = 18/55$ as a Lipschitz constant for g'. Substituting this value into (11), together with $\|g\|_\infty < 1$ and $1 + z_* < 33$, yields

$$\Gamma_{\mathrm{rec}} < 1 + \frac{33}{5} + 8 \cdot \frac{18}{55} \cdot 33^2 = \frac{14294}{5} < 2859.$$

IV. Conclusion

We have proved that

$$T(d) \ge 0.9283 d^2 - O(d).$$

The proof uses only two basic operations. For light columns, we count private pairs to force sufficiently many rows. If a heavy column exists, we delete that column and all rows it occupies, and then recurse on the remaining matrix. The state variable $z = (n - t)/d^2$ incorporates both cases into a single induction scheme. The functions g and a represent the target lower bound and the light–heavy threshold, respectively. A parametric ordinary differential equation provides an explicit construction of these two functions.

The numerical computation serves one precise purpose: it rigorously verifies that the ODE solution extends from its initial value to the first contact point, where $2y = u$. The outward-rounding interval program in the appendix encloses the true value of the integral at every step. Thus, the final sign test does not rely on floating-point plots. The three figures help visualize the construction but do not constitute part of the proof. In contrast with the constant threshold used by Shangguan and Ge [12], our threshold changes with the recursive state. This improvement raises the asymptotic quadratic coefficient from $(15 + \sqrt{33})/24$ to 0.9283.

References

[1] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Stat.*, vol. 14, no. 4, pp. 436–440, 1943. [Online]. Available: https://doi.org/10.1214/aoms/1177731363

[2] W. H. Kautz and R. C. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. Inf. Theory*, vol. 10, no. 4, pp. 363–377, Oct. 1964. [Online]. Available: https://doi.org/10.1109/TIT.1964.1053689

[3] A. G. D'yachkov and V. V. Rykov, "Bounds on the length of disjunctive codes," *Problemy Peredachi Informatsii*, vol. 18, no. 3, pp. 7–13, 1982.

[4] A. G. D'yachkov and V. V. Rykov, "A survey of superimposed code theory," *Probl. Control Inf. Theory*, vol. 12, no. 4, pp. 229–242, 1983.

[5] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*, 2nd ed., Series on Applied Mathematics, vol. 12. River Edge, NJ, USA: World Scientific, 2000.

[6] J. K. Wolf, "Born again group testing: Multiaccess communications," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 185–191, Mar. 1985. [Online]. Available: https://doi.org/10.1109/TIT.1985.1057007

[7] C. C. Chen and F. K. Hwang, "Detecting and locating electrical shorts using group testing," *IEEE Trans. Circuits Syst.*, vol. 36, no. 8, pp. 1113–1116, Aug. 1989. [Online]. Available: https://doi.org/10.1109/31.192423

[8] A. J. Macula and L. J. Popyack, "A group testing method for finding patterns in data," *Discrete Appl. Math.*, vol. 144, nos. 1–2, pp. 149–157, 2004.

[9] Y. Xuan, I. Shin, M. T. Thai and T. Znati, "Detecting application denial-of-service attacks: A group-testing-based approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 8, pp. 1203–1216, Aug. 2010. [Online]. Available: https://doi.org/10.1109/TPDS.2009.147

[10] A. Cohen, A. Cohen and O. Gurewitz, "Secure group testing," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 4003–4018, 2021. [Online]. Available: https://doi.org/10.1109/TIFS.2021.3098077

[11] P. Erdős, P. Frankl and Z. Füredi, "Families of finite sets in which no set is covered by the union of r others," *Israel J. Math.*, vol. 51, no. 1, pp. 79–89, 1985. [Online]. Available: https://doi.org/10.1007/BF02772959

[12] C. Shangguan and G. Ge, "New bounds on the number of tests for disjunct matrices," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7518–7521, 2016. [Online]. Available: https://doi.org/10.1109/TIT.2016.2614726

[13] P. Erdős and T. Gallai, "On maximal paths and circuits of graphs," *Acta Math. Acad. Sci. Hungar*, vol. 10, pp. 337–356, Sept. 1959. [Online]. Available: https://doi.org/10.1007/BF02024498

[14] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*. New York, NY, USA: McGraw-Hill, 1955.

[15] IEEE, "IEEE Standard for Floating-Point Arithmetic," IEEE Std 754-2019, 2019. [Online]. Available: https://doi.org/10.1109/IEEESTD.2019.8766229

APPENDIX

PYTHON IMPLEMENTATION OF THE INTERVAL CERTIFICATE

This appendix gives the executable certificate used in Proposition 2. It verifies three facts. The solution meets no singularity on the integration interval. Its exact value at every grid endpoint lies in the computed interval. Finally, $2y - u$ is positive at the initial point and negative at $u = 1.992$.

The program first checks that Python's `float` uses the IEEE 754 binary64 format with 53 bits of binary precision [15]. The `Fraction` class stores the rational input exactly. After each floating-point endpoint operation, `nextafter` expands both endpoints outward by one ulp. Hence, addition, subtraction, multiplication, division, and squaring in the `Interval` class always produce an interval that contains the exact result.

The program first encloses the unique root u_0 of $\pi_2(u_0) = c_0$ by exact rational inequalities. It then connects u_0 to the first 10^{-6} grid point on its right and proceeds to 1.992 with step size 10^{-6}. If $y(u^{(j)}) \in Y_j$, the program sets

$$B_j := Y_j + [-h_j, h_j], \qquad R_j := [u^{(j)}, u^{(j+1)}] \times B_j,$$

and computes $F(R_j) \subseteq I_j \subset (-1, 1)$. The first-exit argument in the main text ensures that the exact solution cannot leave B_j. Therefore, the exact integral gives the recurrence

$$y(u^{(j+1)}) \in Y_j + h_j I_j =: Y_{j+1}.$$

This is different from the usual Euler approximation, which uses only the slope at the left endpoint. Here no local truncation term is discarded.

The function `verify` checks the floating-point format, the rational enclosure of the initial root, the domain and denominator signs, $I_j \subset (-1, 1)$, the endpoint sign, and the bound $y < 1$ along the whole orbit. Any failed assertion terminates the program. It prints `certificate: PASSED` only if every check succeeds. The key numerical intervals appear in the proof of Proposition 2. The three plots are not used by the certificate.

```
"""Outward-rounded interval certificate for the 0.9283
    orbit."""

from __future__ import annotations

from dataclasses import dataclass
from fractions import Fraction
from math import inf, nextafter
import sys


def down(x: float) -> float:
    return x if x == -inf else nextafter(x, -inf)


def up(x: float) -> float:
    return x if x == inf else nextafter(x, inf)


@dataclass(frozen=True)
class Interval:
    lo: float
    hi: float

    def __post_init__(self) -> None:
        assert self.lo <= self.hi

    def __add__(self, other: "Interval") -> "Interval":
        return Interval(down(self.lo + other.lo),
                        up(self.hi + other.hi))

    def __neg__(self) -> "Interval":
        return Interval(-self.hi, -self.lo)

    def __sub__(self, other: "Interval") -> "Interval":
        return self + (-other)

    def __mul__(self, other: "Interval") -> "Interval":
        values = (
            self.lo * other.lo,
            self.lo * other.hi,
            self.hi * other.lo,
            self.hi * other.hi,
        )
        return Interval(down(min(values)), up(max(values)
    ))

    def reciprocal(self) -> "Interval":
        assert not (self.lo <= 0.0 <= self.hi)
        return Interval(down(1.0 / self.hi), up(1.0 /
     self.lo))

    def __truediv__(self, other: "Interval") -> "Interval
    ":
        return self * other.reciprocal()

    def square(self) -> "Interval":
        if self.lo >= 0.0:
            return Interval(down(self.lo * self.lo),
                            up(self.hi * self.hi))
        if self.hi <= 0.0:
            return Interval(down(self.hi * self.hi),
                            up(self.lo * self.lo))
        return Interval(0.0,
                        up(max(self.lo * self.lo, self.hi
      * self.hi)))


def rational(numerator: int, denominator: int = 1) ->
     Interval:
    exact = Fraction(numerator, denominator)
    rounded = float(exact)
    return Interval(down(rounded), up(rounded))


ZERO = rational(0)
ONE = rational(1)
TWO = rational(2)
THREE = rational(3)
FOUR = rational(4)
EIGHT = rational(8)
FIVE_THIRDS = rational(5, 3)


def pi2_interval(u: Interval) -> Interval:
    return -THREE * u.square() + EIGHT * u - FOUR


def rhs(u: Interval, y: Interval) -> Interval:
    assert u.lo > FIVE_THIRDS.hi and u.hi < TWO.lo
    assert y.lo > ZERO.hi and y.hi < ONE.lo
    pi2_value = pi2_interval(u)
    separation = y - u
```

```
    assert pi2_value.lo > 0.0
    assert separation.hi < 0.0
    return (y * (FOUR - THREE * u) * (TWO * y - u)
            / (pi2_value * separation))


def first_grid_value(
    c0: Interval,
    u0: Interval,
    first_numerator: int,
    scale: int,
) -> tuple[Interval, float]:
    grid_point = rational(first_numerator, scale)
    step = grid_point - u0
    assert step.lo > 0.0
    radius = step.hi
    tube = Interval(down(c0.lo - radius),
                    up(c0.hi + radius))
    slope = rhs(Interval(u0.lo, grid_point.hi), tube)
    assert slope.lo > -1.0 and slope.hi < 1.0
    return c0 + step * slope, tube.hi


def verify() -> None:
    assert sys.float_info.radix == 2
    assert sys.float_info.mant_dig == 53
    assert sys.float_info.max_exp == 1024

    c0_numerator, c0_denominator = 9283, 10000
    c0_exact = Fraction(c0_numerator, c0_denominator)
    c0 = rational(c0_numerator, c0_denominator)

    # Rational endpoints enclosing the unique root pi2(u) =c0.
    u0_lower = Fraction(170077191474675, 100000000000000)
    u0_upper = Fraction(170077191474677, 100000000000000)
    pi2_exact = lambda x: -3 * x * x + 8 * x - 4
    assert pi2_exact(u0_lower) > c0_exact > pi2_exact(u0_upper)
    u0 = Interval(down(float(u0_lower)), up(float(u0_upper)))

    initial_gap = TWO * c0 - u0
    assert initial_gap.lo > 0.0

    scale = 1_000_000
    target_numerator = 1_992_000
    first_numerator = int(u0.hi * scale) + 1
    value, global_upper = first_grid_value(
        c0, u0, first_numerator, scale
    )
    max_width = value.hi - value.lo

    for numerator in range(first_numerator, target_numerator):
        left = rational(numerator, scale)
        right = rational(numerator + 1, scale)
        step = rational(1, scale)

        # The bound |y'|<1 proves that the exact solution stays
        # in value +/- step throughout this grid interval.
        tube = Interval(down(value.lo - step.hi),
                        up(value.hi + step.hi))
        slope = rhs(Interval(left.lo, right.hi), tube)
        assert slope.lo > -1.0 and slope.hi < 1.0

        # y(right)=y(left)+integral y'(u) du.
        value = value + step * slope
        global_upper = max(global_upper, tube.hi, value.hi)
        max_width = max(max_width, value.hi - value.lo)

    target = rational(target_numerator, scale)
    final_gap = TWO * value - target

    assert final_gap.hi < 0.0
    assert global_upper < 1.0

    print("certificate: PASSED")
    print(f"u0 in [{u0.lo:.17g}, {u0.hi:.17g}]")
    print(
        "2*c0-u0 in "
        f"[{initial_gap.lo:.17g}, {initial_gap.hi:.17g}]"
    )
    print(f"y(1.992) in [{value.lo:.17g}, {value.hi:.17g}]")
    print(
        "2*y(1.992)-1.992 in "
        f"[{final_gap.lo:.17g}, {final_gap.hi:.17g}]"
    )
    print(f"global certified upper bound for y: {global_upper:.17g}")
    print(f"maximum endpoint enclosure width: {max_width:.3e}")


if __name__ == "__main__":
    verify()
```