COMPUTER-ASSISTED PROOF OF KERNEL INEQUALITIES

HENRY COHN, ABHINAV KUMAR, STEPHEN D. MILLER,
DANYLO RADCHENKO, AND MARYNA VIAZOVSKA

1. Introduction

This article is a supplementary document for the paper [1], which proves universal optimality of the E_8 and Leech lattices as well as new interpolation formulas. Most of the proof does not require computer assistance, with one major exception: the last step in the proof of universal optimality, described in Section 6 of [1], requires certain inequalities for which we have no conceptual proof. Instead, we verify them by computer calculations using interval arithmetic. In this article, we expand on Section 6 and provide the details and code needed to verify the inequalities. Our code uses the SageMath 9.5 open-source computer algebra system [3].

In Section 2, we summarize some formulas from [1]. In Section 3, we examine series expansions of elliptic integrals and use them to prove equation (6.5) from [1]. In Section 4, we prove bounds on the error introduced by truncating power series. Finally, in Section 5 we explain how our SageMath calculations implement the strategy outlined in Sections 6.5 through 6.7 of [1] for proving the kernel inequalities (Proposition 6.1 in [1]).

2. Elliptic integral formulas

For convenience, we collect here some key formulas from Section 2.2 of [1]. Recall our normalizations

$$K(m) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - m\sin^2\theta}} \quad \text{and} \quad E(m) = \int_0^{\pi/2} \sqrt{1 - m\sin^2\theta}\, d\theta.$$

Note that many references, such as [2, Chapter 19], define K and E in terms of the elliptic modulus k, so that the complete elliptic integrals are what we call $k \mapsto K(k^2)$ and $k \mapsto E(k^2)$. We also use K' to denote the derivative of the function K, rather than an elliptic integral with respect to the complementary modulus. Then these functions satisfy the identities

$$\begin{gathered}
K'(m) = \frac{E(m)}{2m(1-m)} - \frac{K(m)}{2m},\\
K(m)E(1-m) + E(m)K(1-m) - K(m)K(1-m) = \frac{\pi}{2},\\
K(m)K'(1-m) + K'(m)K(1-m) = \frac{\pi}{4m(1-m)},\\
\Theta_3(z)^2 = 2\pi^{-1}K(\lambda(z)),\\
U(z) = 4\pi^{-2}K(\lambda(z))^2,\\
V(z) = 4\pi^{-2}\lambda(z)K(\lambda(z))^2,\\
W(z) = 4\pi^{-2}\lambda_S(z)K(\lambda(z))^2,\\
\frac{K(1-\lambda(z))}{K(\lambda(z))} = -iz,\\
\lambda'(z) = 4i\pi^{-1}\lambda(z)(1-\lambda(z))K(\lambda(z))^2,
\end{gathered}$$

This article and the code described in it are available from DSpace@MIT at `https://hdl.handle.net/1721.1/141226`.

and

$$E_2(z) = 4\pi^{-2} K(\lambda(z))\Big(3E(\lambda(z)) - (2-\lambda(z))K(\lambda(z))\Big)$$

for $|m| < 1$ and $z \in \mathbb{H}$ with z on the imaginary axis. See Section 2.2 of [1] for more details.

3. Asymptotics

In this section we use series expansions of elliptic integrals to prove equation (6.5) from [1]. For nonnegative integers n and m, define the Pochhammer symbol

$$(\alpha)_n = \alpha(\alpha+1)\dots(\alpha+n-1)$$

(for $n = 0$ we have the empty product 1), and define d by

$$d(m) = \psi(m+1) - \psi(m+1/2),$$

where $\psi(m)$ is the digamma function $\psi(z) = \Gamma'(z)/\Gamma(z)$. In particular, we have the special values $\psi(1) = -\gamma$ and $\psi(1/2) = -2\log 2 - \gamma$. Using the properties of ψ, we also obtain the recurrence

$$d(m+1) = d(m) - \frac{2}{(2m+1)(2m+2)}$$

for $m = 0, 1, \dots$. Therefore $d(0) = 2\log 2$ and $d(m)$ tends monotonically to 0 from above as $m \to \infty$.

The elliptic integrals E and K are holomorphic in the open unit disk. Their behavior near 1 is governed by

$$K(1-z) = \sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}\right)_m \left(\frac{1}{2}\right)_m}{m!\, m!} z^m \big(-\log(z)/2 + d(m)\big)$$

and

$$E(1-z) = 1 + \frac{1}{2}\sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}\right)_m \left(\frac{3}{2}\right)_m}{(2)_m\, m!} z^{m+1} \left(-\frac{1}{2}\log(z) + d(m) - \frac{1}{(2m+1)(2m+2)}\right).$$

(see §19.12 in [2]). Because

$$\left(\frac{1}{2}\right)_m = \frac{(2m)!}{2^{2m} m!}, \quad \left(\frac{3}{2}\right)_m = \frac{(2m+1)!}{2^{2m} m!}, \quad \text{and} \quad (2)_m = (m+1)!,$$

we can write

$$K(1-z) = A_1(z) + A_2(z)\log(z) \quad \text{and} \quad E(1-z) = A_3(z) + A_4(z)\log(z),$$

with

$$\begin{aligned}
A_1(z) &= \sum_{m=0}^{\infty} \frac{d(m)}{2^{4m}} \binom{2m}{m}^2 z^m,\\
A_2(z) &= -\sum_{m=0}^{\infty} \frac{1}{2^{4m+1}} \binom{2m}{m}^2 z^m,\\
A_3(z) &= 1 + \sum_{m=0}^{\infty} \frac{1}{2^{4m+1}} \binom{2m}{m}\binom{2m+1}{m} z^{m+1} \left(d(m) - \frac{1}{(2m+1)(2m+2)}\right), \quad \text{and}\\
A_4(z) &= -\sum_{m=0}^{\infty} \frac{1}{2^{4m+2}} \binom{2m}{m}\binom{2m+1}{m} z^{m+1}.
\end{aligned}$$

From these expressions and the inequalities

$$\binom{2m}{m} \le \frac{2^{2m}}{\sqrt{2m+1}} \quad \text{and} \quad \binom{2m+1}{m} < \frac{2^{2m+1}}{\sqrt{2m+2}},$$

we can bound the coefficient $a_i(m)$ of z^m in each of these power series $A_i(z)$, obtaining

$$\begin{aligned}
0 &\le a_1(m) \le (2m+1)^{-1} d(m) \le (3/2)/(m+1),\\
0 &\le -a_2(m) \le (2m+1)^{-1}/2 \le (1/2)/(m+1),\\
0 &\le a_3(m) \le (2m+1)^{-1/2}(2m+2)^{-1/2} d(m) \le 1/(m+1), \quad \text{and}\\
0 &\le -a_4(m) \le (2m+1)^{-1/2}(2m+2)^{-1/2}/2 \le (1/4)/m \quad \text{for } m \ge 1.
\end{aligned}$$

Here we used $d(m) \le 2\log 2 \approx 1.386294$, as well as the recurrence relation for $d(m)$. Hence each A_j is a holomorphic function on the open unit disk with real Taylor coefficients about the origin. Furthermore, A_1 and A_3 have nonnegative coefficients, while A_2 and A_4 have nonpositive coefficients.

We will also need control of the Taylor expansions of $E(z)$ and $K(z)$ about 0, namely

$$\begin{aligned}
K(z) &= \frac{\pi}{2}\sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}\right)_m \left(\frac{1}{2}\right)_m}{m!\, m!} z^m = \frac{\pi}{2}\sum_{m=0}^{\infty} \frac{1}{2^{4m}} \binom{2m}{m}^2 z^m \quad \text{and}\\
E(z) &= \frac{\pi}{2}\sum_{m=0}^{\infty} \frac{\left(-\frac{1}{2}\right)_m \left(\frac{1}{2}\right)_m}{m!\, m!} z^m\\
&= \frac{\pi}{2}\left(1 - \sum_{m=0}^{\infty} \frac{1}{2^{4m+2}(m+1)} \binom{2m+1}{m}\binom{2m}{m} z^{m+1}\right)
\end{aligned}$$

(see §19.5 in [2]). Again $K(z)$ and $E(z)$ converge on the unit ball. Furthermore,

$$K(z) = -\pi A_2(z) \qquad \text{and} \qquad E(z) - K(z) = \pi A_4(z).$$

Next, we verify equation (6.5) from [1]. Specifically, we show that $z \mapsto z^2 e^{2\pi K(1-z)/K(z)}$ is holomorphic on the unit disk and establish bounds on the real interval $(0, 1/2)$. Since the function $\Delta = (UVW)^2/256$ is nonvanishing on the upper half plane (and each of U, V, W is holomorphic), so is U. We can now use the identity $U(z) = 4\pi^{-2} K(\lambda(z))^2$ to conclude that K does not vanish in the open unit disk. (Note that we had stated this identity only for z on the real axis, but we can analytically continue it to the region λ maps to the unit disk, namely the ideal hyperbolic triangle with vertices 0, 2, and $i\infty$.) Thus, dividing by $K(z)$ is not a problem when $|z| < 1$.

Now the identity $K(1-z) = A_1(z) - (1/\pi)K(z)\log z$ for $|z| < 1$ shows that

$$\frac{K(1-z)}{K(z)} = \frac{A_1(z)}{K(z)} - \frac{1}{\pi}\log z,$$

with $A_1(z)/K(z)$ being holomorphic and of course $\log z$ being multivalued, and it follows that

$$z^2 e^{2\pi K(1-z)/K(z)} = e^{2\pi A_1(z)/K(z)}$$

is holomorphic for $|z| < 1$.

Finally, to bound $y^2 e^{2\pi K(1-y)/K(y)}$ for $0 < y < 1/2$, it is convenient to remove factors of π from the notation by setting

$$\Phi(y) = \sum_{m=0}^{\infty} \frac{1}{2^{4m}} \binom{2m}{m}^2 y^m \quad \text{and} \quad \Psi(y) = \sum_{m=1}^{\infty} \frac{e(m)}{2^{4m}} \binom{2m}{m}^2 y^m,$$

with $e(m) = d(0) - d(m)$ being a positive increasing sequence with limit $d(0) = 2\log 2$ as $m \to \infty$. Then

$$K(1-y) = -\frac{1}{2}\log(y)\Phi(y) + d(0)\Phi(y) - \Psi(y) \quad \text{and} \quad K(y) = \frac{\pi}{2}\Phi(y).$$

Because

$$2\pi\frac{K(1-y)}{K(y)} = -2\log(y) + 8\log(2) - \frac{4\Psi(y)}{\Phi(y)},$$

we wish to bound $\Phi(y)$ and $\Psi(y)$ for $0 < y < 1/2$. A straightforward upper bound shows that

$$\Phi(y) \leq 1 + \frac{1}{4}y + \frac{9}{64}y^2 + \frac{25}{256}y^3 + \frac{1225}{8192}y^4,$$

while

$$\Psi(y) \geq \frac{e(1)}{4}y + \frac{9e(2)}{64}y^2 + \frac{25e(3)}{256}y^3 + \frac{1225e(4)}{16384}y^4$$

because it has positive coefficients. On the other hand, we have the crude bound

$$\begin{aligned}\Psi(y) &\leq \sum_{m=1}^{\infty} \frac{d(0)}{2^{4m}}\binom{2m}{m}^2 y^m \\ &\leq d(0)\sum_{m=1}^{\infty} \frac{1}{2m+1}y^m \leq \frac{d(0)}{3}\sum_{m=1}^{\infty}\left(\frac{1}{2}\right)^m = d(0)/3 < 1/2,\end{aligned}$$

as well as $\Phi(y) \geq 1$. It follows that

$$-2 \leq -\frac{4\Psi(y)}{\Phi(y)} \leq -y - \frac{13}{32}y^2 - \frac{23}{96}y^3 - \frac{2701}{16384}y^4 + \frac{2315}{8192}y^5 =: \beta < 0.$$

Exponentiating, we get

$$\exp\left(2\pi\frac{K(1-y)}{K(y)}\right) = \frac{256}{y^2}\exp\left(-\frac{4\Psi(y)}{\Phi(y)}\right) < \frac{256}{y^2}\exp(\beta).$$

Now, note that

$$\exp(\beta) \leq 1 + \beta + \frac{\beta^2}{2} + \frac{\beta^3}{6} + \frac{\beta^4}{24},$$

since the remainder of the series is

$$\frac{\beta^5}{120}\left(1 + \frac{\beta}{6} + \frac{\beta^2}{42} + \dots\right)$$

is a negative number times the series in parenthesis, which converges to a positive number because $\beta \geq -2$. Using this upper bound for $\exp(\beta)$, we can check that

$$\exp(\beta) \leq 1 - y + \frac{3}{32}y^2 + \frac{2}{15}y^4.$$

Finally, we conclude that

$$\exp\left(2\pi\frac{K(1-y)}{K(y)}\right) \leq \frac{256}{y^2} - \frac{256}{y} + 24 + \frac{512}{15}y^2,$$

as desired.

4. Series truncation

Here, we collect some useful lemmas for approximations of power series by their truncated (polynomial) versions, which will be applied in the proof of the kernel inequalities.

Lemma 4.1. *Suppose $f^1(x), \dots, f^s(x)$ are power series such that the n-th coefficient of f^i is bounded by $C_{f^i}/(n+1)$ in absolute value. Then the n-th coefficient f_n of $f = \prod f^i$ satisfies*

$$|f_n| \leq M(s)\prod C_{f^i},$$

where $M(1) = 1$ and $M(s) = 2^{s-1}(s-1)^{s-1}/e^{s-2}$ for $s \geq 2$.

Proof. We can assume without loss of generality that $C_{f^i} = 1$ for all i. First we show by induction on s that

$$|f_n| \leq \frac{2^{s-1}(1+\log(n+1))^{s-1}}{n+1}.$$

This is clear for $s = 1$, and the induction step follows from

$$\begin{aligned}\sum_{0\leq i\leq n} \frac{(1+\log(i+1))^{s-1}}{(i+1)(n-i+1)} &\leq (1+\log(n+1))^{s-1} \sum_{0\leq i\leq n} \frac{1}{n+2}\left(\frac{1}{i+1}+\frac{1}{n-i+1}\right)\\ &\leq \frac{2(1+\log(n+1))^{s-1}}{n+1}\sum_{0\leq i\leq n}\frac{1}{i+1}\\ &\leq \frac{2(1+\log(n+1))^{s-1}}{n+1}(1+\log(n+1)).\end{aligned}$$

It now suffices to show that $(1+\log(n+1))^{s-1}/(n+1) \leq (s-1)^{s-1}/e^{s-2}$, which follows from computing the logarithmic derivative of $u \mapsto (1+\log u)^{s-1}/u$. □

Corollary 4.2. *Let $f(x) = A_1(x)^{j_1}A_2(x)^{j_2}A_3(x)^{j_3}A_4(x)^{j_4}\big(\log(1-x)\big)^{j_5}$. Then each coefficient of $f(x)$ is bounded in absolute value by $M(j_1+j_2+j_3+j_4+j_5)(3/2)^{j_1}(1/2)^{j_2}(1/4)^{j_4}$.*

Proof. Apply the lemma to j_1 copies of $A_1(x)$, j_2 copies of $A_2(x)$, j_3 copies of $A_3(x)$, j_4 copies of $A_4(x)/x$, and j_5 copies of $\log(1-x)/x$. □

Lemma 4.3. *Let $f(x,y) = \sum a_{i,j}x^iy^j$ be a power series satisfying $|a_{i,j}| \leq B$, and let $|\alpha|, |\beta| < 1$. Then the power series $g(x,y) = f(x+\alpha, y+\beta) = \sum b_{i,j}x^iy^j$ satisfies*

$$|b_{i,j}| \leq B(1-|\alpha|)^{-(i+1)}(1-|\beta|)^{-(j+1)}.$$

Proof. We have

$$b_{i,j} = \sum_{n\geq i, m\geq j} a_{nm}\alpha^{n-i}\beta^{m-j}\binom{n}{i}\binom{m}{j},$$

and therefore

$$|b_{i,j}| \leq B\sum_{k\geq 0,\ell\geq 0} |\alpha|^k|\beta|^l\binom{i+k}{i}\binom{j+\ell}{j}$$

Now the result follows from splitting the sum into a product and using the identity

$$\sum_{k\geq 0}|\alpha|^k\binom{i+k}{i} = (1-|\alpha|)^{-(k+1)}.$$ □

Definition 4.4. The *truncation* of a power series $f(x) = \sum a_nx^n$ at exponent or degree N will be denoted

$$f_N(x) = \sum_{i\leq N} a_nx^n.$$

Similarly, for a two-variable power series $f(x,y) = \sum a_{i,j}x^iy^j$ we define the truncation at degree N to be

$$f_N(x,y) = \sum_{i\leq N, j\leq N} a_{i,j}x^iy^j.$$

Corollary 4.5. *Let $f(x,y) = \sum a_{i,j}x^iy^j$ be a power series satisfying $|a_{i,j}| \leq B$. Then the error from truncating $f(1/2+x, 1/2+y)$ at exponent N for x and y in the domain $|x| < \gamma$ and $|y| < \delta$ is at most*

$$\frac{4B}{(1-2\gamma)(1-2\delta)}\Big((2\gamma)^{N+1}+(2\delta)^{N+1}-(4\gamma\delta)^{N+1}\Big).$$

In other words, if we set $g(x, y) = f(1/2 + x, 1/2 + y)$, then

$$|g(x, y) - g_N(x, y)| \le \frac{4B}{(1 - 2\gamma)(1 - 2\delta)}\Big((2\gamma)^{N+1} + (2\delta)^{N+1} - (4\gamma\delta)^{N+1}\Big)$$

whenever $|x| < \gamma$ and $|y| < \delta$.

Proof. Let $g(x, y) = \sum_{i,j} b_{i,j} x^i y^j$. By Lemma 4.3, $|b_{i,j}| \le B2^{i+j+2}$, and so the conclusion follows from computing the sum

$$B \sum_{i \ge N+1 \text{ or } j \ge N+1} \gamma^i \delta^j 2^{i+j+2}.$$

To do so, we break up the sum via

$$\sum_{i \ge N+1 \text{ or } j \ge N+1} = \sum_{i \ge N+1,\ j \ge 0} + \sum_{i \ge 0,\ j \ge N+1} - \sum_{i \ge N+1,\ j \ge N+1}.$$ □

Remark 4.6. With $\gamma = \delta = 0.1$ and $N = 25$, we obtain an upper bound of $8.388608 \cdot 10^{-18}B$.

Lemma 4.7. *Let $f(x) = \sum_{i \ge 0} a_i x^i$ be a power series with $|a_i| \le B$ for all i, and let $f_N(x) = \sum_{0 \le i \le N} a_i x^i$ be its truncation at degree N. Let $g(z) = f(z + \alpha)$ for some $\alpha \in (0, 1)$, and let g_M be its truncation at degree M, with $M \le N$. If α satisfies.*

$$B \sum_{k \ge N-M+1} \binom{k+M}{M} \alpha^k < \varepsilon,$$

then each coefficient of $\big(f_N(x + \alpha)\big)_M - g_M(x)$ is less than ε in absolute value.

Proof. As before, the coefficients b_j of g are given by

$$b_j = \sum_{i \ge j} a_i \binom{i}{j} \alpha^{i-j}.$$

The error if we first cut off at $i = N$ is

$$\sum_{i \ge N+1} a_i \binom{i}{j} \alpha^{i-j},$$

which we can bound in absolute value by

$$B \sum_{k \ge N+1-j} \binom{k+j}{j} \alpha^k.$$

This bound is increasing in j: as j increases, the sum is over a larger range and the binomial coefficients increase. Thus, among all coefficients with $j \le M$, the bound is maximized when $j = M$. □

Corollary 4.8. *If f is one of A_1, A_2, A_3, or A_4, and $\alpha = 1/2$, $M = 25$, and $N = 185$, then the hypotheses of Lemma 4.7 hold with $\varepsilon = 10^{-17}$.*

Proof. We can take $B = 3/2$, and so the conclusion follows from

$$(3/2) \cdot \sum_{k=161}^{\infty} (1/2)^k \binom{k+25}{25} < 10^{-17}.$$ □

Remark 4.9. If $f(x)$ is $\log(x)$ or $\log(1 - x)$, then we can exactly compute the coefficients of $f(x + 1/2)$, with no need for the approximation lemma above.

Lemma 4.10. *Let $f(x,y) = \sum_{i,j\ge 0} a_{i,j}x^iy^j$ be a power series with $|a_{i,j}| < B$ for all i and j. Furthermore, suppose that $f(x,y)$ is divisible by x^{m_x} and y^{m_y}, and let $f_N(x,y)$ be its truncation at degree N in each of x and y. Then for $|x| < \delta_x$, $|y| < \delta_y$, and $\max(m_x, m_y, M) < N$,*

$$\begin{aligned} |f(x,y) - f_N(x,y)| &\le B\frac{|x|^{m_x}|y|^{N+1} + |y|^{m_y}|x|^{N+1} - |x|^{N+1}|y|^{N+1}}{(1-|x|)(1-|y|)} \\ &\le B\frac{|y|^M\delta_x^{m_x}\delta_y^{N+1-M} + |x|^M\delta_y^{m_y}\delta_x^{N+1-M} - |y|^M\delta_x^{N+1}\delta_y^{N+1-M}}{(1-\delta_x)(1-\delta_y)}. \end{aligned}$$

The only role of M in this lemma is to moderate the dependence on $|x|$ and $|y|$ in the final bound. It allows some dependence, to obtain improved estimates when $|x|$ and $|y|$ are small, but less dependence than in the previous bound.

Proof. We may assume without loss of generality that $B = 1$. Now the first inequality follows from summing

$$\sum_{\substack{i\ge m_x,\ j\ge m_y \\ i\ge N+1 \text{ or } j\ge N+1}} |x|^i|y|^j = \sum_{i\ge m_x,\ j\ge N+1} |x|^i|y|^j + \sum_{i\ge N+1,\ j\ge m_y} |x|^i|y|^j - \sum_{i\ge N+1,\ j\ge N+1} |x|^i|y|^j$$

and the ensuing algebra. For the second inequality, we note that $|x|^r/(1-|x|)$ is monotonically increasing in $|x|$ for $r \ge 0$ by the geometric series formula, and so is $(|x|^r - |x|^s)/(1-|x|)$ for integers $s > r \ge 0$ for the same reason. □

Lemma 4.11. *Let $f(x,y) = \sum_{i,j\ge 0} a_{i,j}x^iy^j$ be a power series with $|a_{i,j}| < B$ for all i and j, such that $f(x,x) = 0$. Furthermore, suppose that $f(x,y)$ is divisible by x^{m_x} and y^{m_y}. Let $g(x,y) = f(x,y)/(x-y)$, and let $g_N(x,y)$ be its truncation at degree N in each of x and y. Then for $|x| < \delta_x$, $|y| < \delta_y$, and $\max(m_x, m_y, M) < N$,*

$$\begin{aligned} |g(x,y) - g_N(x,y)| &< \frac{|x|^{m_x}\big(1+(1-|x|)m_x\big) - |x|^{N+1}\big(1+(1-|x|)(N+1)\big)}{(1-|x|)^2}\cdot\frac{|y|^{N+1}}{1-|y|} \\ &\quad + \frac{|y|^{m_y}\big(1+(1-|y|)m_y\big) - |y|^{N+1}\big(1+(1-|y|)(N+1)\big)}{(1-|y|)^2}\cdot\frac{|x|^{N+1}}{1-|x|} \\ &\quad + \frac{(|x||y|)^{N+1}\big(1+(1-|x||y|)(N+1)\big)}{(1-|x|)(1-|y|)(1-|x||y|)} \\ &< \frac{\delta_x^{m_x}\big(1+(1-\delta_x)m_x\big) - \delta_x^{N+1}\big(1+(1-\delta_x)(N+1)\big)}{(1-\delta_x)^2}\cdot\frac{|y|^M\delta_y^{N+1-M}}{1-\delta_y} \\ &\quad + \frac{\delta_y^{m_y}\big(1+(1-\delta_x)m_y\big) - \delta_y^{N+1}\big(1+(1-\delta_y)(N+1)\big)}{(1-\delta_y)^2}\cdot\frac{|x|^M\delta_x^{N+1-M}}{1-\delta_x} \\ &\quad + \frac{|x|^M\delta_x^{N+1-M}\delta_y^{N+1}\big(1+(1-\delta_x\delta_y)(N+1)\big)}{(1-\delta_x)(1-\delta_y)(1-\delta_x\delta_y)}. \end{aligned}$$

Proof. We write

$$\begin{aligned} g(x,y) = \frac{f(x,y)}{x-y} = \frac{f(x,y)-f(y,y)}{x-y} &= \sum_{n,m\ge 0} a_{n,m}\frac{x^n-y^n}{x-y}y^m \\ &= \sum_{n,m\ge 0} a_{n,m}(x^{n-1} + x^{n-2}y + \cdots + y^{n-1})y^m \\ &= \sum_{k,\ell\ge 0}\sum_{i=0}^{\ell} a_{k+i+1,\ell-i}x^ky^\ell. \end{aligned}$$

In this formula, there are at most $\ell+1$ summands of the form $a_{n,m}$ in the expression for the coefficient $b_{k,\ell}$ of $x^k y^\ell$ in g. Similarly, because $g(x,y) = (f(x,y) - f(x,x))/(x-y)$, we see that $b_{k,\ell}$ also has an expression with $k+1$ summands. Therefore, $b_{k,\ell} \le B(1+\min(k,\ell))$. Suppose without loss of generality that $B=1$. We can therefore bound $g - g_N$ by

$$\begin{aligned}
|g(x,y) - g_N(x,y)| &\le \sum_{m=m_x}^{N} \sum_{n=N+1}^{\infty} |x|^m |y|^n (m+1) + \sum_{n=m_y}^{N} \sum_{m=N+1}^{\infty} |x|^m |y|^n (n+1) \\
&\quad + \sum_{m,n=N+1}^{\infty} |x|^m |y|^n (\min(m,n)+1) \\
&= \sum_{m=m_x}^{N} \sum_{n=N+1}^{\infty} |x|^m |y|^n (m+1) + \sum_{n=m_y}^{N} \sum_{m=N+1}^{\infty} |x|^m |y|^n (n+1) \\
&\quad + \sum_{m=N+1}^{\infty} \sum_{n=m}^{\infty} |x|^m |y|^n (m+1) + \sum_{n=N+1}^{\infty} \sum_{m=n}^{\infty} |x|^m |y|^n (n+1) \\
&\quad - \sum_{m=N+1}^{\infty} |x|^m |y|^m (m+1).
\end{aligned}$$

Now, note that

$$\sum_{m=m_x}^{N} z^m (m+1) = \frac{d}{dz} \sum_{m=m_x}^{N} z^{m+1} = \frac{d}{dz} \left(\frac{z^{m_x+1} - z^{N+2}}{1-z} \right)$$

and

$$\sum_{m=N+1}^{\infty} z^m (m+1) = \frac{d}{dz} \sum_{m=N+1}^{\infty} z^{m+1} = \frac{d}{dz} \left(\frac{z^{N+2}}{1-z} \right).$$

Using

$$\frac{d}{dz} \frac{z^m}{1-z} = \frac{z^{m-1}(1+(m-1)(1-z))}{(1-z)^2},$$

after a little bit of algebra we get the first bound in the result. For the second inequality, we use that all of the multiplicands in the three terms of the right side of the first inequality are positive and increasing in $|x|$ and $|y|$, due to the summation identities above. For instance, to deal with the last of the three terms, we note that

$$\frac{(|x||y|)^{N+1}\big(1+(1-|x||y|)(N+1)\big)}{(1-|x|)(1-|y|)(1-|x||y|)} = \frac{(|x||y|)^{N+1}}{(1-|x|)(1-|y|)(1-|x||y|)} + \frac{(|x||y|)^{N+1}(N+1)}{(1-|x|)(1-|y|)}$$

and use the geometric series to get a sum of positive terms, in each of which we can replace $|x|^k|y|^\ell$ by $|x|^M \delta_x^{k-M} \delta_y^\ell$ by monotonicity (note that the exponents k, ℓ for each of the summands are greater than M). □

5. Roadmap

We now have all the tools needed to carry out the strategy from Sections 6.5 through 6.7 of [1] and thereby prove the kernel inequalities (Proposition 6.1 in [1]). Recall from Section 6.5 in [1] that our goal is to check that a function of the form

$$K(x,y) = \frac{Q(x,y)}{(x-y)(1-x-y)}$$

is positive on the unit square $(0,1) \times (0,1)$, where $Q(x,y)$ is an explicitly given function, which vanishes when $x=y$ or $x+y=1$ as well as on the sides of the square. In principle, checking the

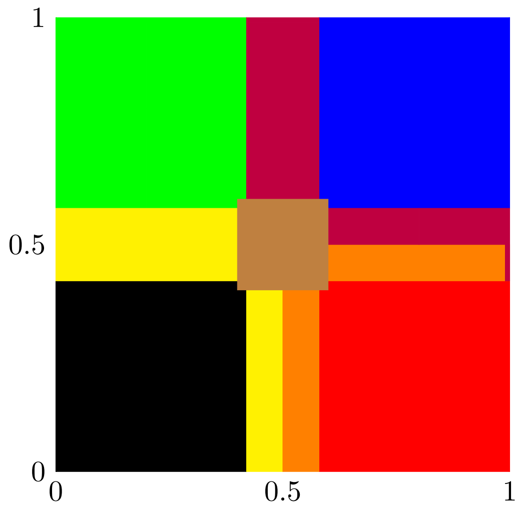

FIGURE 5.1. Regions used for the 8-dimensional kernel.

inequality by interval arithmetic [4] is straightforward away from the locations where Q vanishes. In practice, optimizing for efficient computation is an important consideration, and dealing with neighborhoods of the zero locus of Q requires additional analysis.

There are three cases, each with a different choice of $Q(x, y)$: the 8-dimensional kernel, the 24-dimensional kernel, and the truncated 24-dimensional kernel. In each case, $Q(x, y)$ is obtained as in [1], by setting $x = \lambda(\tau)$ and $y = \lambda(z)$, removing some obviously positive factors, and interchanging $x \leftrightarrow 1 - x$ and $y \leftrightarrow 1 - y$ in the non-truncated cases.[1]

To carry out these calculations and thereby verify positivity, we use different techniques on a patchwork of overlapping subregions of the square, shown in Figures 5.1 through 5.3 (when two regions overlap, the figures show only one of them). Our code keeps track of which regions have been analyzed and verifies that the final list completely covers the square or, in the truncated case, $(0, 1) \times (0, 0.49)$. We always show strict positivity, not just nonnegativity.

The regions we analyze are all made up of rectangles. They are classified into three types:

(1) Corner: This type of region covers a neighborhood of each of the corners of the square, as well as the diagonals away from the center of the square.
(2) Middle: This type covers a neighborhood of the center of the square.
(3) Nonsingular: This type covers the rest, away from the diagonals and the center. Note that the denominator of K does not vanish in these regions.

We now describe in slightly more detail how the verification of positivity is carried out in each of these regions. In this document we will specify the regions only for the 8-dimensional kernel, because the 24-dimensional case is considerably more elaborate (see Figure 5.2). The choice of regions for the remaining two kernels is specified in the code. The more complicated regions in those cases were chosen for the sake of efficiency, while the underlying techniques are the same as those described here and in [1]. All computations are performed using interval arithmetic [4], which obtains rigorous inequalities by keeping track of intervals that provably contain the true values being computed (when rounding is needed, it is always done so as to make the interval larger). Thus, instead of using floating point arithmetic to represent the field $\mathbb{R}$, we use floating point interval arithmetic to represent arithmetic with the set $\mathbb{IR}$ of closed subintervals of $\mathbb{R}$.

[1] This interchange is of course not necessary, but we found it convenient in our numerical exploration.

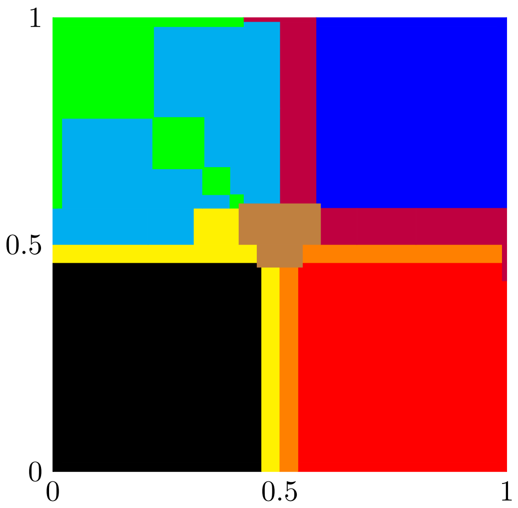

FIGURE 5.2. Regions used for the 24-dimensional kernel.

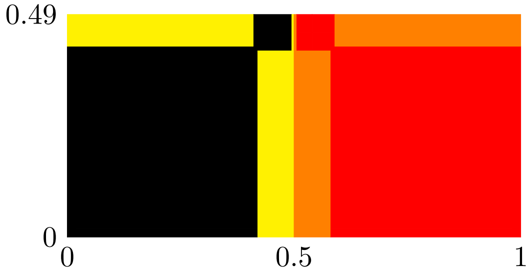

FIGURE 5.3. Regions used for the truncated 24-dimensional kernel.

5.1. **Corners.** For the 8-dimensional kernel function Q_8, the computation for each corner handles the following four regions in local coordinates centered at the corner:

- $C_1 = [0.001, 0.42] \times [0.001, 0.42]$
- $C_2 = [0, 0.01] \times [0.001, 0.42]$
- $C_3 = [0.001, 0.42] \times [0, 0.01]$
- $C_4 = [0, 0.00125] \times [0, 0.00125]$

(Technically, in the northwest corner C_1 is broken up further, for the sake of efficiency.) Specifically, we use the following transformed versions of the kernel for the four corners:

- $Q_8^{\mathrm{sw}}(x, y) = Q_8(x, y)$
- $Q_8^{\mathrm{se}}(x, y) = Q_8(1 - x, y)$
- $Q_8^{\mathrm{nw}}(x, y) = Q_8(x, 1 - y)$
- $Q_8^{\mathrm{ne}}(x, y) = Q_8(1 - x, 1 - y)$

In each case Q_8^*, we need to check that $Q_8^*/(x - y)$ is nonnegative (note that the denominator of $K(x, y)$ is invariant under $x \mapsto 1 - x$ and $y \mapsto 1 - y$, and in the corner region being considered, $1 - x - y$ is positive). Let Q denote one of these transformed functions Q_8^*. The first three rectangles are handled by the function `CP` (for "check positivity"), while the last uses a different strategy and is handled by the function `checktinycorner`.

The basic idea is as follows: we have to check the positivity of $g(x, y) = Q(x, y)/(x - y)$, where $Q(x, x) = 0$. To do so, we apply Lemma 4.11 to write the main term as $g_N(x, y)$ and bound the error $\varepsilon_N(x, y)$ as in that lemma, with $N = 20$. We combine the main and error terms into a polynomial

$h_N(x,y)$ in $\log(x), \log(y), x, y$ with coefficients in the set $\mathbb{IR}$ of closed real intervals, and then apply `CP`.

Our main function `CP` is passed in an interval $[a,b]$ for x and an interval $[c,d]$ for y, as well as the function h_N to evaluate. It computes the value of h_N using Horner evaluation, checking if it can directly be certified to be positive in that rectangle. If not, it breaks up the rectangle into 4 quarters (or 2 halves if one side is more than twice as long as the other), and recursively checks positivity on those. There are two additional subtleties.

1. As described in Section 6.7 of [1], in the northwest corner, even though $Q(x,x) = 0$, the coefficients of $Q(x,y)$ as a polynomial in $\log(x)$ and $\log(y)$ may not individually vanish on the diagonal. To remedy this, we subtract an appropriate polynomial $P(x)$ times $\log(x) - \log(y)$ to obtain $g(x,y)$ and its truncation g_N. Due to the positivity property of P discussed in Section 6.7 of [1] (and verified in the code via explicit factorizations), we do not have to worry about its remainder upon truncation. So the first subtlety is that when we evaluate g inside `CP`, we also supply P to help establish the positivity of the original kernel.
2. The second subtlety has to do with the appearance of $\log(y)^2$ terms in g or g_N. Fortuitously, the coefficient of any such term is a multiple of y^2, and we can fix the entire term (which could otherwise ruin the interval arithmetic near the x-axis $y = 0$) by absorbing one factor of y and using the inequality $y \log(y)^2 \leq 4/e^2$ for $0 < y \leq 1$.

5.2. **Middle.** Here, we check the positivity of the kernel in the region $[0.4, 0.6] \times [0.4, 0.6]$. The main idea is the following: letting $u = x - y$ and $v = 1 - x - y$, we can rewrite

$$K(x,y) = \frac{Q_8(x,y)}{(x-y)(1-x-y)} = \frac{Q_8\big(\frac{1}{2}(1+u-v), \frac{1}{2}(1-u-v)\big)}{uv}.$$

Because $Q_8(x,y) = 0$ when $x = y$ or $x + y = 1$, the function

$$\widetilde{Q}(u,v) := Q_8\big(\tfrac{1}{2}(1+u-v), \tfrac{1}{2}(1-u-v)\big)$$

vanishes identically along $u = 0$ and $v = 0$, and thus we can compute the Taylor expansion of $g(u,v) := \widetilde{Q}(u,v)/(uv)$. By the Lagrange remainder formula,

$$g(u,v) = g(0,0) + g^{(1,0)}(0,0)u + g^{(0,1)}(0,0)v + g^{(2,0)}(\mu,\nu)\frac{u^2}{2} + g^{(1,1)}(\mu,\nu)uv + g^{(0,2)}(\mu,\nu)\frac{v^2}{2}$$

for some $\mu \in (-|u|, |u|)$ and $\nu \in (-|v|, |v|)$. As explained in Section 6.5 of [1],

$$g^{(i,j)}(u,v) = \int_0^1 \int_0^1 s^i t^j \widetilde{Q}^{(i+1,j+1)}(us, vt)\, ds\, dt$$

Therefore, if we have bounds on the fourth partial derivatives of $\widetilde{Q}$ with respect to u and v, we obtain corresponding bounds on the second partial derivatives of g. For example, if $M_{3,1}$ is a constant such that $|\widetilde{Q}^{(3,1)}(u,v)| \leq M_{3,1}$ over the entire middle region, then

$$\left|g^{(2,0)}(u,v)\right| \leq \frac{M_{3,1}}{3}.$$

Using this notation and writing $g(0,0)$, $g^{(1,0)}(0,0)$, and $g^{(0,1)}(0,0)$ in terms of $\widetilde{Q}$, we obtain

$$g(u,v) \geq \widetilde{Q}^{(1,1)}(0,0) + \widetilde{Q}^{(2,1)}(0,0)u/2 + \widetilde{Q}^{(1,2)}(0,0)v/2 - M_{3,1}u^2/6 - M_{1,3}v^2/6 - M_{2,2}|uv|/4,$$

and we can check by interval arithmetic that the lower bound is strictly positive in the middle region.

5.3. **Nonsingular.** Here, we check positivity in a handful of remaining rectangles, which avoid the diagonals of the square. The method of checking positivity is similar to that in the corner regions, except even simpler, since there is no division by $x - y$ involved. Therefore, we use Lemma 4.10 for our error bound, and pass the truncation + error polynomial $h_N(x, y)$ of $h(x, y) = Q_8^*(x, y)$ to `CP`. The first subtlety (vanishing of terms along the diagonal) doesn't appear, and the second is dealt with as in the corners.

5.4. **Further truncation of power series.** Finally, we mention one important computational trick that is used to speed up the verification process significantly. Say we want to verify the positivity of a polynomial $f(x, y)$ with coefficients in $\mathbb{IR}$ in some rectangle $[a, b] \times [c, d]$. (That is, we have already truncated a power series, and absorbed the error term into appropriate terms of the polynomial.) Since the radii of convergence in the x- and y-directions are b and d, we may have to carry around a fairly high degree polynomial f.

Instead, we may choose points $e \in [a, b]$ and $f \in [c, d]$ (for instance, the midpoints of these intervals), and replace f by $g(x, y) = f(x + e, y + f)$, where we are now asking for the positivity of g in $[a - e, b - e] \times [c - f, d - f]$, and now the radius of convergence has shrunk to $\max(|a - e|, |b - e|)$ for x and $\max(|c - f|, |d - f|)$ for y. Therefore, we can now truncate g to a lower degree polynomial h, and convert all the higher degree terms into an error bound, which we can again absorb into h.

5.5. **Computer files.** We now provide some details of the different parts of the code, organized by the files.

- `setup.sage` : This file defines the main rings and functions used in the verification of positivity. In particular, we define the real intervals $\mathbb{IR}$ (called `R` in the code) with 53 bits of precision, S the polynomial ring in x, y, $\log(x)$, $\log(y)$, $\log(1 - x)$, $\log(1 - y)$, $A_i(x)$ and $A_i(y)$ for $1 \leq i \leq 4$, π ,and $\log(2)$ over the rationals. In the Sage code, we call these variables `x`, `y`, `Lx`, `Ly`, `Lpx`, `Lpy`, `A1x` through `A4y`, `piv`, and `l2`, respectively, and we use `Ex`, `Epx`, `Kx`, `Kpx`, etc. for $E(x)$, $E(1 - x)$, $K(x)$, $K(1 - x)$, etc. Its counterpart T over $\mathbb{IR}$ uses the variables `xR`, `yR`, etc. The power series for $\log(1 - x)$ and $A_i(x)$ are also defined up to degree 185 (chosen because of Corollary 4.8). Next, we list each function defined in the file.
  - `split` : This is used to split off the output of `termbound` into the part multiplying $\log(y)^2$ and the lower degree (in $\log(y)$) terms, for the purpose of applying the $y \log(y)^2$ trick.
  - `num` : Convert a polynomial in $x, y, \log(x), \log(y)$ and `piv`, `l2` to T, by replacing `piv` and `l2` with intervals.
  - `trunc` : Truncate a polynomial in the specified variable up to the specified degree.
  - `clean` : Truncate in two variables up to the specified degrees.
  - `condense` : Combine the inputs (a term bound, and an error bound coming from truncation) into a final error bound.
  - `varquoterrorbound` : This is the bound of Lemma 4.11, with x_0 and y_0 being the (interval) coordinate inputs, `cutoff` being the truncation degree N, and `lowbd` being M, with one additional modification: the bound is divided by $\delta_x^{m_x} \delta_y^{m_y}$.
  - `reducepoly` : Given some polynomial in $x, y, \log(x), \log(y)$, it "reduces" the degree in x and y to at most 1 by replacing most of the factors of x and y in every monomial by the upper bound from the x and y intervals supplied to the function. It is used in the verification at the tiny corners.
  - `checktinycorner` : Verify positivity in the tiny corner, i.e., the square $[0, 1/800] \times [0, 1/800]$. This is done by first reducing the polynomial as above, and then (since $\log(x)$ and $\log(y)$ are negative), checking that coefficient of $\log(x)^i \log(y)^j$ has every monomial of the correct sign $(-1)^{i+j}$, when $i + j$ is strictly positive. This allows us to replace $\log(x)$ and $\log(y)$ by their upper bounds (which are negative numbers), and check that the resulting linear polynomial is nonnegative at the vertices of the tiny corner.

- `varerrorbound` : This is the bound of Lemma 4.10, with inputs and modification as in `varquoterrorbound`.
- `errorbound1varpol` : This is an easy 1-variable version of `varerrorbound`.
- `logquot` : This is the function $(x, y) \mapsto (\log(x) - \log(y))/(x - y)$, extended along the diagonal by its limit.
- `logquotinterval` : This is the interval version of `logquot`.
- `subsH` : Horner evaluation of a polynomial `pol`, setting the variable `var` to the value `val`.
- `subsHtrunc` : Horner evaluation of `pol` setting `var` to `val`, while truncating up to degree `mdeg` in the variable `v`.
- `subsH2var` : Two-variable Horner evaluation.
- `subsHxy` : Two-variable Horner evaluation in `xR` and `yR`.
- `evalRdir` : Horner evaluation of a poly in `xR, yR, LxR, LyR` with an extra argument for the "log term" which multiplies $(\log(x) - \log(y))/(x - y)$, and a direction argument (which specifies the order of evaluation for `LxR` and `LyR`); when x is smaller than y, we want to put `LxR` in the outer loop, and vice versa.
- `serbd` : This is the bound of Lemma 4.1.
- `termbound` : Collects the bound of Corollary 4.2 for each coefficient of $\log(x)^i \log(y)^j$ into an array.
- `quot` : Computes the quotient $(f(x, y) - f(y, y))/(x - y)$ for a polynomial f.
- `CP` : This is the main high-level engine for checking positivity of a function f which is provided to it, on a rectangle $Z = [a, b] \times [c, d]$. It maintains a list of rectangles (initially just Z). For each entry $[a_0, b_0, c_0, d_0]$ in the list, it simply evaluates the function on the interval $x = [a_0, b_0]$ and $y = [c_0, d_0]$, and removes the entry if the result is a strictly positive interval. Otherwise it divides the rectangle into two (if it is very thin) or four equal parts, and replaces the original rectangle by the new ones (these are pushed to the end of the list). When the list is empty, positivity of f on the original rectangle is certified. If at any point, the result is strictly negative, the function returns `False`.
- `timesofar` : Print the time since the start time was initialized.
- `totaltime` : Print the total time taken since start.
- `ellmonomialderiv` : Compute the derivative of a monomial in x, y, $\log(x), \log(y)$, $\log(1 - x), \log(1 - y)$ and $A_i(x)$ and $A_i(y)$ for $1 \leq i \leq 4$ with respect to the specified variable (x or y). This is done by hard-coding the derivatives of the elliptic functions, and using the product rule.
- `ellderivpoly` : Compute the derivative of a polynomial in S by summing over monomials.
- `ellderiv` : Compute the derivative of a rational function (i.e., in the fraction field of S) by the quotient rule.
- `absbound` : Compute the error bound for truncating a polynomial `pol` at degree `cutoff`, assuming x and y are bounded by ε. Returns a matrix whose (i, j) entry is the error bound for the coefficient multiplying $\log(x)^i \log(y)^j$.
- `varerrorbounduv` : Similar in spirit to `absbound`, this function takes a polynomial p, shift coordinates u_0 and v_0, cutoff degrees N_x and N_y, and an additional parameter M (called `lowbd` here), and returns the error bound for truncating $p(u + u_0, v + v_0)$ at degree N_x in x and N_y in y, with the absolute value bound on the discarded terms being computed as a number in $\mathbb{IR}$ times u^M or v^M. Once again this returns a matrix indexed by powers of $\log(x)$ and $\log(y)$ (which are not evaluated at $x = u + u_0$ and $y = v + v_0$). Note that the use of u and v for the translated variables has nothing to do with the rotated coordinates used in the middle of the square.
- `subsHuv` : Horner evaluation in u and v.

  - `partial` : Similar to `ellderiv`, except that we also allow partial derivatives with respect to $u = x - y$ or $v = 1 - x - y$.
  - `evalatcenter` : Evaluate an element of S at the center of the square $x = y = 1/2$.
  - `plugin` : Plug in the power series for the elliptic functions and $\log(1 - x), \log(1 - y)$ to the specified cutoff degrees, into the polynomial argument, and truncate the result to the same cutoff degrees.
  - `evalRdiruv` : Like `evalRdir` but for the shifted variables u and v. (Again there is no connection with the rotated coordinates used in the middle.)
- `kernels.sage` : We define the kernels Q_8, Q_{24}, and Q_{24}^{trunc} in a concise format, which agrees with the Mathematica code used in the other version of the calculation. We also define $Q_{8,\text{sw}} := Q_8$ and $Q_{8,\text{se}}(x, y) = Q_8(1 - x, y)$, etc., shifting coordinates so that the corner of the translated kernel is at $(x, y) = (0, 0)$. We check the appropriate vanishing conditions (the exponents of x and y dividing these kernels, as well as vanishing of the numerators along $x = y$ and $x + y = 1$).
- `constructions.sage` : In this file, we verify that the kernels defined in `kernels.sage` agree with those constructed in Section 4.4 of [1], and we prove inequalities (3a) and (3b) from Section 6.2 of [1].
- `processcorner.sage` : The main function defined is `processcorner`, which takes in the name of a corner, a cutoff parameter, and a list of tuples $(a, b, c, d, t, e, f, N_x, N_y)$ where $[a, b] \times [c, d]$ is the rectangle over which we want to check positivity, t is the type of the function evaluation ("eval", "hybrid", or "shift"), and e, f, N_x, N_y are additional optional parameters specifying the center of the shift and the x- and y-cutoffs in case of "shift" evaluation. The function does the appropriate power series substitutions and defines the quotient Q_5 for which it needs to check positivity, a diagonal term P_1, and a helper function `procregion` which will iterate over the list of supplied rectangles and evaluation types and check positivity of a function `Qeval` on them. It also checks positivity in the tiny corner.
- `nonsingular.sage` : The main function `procQuadrant` in this file functions similarly to `processcorner.sage` above, except that instead of defining a quotient, it works with just the power series substitutions. Since we are away from the diagonals, no quotient is needed, but we must keep track of the sign of the quantity $(x - y)(1 - x - y)$. Accordingly, the list of tuples passed has elements of the form $(a, b, c, d, e, f, N_x, N_y, s, \beta)$, where $[a, b] \times [c, d]$ is the rectangle to be certified, (e, f) is the shift (usually close to the center of the rectangle), N_x and N_y are the truncation degrees, s is the sign, and β is a boolean parameter which indicates whether the $y \log(y)^2$ intervention has to be applied.
- `middle.sage` : The computation in the middle proceeds as described in the previous section. The partial derivatives up to the fourth order are computed, and then bounds on the fourth partials $Q_{uuuv}, Q_{uuvv}, Q_{uvvv}$ are computed by interval evaluation over a suitably fine mesh. We then check again whether the almost-quadratic-form

$$Q_{uv}(0,0) + Q_{uuv}(0,0)u/2 + Q_{uvv}(0,0)v/2 - M_{uuuv}u^2/6 - M_{uuvv}|uv|/4 - M_{uvvv}v^2/6$$

  is positive on the middle region specified, again by interval evaluation over a fine mesh.
- `rectangles.sage` : This file sets up the formalism to check whether the list of rectangles for which we have certified positivity of the kernels indeed covers the square (or the region $(0, 1) \times (0, 0.49)$ in the truncated kernel case), and also to plot these regions.
- `processcorner24dtrunc.sage` : This is a modification of `processcorner.sage` for the truncated kernel case, necessitated by efficiency considerations. The kernel has to be broken up into two separate pieces, as the bounds for them are of a somewhat different character. The checking, however, is carried out in an entirely similar way.
- `nonsingular24dtrunc.sage` : This is a similar modification of `nonsingular.sage`.

- `verifyall.sage` : Finally, this file loads all the other files and checks the proof. The full verification can be run via the SageMath command `load("verifyall.sage")`.

References

[1] H. Cohn, A. Kumar, S. Miller, D. Radchenko, and M. Viazovska, *Universal optimality of the E_8 and Leech lattices and interpolation formulas*, Annals of Mathematics, to appear. arXiv:1902.05438

[2] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, eds., *NIST Handbook of Mathematical Functions*, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, DC and Cambridge University Press, Cambridge, 2010. `https://dlmf.nist.gov/`

[3] The Sage Developers, *SageMath, the Sage Mathematics Software System* (Version 9.5), 2022. `https://www.sagemath.org`

[4] W. Tucker, *Validated numerics for pedestrians*, European Congress of Mathematics, 851–860, Eur. Math. Soc., Zürich, 2005.

Microsoft Research New England, Cambridge, MA, USA
Email address: `cohn@microsoft.com`

Stony Brook University, Stony Brook, NY, USA
Email address: `thenav@gmail.com`

Rutgers University, Piscataway, NJ, USA
Email address: `miller@math.rutgers.edu`

Max Planck Institute for Mathematics, Bonn, Germany
Email address: `danradchenko@gmail.com`

École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
Email address: `viazovska@gmail.com`