Rev. Mat. Iberoam. **32** (2016), no. 3, 1039–1126
DOI 10.4171/RMI/908

Fitting a Sobolev function to data III

Charles Fefferman, Arie Israel, and Garving K. Luli

Abstract. In this paper and two companion papers, we produce efficient algorithms to solve the following interpolation problem: Let $m \geq 1$ and $p > n \geq 1$. Given a finite set $E \subset \mathbb{R}^n$ and a function $f : E \to \mathbb{R}$, compute an extension F of f belonging to the Sobolev space $W^{m,p}(\mathbb{R}^n)$ with norm having the smallest possible order of magnitude; secondly, compute the order of magnitude of the norm of F. The combined running time of our algorithms is at most $CN \log N$, where N denotes the cardinality of E, and C depends only on m, n, and p.

1. Introduction

In our previous papers [3] and [4], we provided efficient algorithms to interpolate data by a function $F\colon \mathbb{R}^n \to \mathbb{R}$ whose Sobolev norm has the least possible order of magnitude. More precisely, let $m \geq 1$ and $p > n \geq 1$. Given a function $f : E \to \mathbb{R}$ with $E \subset \mathbb{R}^n$ finite, we compute a function $F \in W^{m,p}(\mathbb{R}^n)$ such that $F = f$ on E, and $\|F\|_{W^{m,p}} \leq C\|\widetilde{F}\|_{W^{m,p}}$ for any competing function $\widetilde{F} \in W^{m,p}(\mathbb{R}^n)$ such that $\widetilde{F} = f$ on E. Here, C depends only on m, n, and p.

Our computations consist of efficient algorithms, to be implemented on an (idealized) von Neumann computer. In the model of computation assumed in [3] and [4], our computer deals with exact real numbers, without roundoff error. In this paper, we explain how the algorithms, theorems and proofs in [3] and [4] may be modified to allow our algorithms to run successfully on a computer that handles only S-bit machine numbers, for some large, fixed S.

2. Modifications for finite-precision

2.1. The finite-precision model of computation

Our model of computation in finite-precision is a slight variant of that described in Section 38 of [2]. We spell out the details.

Mathematics Subject Classification (2010): Primary 65D17; Secondary 65D05.
Keywords: Algorithm, interpolation, Sobolev spaces.

For an integer $S \geq 1$, we work with "machine numbers" of the form $k \cdot 2^{-S}$, with k an integer and $|k| \leq 2^{+2S}$. Our model of computation consists of an idealized von Neumann computer [5], able to handle machine numbers. We make the following assumptions:

- Given two distinct non-negative machine numbers x and y, we can compute the most significant digit in which the binary expansions of x and y differ. (That is, for $x = \sum_{j \geq -S} x_j 2^j$ and $y = \sum_{j \geq -S} y_j 2^j$ with each x_j and y_j equal to 0 or 1, we compute the largest j for which x_j does not equal y_j.) We assume this takes one unit of "work". See also the paragraph following (2.76).

  This assumption is reasonable, since presumably a machine number is encoded in the computer as the sequence of its binary digits.

- Two machine numbers x and y satisfying $|x| \leq 2^\ell$ and $|y| \leq 2^{\ell'}$ with $\ell, \ell' \geq 0$ and $\ell + \ell' \leq S$ can be "multiplied" to produce a machine number $x \otimes y$ satisfying $|x \otimes y - xy| \leq 2^{-S}$.

  We suppose it takes one unit of "work" to compute $x \otimes y$.

  We assume that $0 \otimes x = x \otimes 0 = 0$ and that $x \otimes 1 = 1 \otimes x = x$.

  We assume that if $|x| \leq 2^\ell$ and $|y| \leq 2^{\ell'}$, for ℓ, ℓ' integers, then $|x \otimes y| \leq 2^{\ell + \ell'}$.

- If x is any machine number other than zero, then we suppose we can produce a machine number "$1/x$" in one unit of "work", such that $|\text{"}1/x\text{"} - 1/x| \leq 2^{-S}$.

  We assume that "$1/x$" $= 1$ when $x = 1$.

  We assume that if $|x| \geq 2^\ell$, for an integer ℓ, then $|\text{"}1/x\text{"}| \leq 2^{-\ell}$.

- Two machine numbers x and y satisfying $|x| \leq \ell$ and $|y| \leq \ell'$ for integers ℓ and ℓ' such that $\ell + \ell' \leq 2^S$ may be added to produce their exact sum $x + y$, which is again a machine number.

  We assume it takes one unit of "work" to compute $x + y$.

- If x is any machine number, then $-x$ is again a machine number.

  We assume it takes one unit of "work" to compute $-x$.

- If x and y are machine numbers, then we can decide whether $x < y$, $y < x$, or $x = y$.

  We assume this takes one unit of "work".

- If x is a machine number other than zero, then we can compute the greatest integer ℓ such that $2^\ell \leq |x|$.

  We assume this takes one unit of "work"

- If x is a machine number and ℓ is an integer with $|\ell| \leq S$, then we can compute the greatest integer $\leq 2^\ell x$. (If this integer lies outside $[-2^S, +2^S]$, then we produce an error message, and abort our computation.)

  We assume this takes one unit of "work".

- We assume we can add, subtract, multiply and divide integers of absolute value $\leq 2^S$, in one unit of "work".

  If we compute x/y in integer arithmetic, for integers x, y ($y \neq 0$) of absolute value at most 2^S, then we obtain the greatest integer $\leq$ the real number x/y. If our desired answer lies outside $[-2^S, +2^S]$, then we produce an error message and abort our computation.
- Given integers x, y of absolute value $\leq 2^S$, we can decide whether $x < y$, $y < x$, or $x = y$.

  We assume this takes one unit of "work".
- If ℓ is an integer, with $|\ell| \leq S$, then we can compute exactly the machine number 2^ℓ.

  We assume this takes one unit of "work".
- If x and y are machine numbers satisfying $2^{-\ell} \leq x \leq 2^\ell$ and $|y| \leq \ell'$ for integers ℓ and ℓ' such that $\ell \cdot \ell' \leq S$, then we can compute a machine number "x^y" in one unit of "work", such that $|\text{"}x^y\text{"} - x^y| \leq 2^{-S}$.
- If x is any positive machine number, then we can compute a machine number "$\log x$" in one unit of "work", such that $|\text{"}\log x\text{"} - \log x| \leq 2^{-S}$. Here, $\log x$ is the base two logarithm.
- We assume we can read or write a machine number from (to) the RAM with one unit of "work".
- We assume we can read a machine number from input or write a machine number to output in one unit of "work".
- We assume we can store a single S-bit word in memory using one unit of "storage".
- We assume we can store the address of any memory cell in a single S-bit word.

Under these assumptions, we say that our computer can process "S-bit machine numbers" (though the actual implementation of those machine numbers seems to require at least $2S + 2$ bits.) We call $\Delta_{\min} = 2^{-S}$ the "machine precision" of our computer.

We fix an integer $\overline{S} \geq 1$. We assume that our computer can process S-bit machine numbers for $S = K_{\max} \cdot \overline{S}$, where $K_{\max} \in \mathbb{N}$ satisfies

$$K_{\max} \geq C, \text{ for a large enough universal constant } C. \tag{2.1}$$

We will show that when our algorithms receive their input as $\overline{S}$-bit machine numbers, then the output produced by our algorithm is accurate to at least $\overline{S}$ bits. We will verify that the work and storage required are as promised: at most $CN \log N$ operations for the one-time work, and at most $C \log N$ operations for the query work, with CN storage, where the constant C depends only on m, n, and p.

Throughout the remaining sections, $\Delta_{\min} = 2^{-S}$ will denote the precision of our computer, as just described.

2.2. Algorithms in finite-precision

We recall that a universal constant is one that depends only on the parameters m, n, and p. We impose the following assumptions in this section.

Main assumptions:

- We set $\Delta_0 := 2^{-\overline{S}}$ for an integer $\overline{S} \geq 1$.
- We assume our computer can process S-bit machine numbers, with $S := K_{\max} \cdot \overline{S}$, where $K_{\max}$ satisfies (2.1). Then $\Delta_{\min} = 2^{-S}$ represents the "machine precision" of our computer. A "machine number" will always denote an S-bit machine number.
- We set $\Delta_g = 2^{-K_1 \overline{S}}$ and $\Delta_\epsilon = 2^{-K_2 \overline{S}}$ for integers $K_1, K_2 \geq 1$.
- We assume that $\Delta_{\min} \leq \Delta_\epsilon^C$ and $\Delta_\epsilon \leq \Delta_g^C$ for a large enough universal constant C.

Assume that $w \in \mathbb{R}$ satisfies $|w| \leq 2^S$. We may not be able to represent w perfectly on a computer, but we can always store an approximation to w. We introduce the relevant notation below.

- We say that w is *specified to precision* Δ_ϵ if a machine number w_{fin} is given with $|w - w_{\text{fin}}| \leq \Delta_\epsilon$.
- We say that w is *computed to precision* Δ_ϵ if there is a finite-precision algorithm that computes a machine number w_{fin} with $|w - w_{\text{fin}}| \leq \Delta_\epsilon$.
- We say that w is *specified* (*computed*) *with parameters* $(\Delta_g, \Delta_\epsilon)$ if $|w| \leq \Delta_g^{-1}$, and if w is specified (computed) to precision Δ_ϵ.

We illustrate this terminology in the next result, which establishes the numerical stability of arithmetic operations.

Lemma 1. *Suppose that* Δ_0, $\Delta_{\min}$, Δ_g, *and* Δ_ϵ *are as in the* ***Main assumptions****. Let* $x, y \in \mathbb{R}$ *be specified with parameters* $(\Delta_g, \Delta_\epsilon)$. *Then the following hold.*

- *We can compute* $x + y$ *with parameters* $(c\Delta_g, C\,\Delta_\epsilon)$.
- *We can compute* $x \cdot y$ *with parameters* $(\Delta_g^2, C\,\Delta_\epsilon\,\Delta_g^{-1})$.
- *If* $|y| \geq \Delta_g$, *we can compute* x/y *with parameters* $(\Delta_g^2, C\,\Delta_\epsilon\,\Delta_g^{-3})$.
- *If* $x \in [\Delta_g, \Delta_g^{-1}]$, *we can compute* $\log x$ *with parameters* $(c\Delta_g, C\,\Delta_\epsilon\,\Delta_g^{-1})$.
- *Suppose that* $x \in [\Delta_g, \Delta_g^{-1}]$ *and* $|y| \leq A$ *with* $A \geq 1$, *and suppose that* $K_{\max} \geq 5A \cdot \max\{K_1, K_2\}$. *Then we can compute* x^y *with parameters* $(\Delta_g^A, \Delta_\epsilon\,\Delta_g^{-C \cdot A})$.

The above computations require work at most C.

Here, the constants c *and* C *are independent of all the parameters.*

Proof. By hypothesis, we suppose we are given machine numbers $\overline{x}, \overline{y}$ with $|x - \overline{x}| \leq \Delta_\epsilon$ and $|y - \overline{y}| \leq \Delta_\epsilon$. Moreover, we have $|x| \leq \Delta_g^{-1}$ and $|y| \leq \Delta_g^{-1}$.

Since $\Delta_\epsilon \leq \Delta_g^{-1}$, we learn that $|\overline{x}| \leq 2\,\Delta_g^{-1}$ and $|\overline{y}| \leq 2\,\Delta_g^{-1}$.

(1) Since $|\overline{x}+\overline{y}| \le 4\Delta_g^{-1} \le \Delta_{\min}^{-1}$ and since $\Delta_{\min}^{-1} = 2^S$, we can compute the (S-bit) machine number $A = \overline{x}+\overline{y}$. This computation requires one unit of work, by assumption on our model of computation. Then

$$|A-(x+y)| \le |\overline{x}-x| + |\overline{y}-y| \le 2\,\Delta_\epsilon.$$

Moreover, $|x+y| \le |x|+|y| \le 2\,\Delta_g^{-1}$.

Thus, we can compute the sum $x+y$ with parameters $(\frac{1}{2}\Delta_g, 2\,\Delta_\epsilon)$.

(2) Since $|\overline{x}\cdot\overline{y}| \le 4\Delta_g^{-2} \le \Delta_{\min}^{-1}$, we can compute a machine number P with $|P-\overline{x}\cdot\overline{y}| \le \Delta_{\min} \le \Delta_\epsilon$. (Recall that $\Delta_{\min}$ is the "machine precision".) We have

$$|x\cdot y-\overline{x}\cdot\overline{y}| \le |x-\overline{x}|\cdot|y| + |y-\overline{y}|\cdot|\overline{x}| \le \Delta_\epsilon\cdot\Delta_g^{-1} + \Delta_\epsilon\cdot(2\,\Delta_g^{-1}) = 3\Delta_\epsilon\cdot\Delta_g^{-1}.$$

Hence, $|P-x\cdot y| \le \Delta_\epsilon + 3\Delta_\epsilon\Delta_g^{-1} \le 4\Delta_\epsilon\,\Delta_g^{-1}$. Moreover, $|x\cdot y| \le \Delta_g^{-1}\cdot\Delta_g^{-1} = \Delta_g^{-2}$.

Therefore, we can compute the product $x\cdot y$ with parameters $(\Delta_g^2, 4\Delta_\epsilon\,\Delta_g^{-1})$.

Here, we have only used the assumptions $\Delta_{\min} \le \frac{1}{4}\Delta_g^2$ and $\Delta_{\min} \le \Delta_\epsilon$

(3) Suppose that $|y| \ge \Delta_g$. Since we may assume $\Delta_\epsilon \le \Delta_g^{10}$, we have

$$|\overline{y}| \ge |y| - |y-\overline{y}| \ge \Delta_g - \Delta_\epsilon \ge \Delta_g - \Delta_g^{10}.$$

Since $\Delta_g \le 1/2$, we conclude that $|\overline{y}| \ge \frac{1}{2}\Delta_g$.

Thus, we can compute a machine number A with $\left|A-(\overline{y})^{-1}\right| \le \Delta_{\min} \le \Delta_\epsilon$.

We have

$$\left|y^{-1}-(\overline{y})^{-1}\right| = \frac{|y-\overline{y}|}{|y|\cdot|\overline{y}|} \le \frac{\Delta_\epsilon}{\Delta_g\cdot\frac{1}{2}\Delta_g} = 2\,\Delta_\epsilon\,\Delta_g^{-2}.$$

Hence, $\left|A-y^{-1}\right| \le \Delta_\epsilon + 2\,\Delta_\epsilon\,\Delta_g^{-2} \le 3\Delta_\epsilon\,\Delta_g^{-2}$. Moreover, $\left|y^{-1}\right| \le \Delta_g^{-1}$.

Therefore, we can compute y^{-1} with parameters $(\Delta_g, 4\Delta_\epsilon\,\Delta_g^{-2})$.

(4) Suppose that $|y| \ge \Delta_g$. According to (3), we can compute y^{-1} with parameters $(\Delta_g, \Delta_\epsilon^{\text{new}})$, where $\Delta_\epsilon^{\text{new}} = 4\Delta_\epsilon\,\Delta_g^{-2}$. We have $\Delta_\epsilon^{\text{new}} \le 4\Delta_g^8 \le 1$ (since $\Delta_\epsilon \le \Delta_g^{10}$) and $\Delta_\epsilon^{\text{new}} \ge \Delta_\epsilon \ge \Delta_{\min}$. Hence, applying (2), we can compute $x\cdot y^{-1}$ with parameters $(\Delta_g^2, 4\,\Delta_\epsilon^{\text{new}}\,\Delta_g^{-1}) = (\Delta_g^2, 16\,\Delta_\epsilon\,\Delta_g^{-3})$.

(5) Suppose that $\Delta_g \le x \le \Delta_g^{-1}$. Since $|x-\overline{x}| \le \Delta_\epsilon \le \Delta_g^{10}$, we have $\frac{1}{2}\Delta_g \le \overline{x} \le 2\,\Delta_g^{-1}$.

We can compute a machine number L satisfying $|L-\log\overline{x}| \le \Delta_{\min} \le \Delta_\epsilon$. Then we have

$$|\log x - \log\overline{x}| \le \frac{1}{\ln 2}\cdot|x-\overline{x}|\cdot\max\left\{x^{-1},(\overline{x})^{-1}\right\} \le C\,\Delta_\epsilon\,\Delta_g^{-1}.$$

(Recall that $\log x$ denotes the base two logarithm.) Hence, $|L-\log x| \le \Delta_\epsilon + C\,\Delta_\epsilon\,\Delta_g^{-1} \le C'\Delta_\epsilon\,\Delta_g^{-1}$. Moreover, $|\log x| \le \log\Delta_g^{-1} \le C\,\Delta_g^{-1}$.

Therefore, we can compute $\log x$ with parameters $(c\Delta_g, C'\Delta_\epsilon\,\Delta_g^{-1})$.

(6) Suppose that $\Delta_g \le x \le \Delta_g^{-1}$ and $|y| \le A$ for some $A \ge 1$.

Since $|x-\overline{x}| \le \Delta_\epsilon \le \Delta_g^{10}$ and $|y-\overline{y}| \le \Delta_\epsilon \le 1$, we conclude that $\frac{1}{2}\Delta_g \le \overline{x} \le 2\,\Delta_g^{-1}$ and $|\overline{y}| \le |y|+|y-\overline{y}| \le 2A$.

We have $\left|\overline{x}^{\overline{y}}\right| \le (2\Delta_g^{-1})^{2A} = 2^{2A\cdot(K_2\overline{S}+1)} \le \Delta_{\min}^{-1}$, due to the assumption that $\Delta_{\min} = 2^{-K_{\max}\overline{S}}$ and $K_{\max} \ge 4AK_2$. Thus, we can compute a machine number B with

$$\left|B - \overline{x}^{\overline{y}}\right| \le \Delta_{\min} \le \Delta_\epsilon.$$

This requires work at most C.

We have

$$\begin{aligned}\left|x^y - \overline{x}^{\overline{y}}\right| &\le |x^y - \overline{x}^y| + \left|\overline{x}^y - \overline{x}^{\overline{y}}\right| = \left|e^{y\ln x} - e^{y\ln\overline{x}}\right| + \left|e^{y\ln\overline{x}} - e^{\overline{y}\ln\overline{x}}\right| \\ &\le |y|\cdot|x-\overline{x}|\cdot\max\left\{x^{-1},(\overline{x})^{-1}\right\}\cdot\max\left\{e^{y\ln x}, e^{y\ln\overline{x}}\right\} \\ &\quad + |y-\overline{y}|\cdot|\ln\overline{x}|\cdot\max\left\{e^{y\ln\overline{x}}, e^{\overline{y}\ln\overline{x}}\right\} \\ &\le A\cdot\Delta_\epsilon\cdot 2\Delta_g^{-1}\cdot\Delta_g^{-CA} + \Delta_\epsilon\cdot\ln\left(2\Delta_g^{-1}\right)\cdot\Delta_g^{-CA} \le \Delta_\epsilon\,\Delta_g^{-C'A}.\end{aligned}$$

In the above, we use the estimates $|e^w - e^z| \le |w-z|\cdot\max\{e^w, e^z\}$ and $|\ln x - \ln\overline{x}| \le |x-\overline{x}|\cdot\max\left\{x^{-1},(\overline{x})^{-1}\right\}$; both C and C' are numerical constants.

Hence, $|B - x^y| \le \Delta_\epsilon + \Delta_\epsilon\,\Delta_g^{-C'A} \le \Delta_\epsilon\,\Delta_g^{-C''A}$. Moreover, $|x^y| \le \Delta_g^{-A}$.

Therefore, we can compute x^y with parameters $(\Delta_g^A, \Delta_\epsilon\,\Delta_g^{-C''A})$ for a numerical constant C''.

Thanks to (1)–(6), the conclusions of the lemma are verified. This completes the proof. □

We finish the section with a technical lemma concerning the evaluation of the ℓ^p norm by a finite-precision algorithm.

Lemma 2. *Let $\Delta \in [\Delta_g, 1]$ be a given machine number. Fix an $\overline{S}$-bit machine number $p > 1$. Given real numbers x_j $(1 \le j \le J)$ with parameters $(\Delta_g, \Delta_\epsilon)$, where $J \le \Delta_g^{-1}$, we define*

$$A := \Big(\sum_{1\le j\le J} |x_j|^p + \Delta^p\Big)^{1/p}.$$

Then there is a finite-precision algorithm, requiring work and storage at most $C\cdot J$, which computes a machine number $\widehat{A}$ that satisfies $\frac{1}{2}\cdot A \le \widehat{A} \le 2\cdot A$.

Proof. All constants in the proof denoted by C, C', etc., will depend only on p. We write $\Delta_1 \ll \Delta_2$ to indicate that $\Delta_1 \le \Delta_2^C$ for a sufficiently large universal constant C. We set $\Delta_1 = \Delta_g^{C_0}$ for a sufficiently large universal constant $C_0 \in \mathbb{N}$ that will be determined later. Thus, in the recently introduced notation, we have $\Delta_1 \ll \Delta_g$. By hypothesis, we are given a machine number x_j^* with $\left|x_j^* - x_j\right| \le \Delta_\epsilon$, and we guarantee that $|x_j| \le \Delta_g^{-1}$ for each j. We define

$$B := \Big(\sum_{\substack{1\le j\le J\\ |x_j^*|\ge\Delta_1}} |x_j^*|^p + \Delta^p\Big)^{1/p}. \tag{2.2}$$

Note that

$$|A^p - B^p| \le \sum_{1\le j\le J} \big|\, |x_j|^p - |x_j^*|^p \,\big| + \sum_{\substack{1\le j\le J\\ |x_j^*|<\Delta_1}} |x_j|^p .$$

Since $|x_j - x_j^*| \le \Delta_\epsilon$ and $|x_j|, |x_j^*| \le \Delta_g^{-C}$, the first sum is bounded by $CJ\Delta_\epsilon \cdot \Delta_g^{-C'}$. Since $|x_j| \le \Delta_1 + \Delta_\epsilon \le 2\Delta_1$ whenever $|x_j^*| < \Delta_1$, the second sum is bounded by $CJ\Delta_1^p$. Thus, we have $|A^p - B^p| \le CJ\Delta_\epsilon \cdot \Delta_g^{-C} + CJ\Delta_1^p$. We obtain the bound $|A^p - B^p| \le \Delta_g^{-C''}\Delta_1^p$ for a universal constant C'', because $J \le \Delta_g^{-1}$ and because we may assume that $\Delta_\epsilon \le \Delta_g^{C_0 p} = \Delta_1^p$. Note that A^p and B^p are at least Δ^p. Thus, by the mean value theorem, we have

$$|A - B| \le |A^p - B^p| \cdot \max_{t\in[\Delta^p,\infty)} \left|\frac{d}{dt}(t^{1/p})\right| \le \Delta_g^{-C''}\Delta_1^p \cdot \frac{1}{p}\Delta^{1-p} \le \Delta_1^p \Delta_g^{-C'''}.$$

Here, in the last estimate we use that $\Delta \ge \Delta_g$. Note that C', C'', C''' above are independent of C_0.

All the summands inside the parentheses in (2.2) are at least Δ_1^p (recall that $\Delta \ge \Delta_g \ge \Delta_1$). Also, the number of summands is at most $J+1 \le C\,\Delta_g^{-1} \le C\,\Delta_1^{-1}$. Therefore, by the numerical stability of arithmetic (see Lemma 1) we can compute a machine number B_{fin} such that

$$|B - B_{\text{fin}}| \le \Delta_\epsilon\, \Delta_1^{-C}.$$

We conclude that $|A - B_{\text{fin}}| \le \Delta_1^p \Delta_g^{-C'''} + \Delta_\epsilon\, \Delta_1^{-C}$. We recall that $\Delta_1 = \Delta_g^{C_0}$ and $\Delta_\epsilon \ll \Delta_g$. So, if we pick a sufficiently large universal constant $C_0 \in \mathbb{N}$ then we can guarantee that $|A - B_{\text{fin}}| \le \frac{1}{2}\Delta_g$. Note that $A \ge \Delta \ge \Delta_g$. Thus, we conclude that A and B_{fin} differ by at most a factor of 2. We can therefore define $\widehat{A} = B_{\text{fin}}$ and the conclusion of the lemma follows. □

2.3. Short form

Let $E = \{z_1, \dots, z_N\} \subset \frac{1}{32}Q^\circ$, where $Q^\circ = [0,1)^n$.

We write $\mathbb{X}(E)$ for the space of all real-valued functions f on E, equipped with the trace norm induced by $\mathbb{X} = L^{m,p}(\mathbb{R}^n)$.

Recall that $\mathcal{P}$ denotes the set of all polynomials on $\mathbb{R}^n$ of degree at most $m-1$, and $\mathcal{M}$ denotes the set of all multiindices $\alpha = (\alpha_1, \dots, \alpha_n)$ with $|\alpha| \le m-1$.

We let $\Delta_{\min} \le \Delta_\epsilon \le \Delta_g \le \Delta_0$ be defined as in the **Main assumptions** in Section 2.2. In particular, recall that $\Delta_{\min} = 2^{-S}$ denotes the machine precision of our computer. When we refer to a "machine number" we will always mean an S-bit machine number.

Any linear functional $\omega : \mathbb{X}(E) \to \mathbb{R}$ can be expressed in the form

$$\omega(f) = \sum_{\ell=1}^{L} \lambda_\ell \cdot f(z_{j_\ell}). \tag{2.3}$$

We call (2.3) a *short form* of ω. We do not promise that the coefficients λ_ℓ are nonzero. Thus, in contrast to the notation in infinite-precision, a functional can have more than one short form. The *depth* of ω, denoted $\mathrm{depth}(\omega)$, is the number L. Note that $\mathrm{depth}(\omega)$ depends on the short form (2.3) of ω, and not on ω alone. This abuse of notation should cause no confusion.

The short form (2.3) is given with parameters $(\Delta_g, \Delta_\epsilon)$ if the numbers λ_ℓ are given with parameters $(\Delta_g, \Delta_\epsilon)$, and if the list $j_1, \ldots, j_L$ is given. Recall that this means we specify machine numbers $\overline{\lambda}_\ell$ with $|\lambda_\ell - \overline{\lambda}_\ell| \leq \Delta_\epsilon$, and we promise that $|\lambda_\ell| \leq \Delta_g^{-1}$ for each ℓ. The indices $j_1, \ldots, j_L$ may be represented as pointers to the memory locations in which the corresponding points of E are stored. We assume that each of these pointers is stored using a single unit of memory.

Let $\Omega = \{\omega_1, \ldots, \omega_M\}$ be a list of linear functionals on $\mathbb{X}(E)$.

A functional $\xi\colon \mathbb{X}(E) \to \mathbb{R}$ has Ω-assisted depth d provided that it can be written in the form

$$\xi(f) = \eta(f) + \sum_{\nu=1}^{\nu_{\max}} \mu_\nu \cdot \omega_{k_\nu}(f), \tag{2.4}$$

where η is a linear functional and $\mathrm{depth}(\eta) + \nu_{\max} \leq d$. We call (2.4) a *short form* of ξ. Note that perhaps we can write a given ξ in many different ways in short form.

The short form (2.4) is given with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of assists Ω if the functional η is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$, the numbers μ_ν are given with parameters $(\Delta_g, \Delta_\epsilon)$, and a list of the indices $k_{\nu_1}, \ldots, k_{\nu_{\max}}$ is given.

A functional $\xi\colon \mathbb{X}(E) \oplus \mathcal{P} \to \mathbb{R}$ has Ω-assisted depth d provided that it can be written in the form

$$\xi(f, P) = \eta(f) + \sum_{\nu=1}^{\nu_{\max}} \mu_\nu \cdot \omega_{k_\nu}(f) + \sum_{\alpha \in \mathcal{M}} \theta_\alpha \cdot \frac{1}{\alpha!} \partial^\alpha P(0), \tag{2.5}$$

where η is a linear functional and $\mathrm{depth}(\eta) + \nu_{\max} + \#(\mathcal{M}) \leq d$. We call (2.5) a *short form* of ξ.

The short form (2.5) is given with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of assists Ω if the functional η is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$, the numbers μ_ν and θ_α are given with parameters $(\Delta_g, \Delta_\epsilon)$, and a list of the indices $k_{\nu_1}, \ldots, k_{\nu_{\max}}$ is given.

A linear map $T\colon \mathbb{X}(E) \oplus \mathcal{P} \to \mathcal{P}$ is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of assists Ω, if for each $\beta \in \mathcal{M}$ we exhibit a formula

$$\partial^\beta (T(f, P))(0) = \eta_\beta(f) + \sum_{\nu=1}^{\nu_{\max}} \mu_{\beta\nu} \cdot \omega_{k_\nu}(f) + \sum_{\alpha \in \mathcal{M}} \theta_{\beta\alpha} \cdot \frac{1}{\alpha!} \partial^\alpha P(0),$$

where the functional η_β is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$, the numbers $\mu_{\beta\nu}$ and $\theta_{\beta\alpha}$ are given with parameters $(\Delta_g, \Delta_\epsilon)$, and a list of the indices $k_1, \ldots, k_{\nu_{\max}}$ is given.

Similarly, a linear map $T\colon \mathbb{X}(E) \to \mathcal{P}$ is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of assists Ω, if for each $\beta \in \mathcal{M}$ we exhibit a formula

$$\partial^\beta(T(f))(0) = \eta_\beta(f) + \sum_{\nu=1}^{\nu_{\max}} \mu_{\beta\nu} \cdot \omega_{k_\nu}(f),$$

where the functional η_β is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$, the numbers $\mu_{\beta\nu}$ are given with parameters $(\Delta_g, \Delta_\epsilon)$, and a list of the indices $k_1, \ldots, k_{\nu_{\max}}$ is given.

We say we have computed a linear map $T\colon \mathbb{X}(E) \to \mathbb{X}$ in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of assists Ω if for each $\overline{S}$-bit machine point $\underline{x} \in Q^\circ$ and each multiindex $\alpha \in \mathcal{M}$, we can compute a short form of the linear functional

$$f \mapsto \partial^\alpha Tf(\underline{x})$$

with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of the assists Ω. If the functional $f \mapsto \partial^\alpha(Tf)(\underline{x})$ has Ω-assisted depth d, for all $\underline{x} \in \mathbb{R}^n$ and $\alpha \in \mathcal{M}$, then we say that the map T has Ω-assisted depth d. We extend this notation to linear maps $T\colon \mathbb{X}(E) \oplus \mathcal{P} \to \mathbb{X}$ in the obvious way. We only answer queries if $\underline{x} \in Q^\circ$ because enormous $\underline{x}$'s might lead to overflow errors.

2.4. Main algorithms in finite-precision

Our main theorem concerns extension operators for homogeneous Sobolev spaces and is stated below. Later, in Section 2.18.2, we will present a corresponding result for inhomogeneous Sobolev spaces (see Theorem 2).

We write c, C, C', etc., to denote universal constants, which depend only on m, n, and p.

Let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. We call x an S_0-bit "machine point" if each coordinate x_k is an S_0-bit machine number.

Theorem 1. *There exists a universal constant $C \geq 1$ such that the following holds.*

Let $\overline{S} \geq 1$ be an integer. We fix an $\overline{S}$-bit machine number $p > n$.

We also fix a subset $E \subset \frac{1}{32}Q^\circ$ consisting of $\overline{S}$-bit machine points, with $\#(E) = N \geq 2$, where $Q^\circ = [0,1)^n$.

We assume we are given constants $\Delta_{\min} = 2^{-K_{\max}\overline{S}}$, $\Delta_\epsilon^\circ := 2^{-K_1\overline{S}}$, $\Delta_g^\circ := 2^{-K_2\overline{S}}$, and $\Delta_{\text{junk}}^\circ := 2^{-K_3\overline{S}}$, for integers $K_1, K_2, K_3, K_{\max} \geq 1$ such that $K_{\max} \geq C \cdot K_1 \geq C^2 \cdot K_2 \geq C^3 \cdot K_3 \geq C^4$.

We assume that our computer can perform arithmetic operations on S-bit machine numbers with precision $\Delta_{\min} = 2^{-S}$, where $S = K_{\max}\overline{S}$.

We compute (see below) lists Ω and Ξ, consisting of linear functionals on $\mathbb{X}(E) = \{f : E \to \mathbb{R}\}$, with the following properties.

- *The sum of* $\operatorname{depth}(\omega)$ *over all $\omega \in \Omega$ is bounded by CN, and $\#[\Xi] \leq CN$.*
- *Each ξ in Ξ has Ω-assisted depth at most C.*
- *We compute each $\omega \in \Omega$ in short form with parameters $(\Delta_g^\circ, \Delta_\epsilon^\circ)$, and we compute each $\xi \in \Xi$ in short form with parameters $(\Delta_g^\circ, \Delta_\epsilon^\circ)$ in terms of the assists Ω.*

- *For every* $f \in \mathbb{X}(E)$*, we have*

$$C^{-1}\|f\|_{\mathbb{X}(E)} \le \Big[\sum_{\xi\in\Xi} |\xi(f)|^p\Big]^{1/p} \le C \inf\left\{\|F\|_{\mathbb{X}} + \Delta^{\circ}_{\text{junk}}\|F\|_{L^p(Q^\circ)} : F \in \mathbb{X},\ F = f \text{ on } E\right\}$$

Moreover, there exists a linear map $T\colon \mathbb{X}(E) \to \mathbb{X}$ *with the following properties.*

- T *has* Ω*-assisted depth at most* C*.*
- $Tf = f$ *on* E*, and*

$$\|Tf\|_{\mathbb{X}} \le C \inf\left\{\|F\|_{\mathbb{X}} + \Delta^{\circ}_{\text{junk}}\|F\|_{L^p(Q^\circ)} : F \in \mathbb{X},\ F = f \text{ on } E\right\}$$

for every $f \in \mathbb{X}(E)$*.*

- *We produce a query algorithm with the following properties.*

  Given an $\overline{S}$*-bit machine point* $\underline{x} \in Q^\circ$*, and given* $\alpha \in \mathcal{M}$*, we compute a short form of the linear functional* $f \mapsto \partial^\alpha (Tf)(\underline{x})$ *in terms of the assists* Ω*. This linear functional is computed with parameters* $(\Delta^\circ_g, \Delta^\circ_\epsilon)$*. This computation requires work at most* $C \log N$*.*

The above computations require one-time work at most $CN \log N$ *in space* CN*.*

2.5. Bases for the space of polynomials

We discuss the first algorithm in the infinite-precision text, namely the algorithm FIT BASIS TO CONVEX BODY from Section 2.7.3 of [3]. This is a preparatory algorithm that will be used later in the text. Our finite-precision version of FIT BASIS TO CONVEX BODY will require several additional assumptions, stated below.

We impose the assumptions in Theorem 1. In particular, $\Delta_{\min} \le \Delta_\epsilon \le \Delta_g \le \Delta_0$ are as in the **Main assumptions** in Section 2.2. Our computer can perform arithmetic operations on S-bit machine numbers with precision $\Delta_{\min} = 2^{-S}$, where $S = K_{\max} \cdot \overline{S}$.

We introduce a few conventions that are used in the rest of the paper. A machine number will mean an S-bit machine number, and a machine point will mean an S-bit machine point. A bounded interval $I \subset \mathbb{R}$ is called a machine interval if its endpoints are machine numbers.

We assume

$$p > n \text{ is an } \overline{S}\text{-bit machine number,} \tag{2.6}$$

while

$$x \in \mathbb{R}^n \text{ is an } (S\text{-bit}) \text{ machine point.} \tag{2.7}$$

Recall that $\mathcal{P}$ is the vector space of polynomials on $\mathbb{R}^n$ of degree at most $m-1$, and we denote $D = \dim \mathcal{P}$. We identify $\mathcal{P}$ with $\mathbb{R}^D$, by identifying $P \in \mathcal{P}$ with

$\left(\frac{1}{\alpha!}\partial^\alpha P(x)\right)_{\alpha\in\mathcal{M}}$, where $\mathcal{M}$ denotes the set of all multiindices of order at most $m-1$. We define

$$|P|_x = \Big(\sum_{\alpha\in\mathcal{M}} |\partial^\alpha P(x)|^p\Big)^{1/p}.$$

We assume we are given $\Lambda \geq 1$. We write $c(\Lambda)$, $C(\Lambda)$, etc. to denote constants depending on m, n, p, and Λ. We write c, $\tilde{c}$, C, etc. to denote constants depending only on m, n, and p.

We are given a quadratic form q on $\mathcal{P}$. We assume that q is specified as a $D \times D$ matrix $(q_{\beta\gamma})_{\beta,\gamma\in\mathcal{M}}$ acting on the above coordinates:

$$q(P) = \sum_{\beta,\gamma\in\mathcal{M}} q_{\beta\gamma}\cdot\partial^\beta P(x)\cdot\partial^\gamma P(x). \tag{2.8}$$

We assume that the matrix $(q_{\alpha\beta})_{\alpha,\beta\in\mathcal{M}}$ satisfies the following conditions:

$$|q_{\alpha\beta}| \leq \Delta_g^{-1}, \text{ and } q_{\alpha\beta} \text{ is specified to precision } \Delta_\epsilon, \text{ for all } \alpha,\beta\in\mathcal{M}. \tag{2.9}$$

$$(q_{\alpha\beta})_{\alpha,\beta\in\mathcal{M}} \geq \Delta_g\cdot(\delta_{\alpha\beta})_{\alpha,\beta\in\mathcal{M}}. \tag{2.10}$$

From (2.10) we learn that

$$|q(P)| \geq c\,\Delta_g\cdot|P|_x^2 \quad \text{for all } P\in\mathcal{P}, \tag{2.11}$$

for a universal constant $c > 0$.

We fix a multiindex set $\mathcal{A}\subset\mathcal{M}$. The main result of the section is as follows.

ALGORITHM: FIT BASIS TO CONVEX BODY (FINITE-PRECISION VERSION)

Given q, x, $\mathcal{A}$ as above: We compute a partition of $[\Delta_g,\Delta_g^{-1}]$ into machine intervals I_ℓ, and for each ℓ we compute machine numbers λ_ℓ, c_ℓ with $c_\ell \geq 0$, such that the function $\eta_* : [\Delta_g,\Delta_g^{-1}] \to \mathbb{R}$, defined by

$$\eta_*(\delta) := c_\ell\cdot\delta^{\lambda_\ell} \quad \text{for } \delta\in I_\ell,$$

has the following properties.

- Let $\overline{\overline{\sigma}}$ satisfy $\{q\leq\Lambda^{-1}\}\subset\overline{\overline{\sigma}}\subset\{q\leq\Lambda\}$. Then, for any $\delta\in[\Delta_g,\Delta_g^{-1}]$,
  - $\overline{\overline{\sigma}}$ has an $(\mathcal{A},x,\eta^{1/2},\delta)$-basis for any $\eta > C(\Lambda)\cdot\eta_*(\delta)$,
  - $\overline{\overline{\sigma}}$ does not have an $(\mathcal{A},x,\eta^{1/2},\delta)$-basis for any $\eta < c(\Lambda)\cdot\eta_*(\delta)$.

  (See Section 2.7.1 of [3] for the definition of a basis for a convex set of polynomials.)
- Moreover, $c\cdot\eta_*(\delta_1)\leq\eta_*(\delta_2)\leq C\cdot\eta_*(\delta_1)$ whenever $\frac{1}{10}\delta_1\leq\delta_2\leq 10\delta_1$.
- Also, $\eta_*(\delta)\geq\Delta_g^C$ for any $\delta\in[\Delta_g,\Delta_g^{-1}]$.
- The numbers c_ℓ belong to the interval $[\Delta_g^C,\Delta_g^{-C}]$, and the exponents λ_ℓ are of the form $\mu+\nu/p$ for integers μ,ν with $|\mu|,|\nu|\leq C$.
- The computation of I_ℓ, λ_ℓ, and c_ℓ, requires work and storage at most C.

Here, $c > 0$ and $C\geq 1$ are constants determined by m, n, and p, while $c(\Lambda)$ and $C(\Lambda)$ are constants determined by m,n,p, and Λ.

Explanation. We recall the basic structure of the argument given in Section 2.7.3 of [3]. We first define a rational function $\eta_{\min}(\delta)$ with nice properties, and then we explain how to compute an approximation $\eta_*(\delta)$ for $\eta_{\min}(\delta)$. The structure of our argument here is quite similar. The main difference being that we need to take additional care to ensure the numerical stability of our computations with respect to rounding error.

We consider the quadratic form

$$\begin{aligned} M^\delta(\vec{P}) &:= \sum_{\alpha\in\mathcal{A}} q\left(\delta^{m-n/p-|\alpha|}P_\alpha\right) + \sum_{\substack{\alpha\in\mathcal{A},\beta\in\mathcal{M}\\ \beta>\alpha}} \left(\delta^{|\beta|-|\alpha|}\partial^\beta P_\alpha(x)\right)^2 \\ &= \sum_{\alpha\in\mathcal{A}}\sum_{\beta,\gamma\in\mathcal{M}} \delta^{2(m-n/p-|\alpha|)} q_{\beta\gamma}\cdot\partial^\beta P_\alpha(x)\cdot\partial^\gamma P_\alpha(x) \\ &\qquad + \sum_{\substack{\alpha\in\mathcal{A},\beta\in\mathcal{M}\\ \beta>\alpha}} \left(\delta^{|\beta|-|\alpha|}\partial^\beta P_\alpha(x)\right)^2 \quad (\text{for } \vec{P}=(P_\alpha)_{\alpha\in\mathcal{A}}), \end{aligned} \tag{2.12}$$

restricted to the affine subspace

$$H := \left\{\vec{P}=(P_\alpha)_{\alpha\in\mathcal{A}} : \partial^\beta P_\alpha(x) = \delta_{\beta\alpha} \text{ for } \alpha,\beta\in\mathcal{A}\right\}.$$

Let

$$\eta_{\min}(\delta) := \min_{\vec{P}\in H} M^\delta(\vec{P}),$$

which is regarded as a function of $\delta \in \left[\Delta_g, \Delta_g^{-1}\right]$.

Recall from Section 2.7.3 of [3], we showed that

$$\eta_{\min}(\delta_1) \le \eta_{\min}(\delta_2) \le \left(\frac{\delta_2}{\delta_1}\right)^{2m} \eta_{\min}(\delta_1) \text{ for } \delta_1 \le \delta_2, \tag{2.13}$$

and

$$\begin{aligned} &\overline{\overline{\sigma}} \text{ has a } (\mathcal{A}, x, \eta^{1/2}, \delta)\text{-basis if } \eta > C(\Lambda)\cdot\eta_{\min}(\delta), \\ &\qquad\qquad\qquad\qquad \text{but not if } \eta < c(\Lambda)\cdot\eta_{\min}(\delta). \end{aligned} \tag{2.14}$$

Using (2.11), we see that

$$|M^\delta(\vec{P})| \ge c\Delta_g^{2m+1}\cdot\sum_{\alpha\in\mathcal{A}}|P_\alpha|_x^2 \quad \text{for any } \vec{P}\in H,\ \ \delta\in\left[\Delta_g,\Delta_g^{-1}\right]. \tag{2.15}$$

Furthermore, if $\vec{P}\in H$ and $\alpha\in\mathcal{A}$ then $\partial^\alpha P_\alpha(x) = 1$, hence $|P_\alpha|_x \ge 1$. Thus, by definition of $\eta_{\min}(\delta)$ as the minimum of $M^\delta(\cdot)$ on H, we have

$$\eta_{\min}(\delta) \ge \widetilde{c}\,\Delta_g^{2m+1}, \tag{2.16}$$

for a universal constant $\widetilde{c} > 0$.

Next, we compute a piecewise-rational function $\widetilde{\eta}_{\min}(\delta)$ that approximates $\eta_{\min}(\delta)$. For $w = (w_{\alpha\beta})_{\alpha\in\mathcal{A},\beta\in\mathcal{M}\setminus\mathcal{A}} \in \mathbb{R}^J$ we set

$$P_\alpha^w(z) := \frac{1}{\alpha!}\cdot(z-x)^\alpha + \sum_{\beta\in\mathcal{M}\setminus\mathcal{A}} \frac{1}{\beta!}\cdot w_{\alpha\beta}\cdot(z-x)^\beta \quad (\alpha\in\mathcal{A}). \tag{2.17}$$

This gives a coordinate mapping $w \mapsto \vec{P}^w = (P_\alpha^w)_{\alpha\in\mathcal{A}} \in H$, and we set

$$\begin{aligned}\widetilde{M}^\delta(w) &:= M^\delta(\vec{P}^w)\\ &= \langle A^\delta w, w\rangle - 2\langle b^\delta, w\rangle + m^\delta \quad (w\in\mathbb{R}^J).\end{aligned} \tag{2.18}$$

Here, A^δ is a matrix, b^δ is a vector, m^δ is a scalar - all functions of δ - and $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product on $\mathbb{R}^J$. The entries of A^δ, b^δ, and m^δ are all sums of monomials of the form $a\cdot\delta^{\mu+\nu/p}$ with $\mu,\nu\in\mathbb{Z}$ and $a\in\mathbb{R}$.

We have $\|w\|^2 = \sum_{\alpha,\beta}|w_{\alpha\beta}|^2 \le c\sum_\alpha |P_\alpha^w|_x^2$, since $w_{\alpha\beta} = \partial^\beta P_\alpha^w(x)$ for $\alpha\in\mathcal{A}$, $\beta\in\mathcal{M}\setminus\mathcal{A}$. Thus, from (2.15) we have $\left|\widetilde{M}^\delta(w)\right| \ge c\Delta_g^{2m+1}\cdot\|w\|^2$, hence

$$A^\delta \ge c\,\Delta_g^{2m+1}\cdot(\delta_{ij}), \tag{2.19}$$

for a universal constant $c>0$. In particular, the matrix A^δ is invertible.

Recall that $A^\delta = (A_{ij}^\delta)$, $b^\delta = (b_i^\delta)$, and m^δ are given in the form

$$\begin{cases} A_{ij}^\delta = \sum_{\mu,\nu} c_{\mu\nu}^{ij}\delta^{\mu+\nu/p} & (1\le i,j\le J),\\ b_j^\delta = \sum_{\mu,\nu} c_{\mu\nu}^{j}\delta^{\mu+\nu/p} & (1\le j\le J),\\ m^\delta = \sum_{\mu,\nu} c_{\mu\nu}\delta^{\mu+\nu/p}. \end{cases} \tag{2.20}$$

There are at most C pairs $(\mu,\nu)\in\mathbb{Z}\times\mathbb{Z}$ relevant to the above sums, and we have $|\mu|,|\nu|\le C$ for each pair. (See (2.53)–(2.55) in [3].)

We insert the formula (2.17) for the polynomials $P_\alpha = P_\alpha^w$ $(\alpha\in\mathcal{A})$ in the second line of the definition (2.12) of M_δ to produce the expression $\widetilde{M}^\delta(w) = \langle A^\delta w, w\rangle - 2\langle b^\delta, w\rangle + m^\delta$. We compute each of the numbers $c_{\mu\nu}^{ij}$, $c_{\mu\nu}^{j}$, and $c_{\mu\nu}$ as a linear combination of the entries of $(q_{\alpha\beta})$ (and the constant 1). Hence, since the $q_{\alpha\beta}$ are given with parameters $(\Delta_g,\Delta_\epsilon)$, the numbers $c_{\mu\nu}^{ij}$, $c_{\mu\nu}^{j}$, $c_{\mu\nu}$ can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

We can compare exponents of the form $\lambda = \mu+\nu/p$ and $\overline{\lambda} = \overline{\mu}+\overline{\nu}/p$ by expressing λ, $\overline{\lambda}$ as a ratio of integers and cross-multiplying (recall that p is an $\overline{S}$-bit machine number). This comparison requires at most C units of work. By summing the coefficients of the monomials with the same exponent, we may assume that the exponents $\mu+\nu/p$ in (2.20) are pairwise distinct.

We compute a formula for the inverse matrix $(A^\delta)^{-1}$ by applying Cramer's rule. Hence,

$$(A^\delta)^{-1}_{ij} = \frac{[A^\delta]_{ij}}{\det(A^\delta)} = \frac{\sum_k a_k^{ij}\cdot\delta^{\lambda_k}}{\sum_\ell b_\ell\cdot\delta^{\gamma_\ell}} \quad (1\le i,j\le J), \tag{2.21}$$

where $[A^\delta]_{ij}$ denotes the (i,j)-cofactor of the matrix A^δ. The number of terms in the sums in the numerator and denominator of (2.21) is bounded by C.

We compute the numbers a_k^{ij} and b_ℓ in (2.21) by evaluating a polynomial function of the coefficients $c_{\mu\nu}^{ij}$ arising in the entries of the matrix (A_{ij}^δ) in (2.20). The numbers $c_{\mu\nu}^{ij}$ are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, so we can compute a_k^{ij} and b_ℓ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

The exponents λ_k and γ_ℓ in (2.21) have the form $\mu + \nu/p$, where $\mu, \nu \in \mathbb{Z}$ are bounded in magnitude by a universal constant C.

We compute an expression for $\eta_{\min}(\delta) = \min_w \widetilde{M}^\delta(w)$ as follows. Note that the quadratic function $\widetilde{M}^\delta(w)$ in (2.18) achieves its minimum at $w^\delta := (A^\delta)^{-1} b^\delta$. Thus,

$$\eta_{\min}(\delta) = \widetilde{M}^\delta(w^\delta) = -\langle b^\delta, (A^\delta)^{-1} b^\delta\rangle + m^\delta = \sum_{i,j=1}^{J} b_i^\delta \cdot (A^\delta)_{ij}^{-1} \cdot b_j^\delta + m^\delta.$$

Inserting the expressions for the entries of $(A^\delta)^{-1}$, b^δ, and the expression for m^δ, we compute (see below) a rational expression

$$\eta_{\min}(\delta) = \frac{\sum_k a_k \cdot \delta^{\lambda_k}}{\sum_\ell b_\ell \cdot \delta^{\gamma_\ell}} = \frac{N(\delta)}{D(\delta)}. \tag{2.22}$$

The number of terms in the sums in the numerator and denominator of (2.22) is bounded by C.

The denominator $D(\delta) = \sum_\ell b_\ell \cdot \delta^{\gamma_\ell}$ in (2.22) is the same as the common denominator in the expression for $(A^\delta)_{ij}^{-1}$ in (2.21), namely $\det(A^\delta)$. From (2.19) we have $\det(A^\delta) \geq \Delta_g^C$, hence

$$\sum_\ell b_\ell \cdot \delta^{\gamma_\ell} \geq \Delta_g^C. \tag{2.23}$$

The exponents γ_ℓ and λ_k in (2.22) have the form $\mu + \nu/p$, where $\mu, \nu \in \mathbb{Z}$ are bounded in magnitude by a universal constant. We can assume that the γ_ℓ are distinct, as are the λ_k. (We combine all the monomials in the numerator or denominator that have the same exponent.)

The numbers a_k in the numerator in (2.22) are defined by evaluating a polynomial function on the coefficients a_k^{ij}, b_ℓ in $(A^\delta)_{ij}^{-1}$ (see (2.21)), and the coefficients $c_{\mu\nu}^j$ and $c_{\mu\nu}$ in b_j^δ and m^δ (see (2.20)). Thus, we can compute a_k with parameters $(\Delta_g^{C_1}, \Delta_g^{-C_1}\Delta_\epsilon)$ for a large enough universal constant C_1.

As explained before, the numbers b_ℓ are given with parameters $(\Delta_g^{C_1}, \Delta_g^{-C_1}\Delta_\epsilon)$.

The exponents λ_k in (2.22) are pairwise distinct and have the form $\mu + \nu/p$ for integers $\mu, \nu \in \mathbb{Z}$ with $|\mu|, |\nu| \leq C$. The same is true of the exponents γ_ℓ. Hence,

$$|\lambda_k - \lambda_{k'}| \geq c_0 \quad \text{and} \quad |\gamma_\ell - \gamma_{\ell'}| \geq c_0 \tag{2.24}$$

for all $k \neq k'$ and $\ell \neq \ell'$.[1]

[1]The constant c_0 here depends only on m,n,p, but it may depend sensitively on the approximation of $1/p$ by rationals with low denominators.

We now perform a crucial rounding step.

We introduce a parameter $\Delta = 2^{-S_1}$ of the form $\Delta = \Delta_g^{C_2}$, for $C_2 \in \mathbb{N}$ that is assumed to be greater than a large enough universal constant. We will later fix C_2 to be a universal constant, but not yet. We assume $\Delta_\epsilon \le \Delta_g^{C_1+C_2}$, hence

$$\Delta_g^{-C_1}\Delta_\epsilon \le \Delta. \tag{2.25}$$

The numbers a_k and b_ℓ are given with parameters $(\Delta_g^{C_1}, \Delta_g^{-C_1}\Delta_\epsilon)$, so we can compute S_1-bit machine numbers $\widetilde{a}_k$ and $\widetilde{b}_\ell$ with

$$\begin{cases} |a_k - \widetilde{a}_k| \le \Delta, & |b_\ell - \widetilde{b}_\ell| \le \Delta, \\ |\widetilde{a}_k| \le \Delta_g^{-C}, & |\widetilde{b}_\ell| \le \Delta_g^{-C}. \end{cases} \tag{2.26}$$

We set

$$\widetilde{\eta}_{\min}(\delta) = \frac{\sum_k \widetilde{a}_k\,\delta^{\lambda_k}}{\sum_\ell \widetilde{b}_\ell\,\delta^{\gamma_\ell}} = \frac{\widetilde{N}(\delta)}{\widetilde{D}(\delta)}. \tag{2.27}$$

We use (2.26) and the fact that λ_k and γ_ℓ are bounded by C to estimate the difference between $\eta_{\min}(\delta)$ and $\widetilde{\eta}_{\min}(\delta)$. For $\delta \in [\Delta_g, \Delta_g^{-1}]$, we have

$$\left|a_k\,\delta^{\lambda_k} - \widetilde{a}_k\,\delta^{\lambda_k}\right| \le \Delta\,\Delta_g^{-C}.$$

Hence,

$$\left|N(\delta) - \widetilde{N}(\delta)\right| = \left|\sum_k a_k\,\delta^{\lambda_k} - \sum_k \widetilde{a}_k\,\delta^{\lambda_k}\right| \le C\,\Delta\,\Delta_g^{-C} \le \Delta\,\Delta_g^{-C'}.$$

Moreover,

$$|N(\delta)| = \left|\sum_k a_k\,\delta^{\lambda_k}\right| \le \sum_k |a_k|\left|\delta^{\lambda_k}\right| \le C\,\Delta_g^{-C}\cdot\Delta_g^{-C} \le \Delta_g^{-C'}.$$

Similarly,

$$\left|D(\delta) - \widetilde{D}(\delta)\right| = \left|\sum_\ell b_\ell\,\delta^{\gamma_\ell} - \sum_\ell \widetilde{b}_\ell\,\delta^{\gamma_\ell}\right| \le \Delta\,\Delta_g^{-C'}.$$

Moreover,

$$D(\delta) = \sum_\ell b_\ell\,\delta^{\gamma_\ell} \overset{(2.23)}{\ge} \Delta_g^{C}, \quad \text{and}$$

$$\left|\widetilde{D}(\delta)\right| \ge |D(\delta)| - \left|D(\delta) - \widetilde{D}(\delta)\right| \ge \Delta_g^{C} - \Delta\,\Delta_g^{-C'} \ge \frac{1}{2}\Delta_g^{C},$$

since $\Delta\,\Delta_g^{-C'} = \Delta_g^{C_2-C'} \le \frac{1}{2}\Delta_g^{C}$, for large enough C_2.

Using the previous estimates, we have

$$|\eta_{\min}(\delta) - \widetilde{\eta}_{\min}(\delta)| = \left|\frac{N(\delta)}{D(\delta)} - \frac{\widetilde{N}(\delta)}{\widetilde{D}(\delta)}\right| \le \left|\frac{N(\delta)\cdot(D(\delta)-\widetilde{D}(\delta))}{D(\delta)\cdot\widetilde{D}(\delta)}\right| + \left|\frac{N(\delta)-\widetilde{N}(\delta)}{\widetilde{D}(\delta)}\right|$$
$$\le \frac{\Delta_g^{-C'}\cdot\Delta\Delta_g^{-C'}}{\Delta_g^{C}\cdot\frac{1}{2}\Delta_g^{C}} + \frac{\Delta\Delta_g^{-C'}}{\frac{1}{2}\Delta_g^{C}} \le \Delta\Delta_g^{-C''}.$$

Thus,

$$|\eta_{\min}(\delta) - \widetilde{\eta}_{\min}(\delta)| \le \Delta\Delta_g^{-C''} \le \widetilde{c}\,\Delta_g^{2m+2} \overset{(2.16)}{\le} \frac{1}{2}\cdot\eta_{\min}(\delta),$$

where we may choose C_2 large enough so that $\Delta\Delta_g^{-C''} = \Delta_g^{C_2-C''} \le \widetilde{c}\,\Delta_g^{2m+2}$, with $\widetilde{c}$ as in (2.16). We fix $C_2 \in \mathbb{N}$ to be a universal constant satisfying the previous bounds.

Therefore, thanks to (2.16) we have

$$\Delta_g^{C} \le \frac{1}{2}\cdot\eta_{\min}(\delta) \le \widetilde{\eta}_{\min}(\delta) \le 2\cdot\eta_{\min}(\delta) \quad (\Delta_g \le \delta \le \Delta_g^{-1}). \tag{2.28}$$

We assume that none of the coefficients in the expression (2.27) are equal to zero, for otherwise we could discard the vanishing terms. Since $\widetilde{a}_k$ and $\widetilde{b}_\ell$ are S_1-bit machine numbers and $\Delta = 2^{-S_1}$, this means that

$$|\widetilde{a}_k| \ge \Delta = \Delta_g^{C_2}, \quad |\widetilde{b}_\ell| \ge \Delta = \Delta_g^{C_2} \quad \text{for all } k, \ell. \tag{2.29}$$

We will now explain how to compute a piecewise monomial function $\eta_*(\delta)$ that differs from $\eta_{\min}(\delta)$ by at most a universal constant factor. The first, second, and third bullet points in FIT BASIS TO CONVEX BODY (finite-precision) will then be consequences of (2.14), (2.13), and (2.16), respectively. The guarantees in the fourth bullet point will follow by examining the construction below.

PROCEDURE: APPROXIMATE RATIONAL FUNCTION

We are given machine numbers $\widetilde{a}_k, \widetilde{b}_\ell$ satisfying

$$|\widetilde{a}_k|,\ |\widetilde{b}_\ell| \in \left[\Delta_g^{C}, \Delta_g^{-C}\right].$$

We are given numbers λ_k and γ_ℓ of the form $\mu + \nu/p$, for integers μ, ν with $|\mu|, |\nu| \le C$, such that

$$|\lambda_k - \lambda_{k'}| \ge c_0,\ |\gamma_\ell - \gamma_{\ell'}| \ge c_0 \quad \text{for all } k \ne k',\ \ell \ne \ell'.$$

Let

$$\widetilde{\eta}_{\min}(\delta) = \frac{\sum_k \widetilde{a}_k\,\delta^{\lambda_k}}{\sum_\ell \widetilde{b}_\ell\delta^{\gamma_\ell}}.$$

Assume that the number of summands in the numerator and denominator is bounded by a universal constant C. Suppose that there exists a function $\eta_{\min}(\delta)$ satisfying (2.13) and (2.28).

We compute machine intervals I_ℓ, machine numbers d_ℓ, and numbers ω_ℓ, such that $\left[\Delta_g, \Delta_g^{-1}\right]$ is the disjoint union of the I_ℓ, and such that the function $\eta_*\colon \left[\Delta_g, \Delta_g^{-1}\right] \to \mathbb{R}$, defined by

$$\eta_*(\delta) := d_\ell \cdot \delta^{\omega_\ell} \quad \text{for } \delta \in I_\ell,$$

satisfies

$$c \cdot \eta_*(\delta) \le \eta_{\min}(\delta) \le C \cdot \eta_*(\delta) \quad \text{for all } \delta \in \left[\Delta_g, \Delta_g^{-1}\right].$$

Here, c and C are universal constants.

The numbers ω_ℓ are of the form $\mu + \nu/p$ for integers μ, ν with $|\mu|, |\nu| \le C$.

The machine numbers d_ℓ are contained in the interval $[\Delta_g^C, \Delta_g^{-C}]$.

This computation requires work and storage at most C.

Explanation. We define

$$\mathcal{B} := \bigcup_{k \neq k'} I_{kk'}, \quad \text{where}$$

$$I_{kk'} := \left\{\delta \in \left[\Delta_g, \Delta_g^{-1}\right] : 5^{-1} \cdot |\widetilde{a}_k\, \delta^{\lambda_k}| \le |\widetilde{a}_{k'}\, \delta^{\lambda_{k'}}| \le 5 \cdot |\widetilde{a}_k\, \delta^{\lambda_k}|\right\},$$

and similarly

$$\mathcal{C} := \bigcup_{\ell \neq \ell'} J_{\ell\ell'}, \quad \text{where}$$

$$J_{\ell\ell'} := \left\{\delta \in \left[\Delta_g, \Delta_g^{-1}\right] : 5^{-1} \cdot |\widetilde{b}_\ell\, \delta^{\gamma_\ell}| \le |\widetilde{b}_{\ell'}\, \delta^{\gamma_{\ell'}}| \le 5 \cdot |\widetilde{b}_\ell\, \delta^{\gamma_\ell}|\right\}.$$

For any interval $I \subset \left[\Delta_g, \Delta_g^{-1}\right] \setminus (\mathcal{B} \cup \mathcal{C})$, we have

$$\text{(2.30)} \quad \begin{cases} \text{there exist unique } k = k(I) \in \{1, \dots, K\} \text{ and } \ell = \ell(I) \in \{1, \dots, L\} \text{ such that} \\ |\widetilde{a}_k\, \delta^{\lambda_k}| > 2 \sum_{k' \neq k} |\widetilde{a}_{k'} \delta^{\lambda_{k'}}| \ \text{ and } \ |\widetilde{b}_\ell \delta^{\gamma_\ell}| > 2 \sum_{\ell' \neq \ell} |\widetilde{b}_{\ell'} \delta^{\gamma_{\ell'}}| \text{ for all } \delta \in I. \end{cases}$$

Moreover, $\int_{\mathcal{B} \cup \mathcal{C}} dt/t \le 2A$ for a universal constant A. (The proof is by the same reasoning used in [3].)

To compute the endpoints of a nonempty interval $I_{kk'} = [h^-_{kk'}, h^+_{kk'}]$ $(k \neq k')$ we solve the equations

$$\delta^{\lambda_k - \lambda_{k'}} = 5^{\pm 1} \frac{|\widetilde{a}_{k'}|}{|\widetilde{a}_k|}.$$

The solutions $\delta = \delta_\pm$ are given by

$$\delta_\pm = \left(5^{\pm 1} \frac{|\widetilde{a}_{k'}|}{|\widetilde{a}_k|}\right)^{(\lambda_k - \lambda_{k'})^{-1}} \qquad \text{(for each choice of } \pm).$$

From the lower/upper bound on $|\widetilde{a}_k|$ by $\Delta_g^{\pm C}$, we see that we can compute $|\widetilde{a}_{k'}/\widetilde{a}_k|$ to precision $\Delta_g^{-C}\Delta_\epsilon$, and $|\widetilde{a}_{k'}/\widetilde{a}_k| \in [\Delta_g^{-C}, \Delta_g^C]$. Since $\left|(\lambda_k - \lambda_{k'})^{-1}\right| \le c_0^{-1} \le C$, we can compute δ_+ and δ_- to precision $\Delta_g^{-C}\Delta_\epsilon$, due to the numerical stability of exponentiation.

Now, note that

$$h^-_{kk'} = \min\{\delta_-, \delta_+, \Delta_g\}, \qquad h^+_{kk'} = \max\{\delta_-, \delta_+, \Delta_g^{-1}\}.$$

Both $h^-_{kk'}$ and $h^+_{kk'}$ can be computed with parameters $(\Delta_g, \Delta_g^{-C}\Delta_\epsilon)$. Thus, we can compute a machine interval $\widetilde{I}_{kk'} \subset [\Delta_g, \Delta_g^{-1}]$ with $I_{kk'} \subset \widetilde{I}_{kk'}$ and

$$\operatorname{dist}(I_{kk'}, \widetilde{I}_{kk'}) \le \Delta_g^{-C}\,\Delta_\epsilon, \tag{2.31}$$

where $\operatorname{dist}(\cdot,\cdot)$ is the Hausdorff distance. Due to the previous inclusion, we know that the union of the intervals $\widetilde{I}_{kk'}$ contains the set $\mathcal{B} = \cup_{k\neq k'} I_{kk'}$.

Similarly, we compute a machine interval $\widetilde{J}_{\ell\ell'} \subset [\Delta_g, \Delta_g^{-1}]$ with $J_{\ell\ell'} \subset \widetilde{J}_{\ell\ell'}$ and

$$\operatorname{dist}(J_{\ell\ell'}, \widetilde{J}_{\ell\ell'}) \le \Delta_g^{-C}\,\Delta_\epsilon. \tag{2.32}$$

Again, note that the union of the intervals $\widetilde{J}_{\ell\ell'}$ contains the set $\mathcal{C}$.

We next compute pairwise disjoint machine intervals $I^{\text{bad}}_\nu \subset \left[\Delta_g, \Delta_g^{-1}\right]$ such that

$$\bigcup_{\nu=1}^{\nu_{\max}} I^{\text{bad}}_\nu = \bigcup_{k\neq k'} \widetilde{I}_{kk'} \cup \bigcup_{\ell\neq\ell'} \widetilde{J}_{\ell\ell'}.$$

We form the intervals I^{bad}_ν by concatenating the intersecting intervals among $\widetilde{I}_{kk'}$ and $\widetilde{J}_{\ell\ell'}$. Note that the union of the I^{bad}_ν contains the set $\mathcal{B}\cup\mathcal{C}$.

Because the intervals below are contained in $\left[\Delta_g, \Delta_g^{-1}\right]$, for each ν we have

$$\begin{aligned}
\int_{I^{\text{bad}}_\nu} \frac{dt}{t} &\le \sum_{k,k'} \int_{\widetilde{I}_{kk'}} \frac{dt}{t} + \sum_{\ell,\ell'} \int_{\widetilde{J}_{\ell\ell'}} \frac{dt}{t} \\
&\overset{(2.31),(2.32)}{\le} \sum_{k,k'} \left[\int_{I_{kk'}} \frac{dt}{t} + \Delta_g^{-1}\cdot\Delta_g^{-C}\Delta_\epsilon \right] + \sum_{\ell,\ell'} \left[\int_{J_{\ell\ell'}} \frac{dt}{t} + \Delta_g^{-1}\Delta_g^{-C}\Delta_\epsilon \right] \\
&\le \int_{\mathcal{B}\cup\mathcal{C}} \frac{dt}{t} + \Delta_g^{-C'}\Delta_\epsilon \le 3A. \qquad (2.33)
\end{aligned}$$

Recall that $A \ge 1$ is a universal constant.

We compute pairwise disjoint machine intervals $I_\mu \subset \left[\Delta_g, \Delta_g^{-1}\right]$ such that

$$\bigcup_{\mu=1}^{\mu_{\max}} I_\mu = \left[\Delta_g, \Delta_g^{-1}\right] \setminus \bigcup_{\nu=1}^{\nu_{\max}} I^{\text{bad}}_\nu.$$

Thus, since the union of the I^{bad}_ν contains $\mathcal{B}\cup\mathcal{C}$, we have $I_\mu \subset \left[\Delta_g, \Delta_g^{-1}\right] \setminus (\mathcal{B}\cup\mathcal{C})$ for each μ. By (2.30), there exist $k = k(\mu) \in \{1,\dots,K\}$ and $\ell = \ell(\mu) \in \{1,\dots,L\}$ such that

$$|\widetilde{a}_k\,\delta^{\lambda_k}| > 2\sum_{k'\neq k} |\widetilde{a}_{k'}\delta^{\lambda_{k'}}| \quad\text{and}\quad |\widetilde{b}_\ell\delta^{\gamma_\ell}| > 2\sum_{\ell'\neq\ell} |\widetilde{b}_{\ell'}\delta^{\gamma_{\ell'}}| \quad \text{for all } \delta \in I_\mu.$$

We compute $k = k(\mu)$ and $\ell = \ell(\mu)$, for each μ, by searching over all k, ℓ to determine the maximal value of $\left|\widetilde{a}_k\,\delta_*^{\lambda_\ell}\right|$ and $\left|\widetilde{b}_\ell\,\delta_*^{\gamma_\ell}\right|$ for any fixed $\delta_* \in I_\mu$. Then, by

the definition of $\widetilde{\eta}_{\min}(\delta)$ we see that

$$c \cdot \widetilde{\eta}_{\min}(\delta) \le \frac{\widetilde{a}_k \, \delta^{\lambda_k}}{\widetilde{b}_\ell \, \delta^{\gamma_\ell}} \le C \cdot \widetilde{\eta}_{\min}(\delta) \quad \text{for all } \delta \in I_\mu. \tag{2.34}$$

According to (2.28), we also have $c \cdot \widetilde{\eta}_{\min}(\delta) \ge c \cdot \Delta_g^C$.

We compute machine numbers d_μ such that $\left| d_\mu - \widetilde{a}_k/\widetilde{b}_\ell \right| \le \Delta_g^{-C} \Delta_\epsilon$, and numbers $\omega_\mu = \lambda_k - \gamma_\ell$, where $k = k(\mu)$ and $\ell = \ell(\mu)$. We claim that

$$c \cdot \eta_{\min}(\delta) \le d_\mu \cdot \delta^{\omega_\mu} \le C \cdot \eta_{\min}(\delta) \quad \text{for all } \delta \in I_\mu.$$

Indeed, $d_\mu \cdot \delta^{\omega_\mu}$ differs from $\widetilde{a}_k \delta^{\lambda_k}/(\widetilde{b}_\ell \delta^{\gamma_\ell})$ by at most an additive error of $\Delta_g^{-C}\Delta_\epsilon$, since $\Delta_g \le \delta \le \Delta_g^{-1}$ and $|\omega_\mu| \le C$. This additive error is bounded by $\frac{1}{2} c \cdot \Delta_g^C \le c \cdot \widetilde{\eta}_{\min}(\delta)$ since, by assumption, $\Delta_\epsilon \le \frac{1}{2} c \Delta_g^{2C}$. Hence, (2.34) implies the above claim.

We compute a machine number δ_ν in each interval I_ν^{bad}. We know that $e^{-3A} \le \delta/\delta_\nu \le e^{3A}$ for all $\delta \in I_\nu^{\text{bad}}$, due to (2.33). Hence, (2.13) implies that

$$c \cdot \eta_{\min}(\delta) \le \eta_{\min}(\delta_\nu) \le C \cdot \eta_{\min}(\delta) \quad \text{for all } \delta \in I_\nu^{\text{bad}}. \tag{2.35}$$

We then compute a machine number Γ_ν such that $|\Gamma_\nu - \widetilde{\eta}_{\min}(\delta_\nu)| \le \Delta_g^{-C}\Delta_\epsilon$. Thus, from (2.28) and (2.35) we conclude that

$$c' \cdot \eta_{\min}(\delta) \le \Gamma_\nu \le C' \cdot \eta_{\min}(\delta) \quad \text{for all } \delta \in I_\nu^{\text{bad}}.$$

We define $\eta_* : \left[\Delta_g, \Delta_g^{-1}\right] \to \mathbb{R}$ by

$$\eta_*(\delta) = \begin{cases} d_\mu \cdot \delta^{\omega_\mu} & \text{if } \delta \in I_\mu, \\ \Gamma_\nu & \text{if } \delta \in I_\nu^{\text{bad}}. \end{cases} \tag{2.36}$$

As shown above, we have $c \cdot \eta_*(\delta) \le \eta_{\min}(\delta) \le C \cdot \eta_*(\delta)$ for $\delta \in \left[\Delta_g, \Delta_g^{-1}\right]$, hence we obtain the main estimate in the conclusion of the procedure APPROXIMATE RATIONAL FUNCTION (finite-precision). This completes the explanation. □

As mentioned before, by applying the procedure APPROXIMATE RATIONAL FUNCTION we compute a function $\eta_*(\delta)$ satisfying the conditions of the algorithm FIT BASIS TO CONVEX BODY (finite-precision). This completes the explanation. □

2.6. Compressing norms in finite-precision

We assume that $\Delta_{\min} \le \Delta_\epsilon \le \Delta_g \le \Delta_0 \le 1$ are as in the **Main assumptions** in Section 2.2. In particular, $\Delta_{\min} = 2^{-S}$ ($S = K_{\max}\overline{S}$) denotes the machine precision of our computer, and $\Delta_0 = 2^{-\overline{S}}$. We assume that $\Delta_\epsilon \le \Delta_g^C$ for a large enough universal constant C.

Let μ be a linear functional on $\mathbb{R}^D$ given in the form $\mu(v) = v \cdot w$, where $w \in \mathbb{R}^D$ is given as $w = (w_1, \dots, w_D)$. We define

$$\|\mu\| := \max_{1 \le i \le D} |w_i| .$$

We say that μ is specified with parameters $(\Delta_g, \Delta_\epsilon)$ if $\|\mu\| \leq \Delta_g^{-1}$ and if w_j is specified to precision Δ_ϵ for each $i = 1, \ldots, D$. This means that machine numbers w_i^{fin} are given with $\left|w_i - w_i^{\text{fin}}\right| \leq \Delta_\epsilon$ for each $1 \leq i \leq D$.

We assume that the following data are given.

- We fix a machine number $\Delta \in \left[\Delta_g, 1\right]$ of the form $\Delta = 2^{-K\overline{S}}$ for an integer $K \geq 1$.
- We specify linear functionals $\overline{\mu}_1, \ldots, \overline{\mu}_{\overline{L}}$ on $\mathbb{R}^D$ with parameters $(\Delta_g, \Delta_\epsilon)$. We assume that $\overline{L} \leq \Delta_g^{-1}$, and that $D \leq \widetilde{C}$ for a universal constant $\widetilde{C}$.
- We fix an $\overline{S}$-bit machine number $p > 1$.

We denote $|\nu| = \left(\sum_{i=1}^D |\nu_i|^p\right)^{1/p}$ for $\nu = (\nu_1, \ldots, \nu_D) \in \mathbb{R}^D$.

ALGORITHM: COMPRESS NORMS (FINITE-PRECISION VERSION)

Fix $1 < p < \infty$, and fix an integer $D \geq 1$ as above. Let $\overline{\mu}_1, \cdots, \overline{\mu}_{\overline{L}}$ be linear functionals on $\mathbb{R}^D$, and let $\Delta \in \left[\Delta_g, 1\right]$ be as above.

We compute linear functionals $\mu_1^*, \ldots, \mu_D^*$ on $\mathbb{R}^D$ such that

$$c \cdot \sum_{i=1}^{D} |\mu_i^*(\nu)|^p \leq \sum_{\ell=1}^{\overline{L}} |\overline{\mu}_\ell(\nu)|^p + \Delta^p |\nu|^p \leq C \cdot \sum_{i=1}^{D} |\mu_i^*(\nu)|^p \quad \text{for all } \nu \in \mathbb{R}^D. \tag{2.37}$$

The μ_i^* are represented as $\nu \mapsto \nu \cdot w_i^*$, where $w_i^* = (w_{i,1}^*, \ldots, w_{i,D}^*)$ and the $w_{i,k}^*$ are computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

This computation requires work and storage at most $C\overline{L}$.

Here, $c > 0$ and $C \geq 1$ are universal constants.

Explanation. We proceed by induction on D.

First consider the base case $D = 1$. The given functionals on $\mathbb{R}^1$ have the form $\overline{\mu}_\ell(\nu) = w_\ell \cdot \nu$ $(1 \leq \ell \leq \overline{L})$, where the numbers w_ℓ are specified with parameters $(\Delta_g, \Delta_\epsilon)$. We define $\gamma := \left(|w_1|^p + \cdots + \left|w_{\overline{L}}\right|^p + \Delta^p\right)^{1/p}$. Using Lemma 2, we compute a machine number $\widehat{\gamma}$ such that $\gamma/2 \leq \widehat{\gamma} \leq 2\gamma$. Define the functional $\mu_1^*(\nu) = \widehat{\gamma} \cdot \nu$ on $\mathbb{R}^1$. Then the estimate (2.37) holds with $c = 1/2$ and $C = 2$.

We now treat the induction step. Fix an integer $D \geq 2$. We assume by induction that the algorithm COMPRESS NORMS has been established when D is replaced by $D - 1$.

We write c, c', C, C', etc., to denote constants depending only on p and D.

We define the functionals

$$\omega_i(\nu) := \Delta \cdot \nu_i \quad \text{for } \nu = (\nu_1, \ldots, \nu_D) \in \mathbb{R}^D, \quad \text{for each } i = 1, \ldots, D. \tag{2.38}$$

Let $\{\mu_1, \ldots, \mu_L\}$ denote the collection $\{\overline{\mu}_1, \cdots \overline{\mu}_{\overline{L}}, \omega_1, \ldots, \omega_D\}$ of linear functionals on $\mathbb{R}^D$. Except for minor modifications, we mimic the computation in the infinite-precision version of COMPRESS NORMS (see Section 2.8 of [3]), using the collection $\{\mu_1, \ldots, \mu_L\}$ as input. We include the extra functionals ω_i in order to ensure that we never encounter division by a small number. This leads to the required numerical stability. We provide details of the computation below.

For each $1 \le i \le L$, we write

$$\begin{aligned}\mu_i(\nu_1,\dots,\nu_D) &:= \beta_i^* \cdot \nu_D + \mu_i(\nu_1,\dots,\nu_{D-1},0) \\ &= \epsilon_i \cdot \left[\beta_i \nu_D - \widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})\right],\end{aligned} \tag{2.39}$$

where $\beta_i = |\beta_i^*|$, $\epsilon_i = \operatorname{sgn}(\beta_i^*)$, and $\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1}) = -\epsilon_i \cdot \mu_i(\nu_1,\dots,\nu_{D-1},0)$. Here, $\operatorname{sgn}(\cdot)$ denotes the "signum" function: $\operatorname{sgn}(\alpha) = 1$ if $\alpha \ge 0$, and $\operatorname{sgn}(\alpha) = -1$ if $\alpha < 0$.

The numbers β_i^* in (2.39) are given with parameters $(\Delta_g, \Delta_\epsilon)$, since the functionals $\overline{\mu}_\ell$ are given with parameters $(\Delta_g, \Delta_\epsilon)$ and the ω_i are given exactly. Hence, we can compute β_i with parameters $(\Delta_g, 10\Delta_\epsilon)$ for each i. We cannot compute ϵ_i or $\widetilde{\mu}_i$ with any accuracy unless $|\beta_i^*| > \Delta_\epsilon$, but this remark will not cause much difficulty.

We set $\Delta_1 = \Delta_g^{C_0}$, for a universal constant $C_0 \in \mathbb{N}$ that will be determined later. Recall that β_i is specified to precision Δ_ϵ, and that $\Delta_\epsilon \le \frac{1}{4}\Delta_g^{C_0} = \frac{1}{4}\Delta_1$. Hence, we can compute a subset $I^{\text{fin}} \subset \{1,\dots,L\}$ such that

$$\beta_i \le 2\Delta_1 \text{ for } i \notin I^{\text{fin}}, \text{ and } \beta_i \ge \Delta_1 \text{ for } i \in I^{\text{fin}}. \tag{2.40}$$

(Just compare the machine approximation of each β_i to $\frac{3}{2}\Delta_1$.)

We compute $\epsilon_i = \operatorname{sgn}(\beta_i^*)$ exactly if $i \in I^{\text{fin}}$, since then we have $|\beta_i^*| = \beta_i \ge \Delta_1 \ge 2\Delta_\epsilon$. (We do not attempt to compute ϵ_i for $i \notin I^{\text{fin}}$.) Hence, we can compute the functional $\widetilde{\mu}_i$ with parameters $(\Delta_g, \Delta_\epsilon)$ for each $i \in I^{\text{fin}}$.

Alternatively, for each $i \notin I^{\text{fin}}$, we define the functional $\widetilde{\widetilde{\mu}}_i(\nu_1,\dots,\nu_{D-1}) = \mu_i(\nu_1,\dots,\nu_{D-1},0)$, which is given with parameters $(\Delta_g, \Delta_\epsilon)$. We have either $\widetilde{\widetilde{\mu}}_i = -\widetilde{\mu}_i$ or $\widetilde{\widetilde{\mu}}_i = \widetilde{\mu}_i$, though we do not guarantee which case occurs.

We have $\mu_{i_0} = \omega_D$ for some $i_0 \in \{1,\dots,L\}$. From (2.38), we see that $\beta_{i_0} = \Delta \ge \Delta_g > 2\Delta_1$, since β_{i_0} is the magnitude of the coefficient of ν_D in $\mu_{i_0} = \omega_D$. Hence, $i_0 \in I^{\text{fin}}$, thanks to (2.40). Therefore,

$$\mathbf{B} := \sum_{i \in I^{\text{fin}}} |\beta_i|^p \ge |\beta_{i_0}|^p = \Delta^p. \tag{2.41}$$

Each β_i in (2.41) is given with parameters $(\Delta_g, \Delta_\epsilon)$. Hence, we can compute $\mathbf{B}$ to precision $L \cdot \Delta_g^{-C}\Delta_\epsilon \le \Delta_g^{-C'}\Delta_\epsilon$, since $\#(I^{\text{fin}}) \le L \le \Delta_g^{-C}$ (note: the error invoked in computing each exponentiation $|\beta_i|^p$ is bounded by $\Delta_g^{-C}\Delta_\epsilon$). Clearly, also $\mathbf{B} \in \left[\Delta_g^C, \Delta_g^{-C}\right]$. Hence, for each $i \in I^{\text{fin}}$, we can compute $\text{Prob}(i) := |\beta_i|^p / \mathbf{B}$ with parameters $(1, \Delta_g^{-C}\Delta_\epsilon)$.

Recall that the coefficients of $\widetilde{\mu}_i : \mathbb{R}^{D-1} \to \mathbb{R}$ are bounded by Δ_g^{-1}, and $D \le C$. Therefore,

$$|\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p \le \Delta_g^{-C} |\nu|^p . \tag{2.42}$$

The list $\{\mu_1,\dots,\mu_L\}$ consists of the functionals $\overline{\mu}_\ell$ and ω_i (defined in (2.38)). Hence,

$$\sum_{\ell=1}^{\overline{L}} |\overline{\mu}_\ell(\nu_1,\dots,\nu_D)|^p + \Delta^p |\nu|^p = \sum_{i=1}^{L} |\mu_i(\nu_1,\dots,\nu_D)|^p \tag{2.43}$$

which differs by at most a factor of $\overline{C}$ from

$$\begin{aligned}(2.44)\qquad & \mathbf{B}\cdot|\nu_D-\overline{\mu}(\nu_1,\dots,\nu_{D-1})|^p \\ &+\Big\{\mathbf{B}\cdot\sum_{i\in I^{fin}}\mathrm{Prob}(i)\cdot\big|\overline{\mu}(\nu_1,\dots,\nu_{D-1})-\beta_i^{-1}\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})\big|^p \\ &\quad+\Big[\sum_{i\notin I^{fin}}|\beta_i\nu_D-\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p\Big]\Big\},\end{aligned}$$

where

$$\begin{aligned}(2.45)\qquad \overline{\mu}(\nu_1,\dots,\nu_{D-1}) &:= \sum_{i\in I^{fin}}\mathrm{Prob}(i)\cdot\big\{\beta_i^{-1}\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})\big\} \\ &= \mathbf{B}^{-1}\cdot\sum_{i\in I^{fin}}\beta_i^{p-1}\cdot\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1}).\end{aligned}$$

We prove these estimates by the same argument used in the estimations following equation (2.67) in [3]. (In contrast to the prior setting, we no longer guarantee here that $\beta_i=0$ for $i\notin I^{fin}$, which is why the third line in (2.44) contains an extra term of the form $\beta_i\nu_D$.)

Note that $\mathcal{I}^{fin}\neq\emptyset$, as we saw just before (2.41). Recall that $|\mathrm{Prob}(i)|\le 1$ for each i, and that $\beta_i\ge\Delta_1=\Delta_0^{C_0}$ for each $i\in I^{fin}$. Therefore, from (2.45) we see that

$$\|\overline{\mu}\|\le\#(I^{fin})\cdot\Delta_0^{-C}\cdot\max_i\{\|\widetilde{\mu}_i\|\}\le\Delta_0^{-C'}.$$

Moreover, we can compute $\overline{\mu}$ in (2.45) to precision $\Delta_0^{-C}\Delta_\epsilon$. Hence, we can compute $\overline{\mu}$ with parameters $(\Delta_0^C,\Delta_0^{-C}\Delta_\epsilon)$.

We next estimate the term inside the brackets in (2.44). Applying the estimate $\big||x+y|^p-|x|^p\big|\le p\cdot|y|\cdot(|x|+|y|)^{p-1}$, we have

$$\begin{aligned}&\Big|\sum_{i\notin I^{fin}}|\beta_i\nu_D-\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p-\sum_{i\notin I^{fin}}|\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p\Big| \\ &\qquad\le p\cdot\sum_{i\notin I^{fin}}|\beta_i\nu_D|\cdot\big\{|\beta_i\nu_D|+|\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|\big\}^{p-1} \\ &\qquad\le C\,L\,\Delta_0^{-C}\Delta_1\,|\nu|^p\le\Delta_0^{C_0-C'}\,|\nu|^p.\end{aligned}$$

The constant C' is independent of C_0. Here, we use estimate (2.42), that $|\beta_i|\le 2\Delta_1$ for $i\notin I^{fin}$ (see (2.40)), and that the number of relevant i is bounded by $L\le\Delta_0^{-C}$. Hence,

$$\mathfrak{S}-\Delta_0^{C_0-C'}\,|\nu|^p\le\big[\text{bracketed expression in (2.44)}\big]\le\mathfrak{S}+\Delta_0^{C_0-C'}\,|\nu|^p,$$
$$\text{where }\mathfrak{S}:=\sum_{i\notin I^{fin}}|\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p.$$

We now fix the constant C_0 used to define $\Delta_1=\Delta_0^{C_0}$. We take C_0 much larger than C' above, so that the junk term $\Delta_0^{C_0-C'}\,|\nu|^p$ is bounded by $\frac{1}{10}(\overline{C})^{-1}\Delta_0^p\,|\nu|^p\le$

$\frac{1}{10}(\overline{C})^{-1}\Delta^p\,|\nu|^p$. Hence, we can replace the expression inside square brackets in (2.44) with $\mathfrak{S}$, and we can absorb the junk term $\Delta_g^{C_0-C'}\,|\nu|^p$ into the junk term $\Delta^p\,|\nu|^p$ in (2.43). Consequently, $\sum_{\ell=1}^{\overline{L}}|\overline{\mu}_\ell(\nu_1,\dots,\nu_D)|^p+\Delta^p\,|\nu|^p$ differs by at most a factor of C'' from

$$\begin{aligned}&\mathbf{B}\cdot|\nu_D-\overline{\mu}(\nu_1,\dots,\nu_{D-1})|^p\\&+\Bigg\{\mathbf{B}\cdot\sum_{i\in I^{fin}}\mathrm{Prob}(i)\cdot\left|\overline{\mu}(\nu_1,\dots,\nu_{D-1})-\beta_i^{-1}\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})\right|^p\\&\quad+\Big[\sum_{i\notin I^{fin}}|\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p\Big]\Bigg\}\end{aligned}$$

We add $\Delta^p\,|(\nu_1,\dots,\nu_{D-1})|^p$ to both expressions in the previous sentence. Note that $\Delta^p\,|(\nu_1,\dots,\nu_{D-1})|^p+\Delta^p\,|\nu|^p$ differs by at most a factor of 2 from $\Delta^p\,|\nu|^p$. Therefore, $\sum_{\ell=1}^{\overline{L}}|\overline{\mu}_\ell(\nu_1,\dots,\nu_D)|^p+\Delta^p\,|\nu|^p$ differs by at most a factor of C''' from

$$\begin{aligned}&\mathbf{B}\cdot|\nu_D-\overline{\mu}(\nu_1,\dots,\nu_{D-1})|^p\\&\quad+\Bigg\{\sum_{i\in I^{fin}}|\beta_i\overline{\mu}(\nu_1,\dots,\nu_{D-1})-\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1})|^p\\&\qquad+\sum_{i\notin I^{fin}}\left|\widetilde{\widetilde{\mu}}_i(\nu_1,\dots,\nu_{D-1})\right|^p+\Delta^p\,|(\nu_1,\dots,\nu_{D-1})|^p\Bigg\}\end{aligned}\tag{2.46}$$

(Recall that $\mathrm{Prob}(i)=|\beta_i|^p/\mathbf{B}$ and that $\widetilde{\mu}_i=\pm\widetilde{\widetilde{\mu}}_i$.) We consider the functionals arising inside the curly brackets above, namely

$$\widehat{\mu}_i(\nu_1,\dots,\nu_{D-1}):=\begin{cases}\beta_i\overline{\mu}(\nu_1,\dots,\nu_{D-1})-\widetilde{\mu}_i(\nu_1,\dots,\nu_{D-1}) & \text{if } i\in I^{fin},\\ \widetilde{\widetilde{\mu}}_i(\nu_1,\dots,\nu_{D-1}) & \text{if } i\notin I^{fin}.\end{cases}$$

Note that $\|\widehat{\mu}_i\|\le\Delta_g^{-C}$, since the same upper bound holds for $\overline{\mu}$, $\widetilde{\widetilde{\mu}}_i=\pm\widetilde{\mu}_i$, and β_i. Moreover, each $\widehat{\mu}_i$ can be computed to precision $\Delta_g^{-C}\Delta_\epsilon$. Hence, we can compute $\widehat{\mu}_i$ $(1\le i\le L)$ with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$.

The functionals $\widehat{\mu}_i$ are given with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$ and $\Delta_g^C\le\Delta_g\le\Delta\le 1$. By the induction hypothesis, we can compute functionals $\mu_1^*,\dots,\mu_{D-1}^*:\mathbb{R}^{D-1}\to\mathbb{R}$ such that

$$\sum_{i=1}^{D-1}|\mu_i^*(\nu_1,\dots,\nu_{D-1})|^p$$

differs by at most a factor of C from the expression in curly brackets in (2.46). The $\mu_1^*,\dots,\mu_{D-1}^*$ are specified with parameters $(\Delta_g^{C'},\Delta_g^{-C'}\Delta_\epsilon)$.

We define

$$\mu_D^*(\nu_1,\dots,\nu_D):=\mathbf{B}^{1/p}\cdot[\nu_D-\overline{\mu}(\nu_1,\dots,\nu_{D-1})]\,.$$

We can compute μ_D^* with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$, since the same is true of $\mathbf{B}$ and $\overline{\mu}$, and since $\mathbf{B}\ge\Delta^p\ge\Delta_g^p$ (see (2.41)).

Thus, from (2.46), we see that

$$c\cdot\Big[\sum_{\ell=1}^{\overline{L}}|\overline{\mu}_\ell(\nu_1,\dots,\nu_D)|^p+\Delta^p\,|\nu|^p\Big]\le|\mu_D^*(\nu_1,\dots,\nu_D)|^p+\sum_{i=1}^{D-1}|\mu_i^*(\nu_1,\dots,\nu_{D-1})|^p$$
$$\le C\cdot\Big[\sum_{\ell=1}^{\overline{L}}|\overline{\mu}_\ell(\nu_1,\dots,\nu_D)|^p+\Delta^p\,|\nu|^p\Big].$$

This completes the explanation of the finite-precision version of COMPRESS NORMS. □

2.7. Algorithm: Optimize via matrix

We define $\Delta_{\min}\le\Delta_\epsilon\le\Delta_g\le\Delta_0\le 1$ as in the **Main assumptions** in Section 2.2. In particular, $\Delta_{\min}=2^{-S}$ ($S=K_{\max}\overline{S}$) denotes the machine precision of our computer, and $\Delta_0=2^{-\overline{S}}$. We assume that $\Delta_\epsilon\le\Delta_g^C$ for a large enough universal constant C.

We are given the following data:

- We fix a machine number $\Delta\in[\Delta_g,1]$ of the form $\Delta=2^{-K\overline{S}}$ for an integer $K\ge 1$.
- We are given a matrix $A=(a_{\ell j})_{1\le\ell\le L,\,1\le j\le J}$. The numbers $a_{\ell j}$ are specified with parameters $(\Delta_g,\Delta_\epsilon)$. We have $1\le L\le\Delta_g^{-1}$ and $1\le J\le C$ for a universal constant C.
- We fix an $\overline{S}$-bit machine number $p>1$.

ALGORITHM: OPTIMIZE VIA MATRIX (FINITE-PRECISION)

Given $1<p<\infty$, given Δ, and given a matrix $A=(a_{\ell j})_{1\le\ell\le L,\,1\le j\le J}$ as above, we compute a matrix $B=(b_{j\ell})_{1\le j\le J,\,1\le\ell\le L}$. We guarantee that the following conditions hold.

Let $y_1,\dots,y_L$ be real numbers, and set $x_j^*=\sum_{\ell=1}^L b_{j\ell}y_\ell$ for each $j=1,\dots,J$. Then

$$\sum_{\ell=1}^L|y_\ell+\sum_{j=1}^J a_{\ell j}\,x_j^*|^p\le C_1\cdot\Big[\sum_{\ell=1}^L|y_\ell+\sum_{j=1}^J a_{\ell j}\,x_j|^p+\Delta^p\sum_{j=1}^J|x_j|^p\Big]$$

for any real numbers $x_1,\dots,x_J$.

The numbers $b_{j\ell}$ are computed with parameters $(\Delta_g^{C_1},\Delta_g^{-C_1}\Delta_\epsilon)$.

The algorithm requires work and storage at most $C_1\cdot L$.

Here, C_1 is a universal constant.

Explanation. We write c, C, C', etc., to denote universal constants.

We proceed by induction on J. We first handle the case $J=1$.

Assume that an $L\times 1$ matrix $(a_\ell)_{1\le\ell\le L}$ is given, with each number a_ℓ specified with parameters $(\Delta_g,\Delta_\epsilon)$.

Let $y_1,\dots,y_L$ be given real numbers.

We define $y_0=0$ and $a_0=\Delta$.

We compute an index set $\mathcal{L} \subset \{0, \dots, L\}$ such that $|a_\ell| \geq \Delta_g^{10}$ for $\ell \in \mathcal{L}$, and $|a_\ell| \leq 2\Delta_g^{10}$ for $\ell \in \{0, \dots, L\} \setminus \mathcal{L}$. To do so, we compare the machine approximation of each $|a_\ell|$ to the machine number $\frac{3}{2}\Delta_g^{10}$.

Note that $a_0 = \Delta \geq \Delta_g > 2\Delta_g^{10}$, which implies that $0 \in \mathcal{L}$. In particular, $\mathcal{L} \neq \emptyset$.

If $|a_\ell| \leq 2\Delta_g^{10}$ then the quantities $|y_\ell| + 2\Delta_g^{10} \cdot |x|$ and $|y_\ell + a_\ell x| + 2\Delta_g^{10} \cdot |x|$ differ by at most a factor of 2, thanks to the triangle inequality. Thus,

$$|y_\ell + a_\ell x|^p + \Delta_g^{10p} \cdot |x|^p \sim |y_\ell|^p + \Delta_g^{10p} \cdot |x|^p \quad \text{for } \ell \in \{0, \dots, L\} \setminus \mathcal{L},$$

where $A \sim B$ indicates that $c \cdot A \leq B \leq C \cdot A$ for some universal constants $c > 0$ and $C \geq 1$. Therefore, we have

$$\sum_{\ell=0}^{L} |y_\ell + a_\ell x|^p + \mathcal{E}(x) \sim \sum_{\ell \in \mathcal{L}} |y_\ell + a_\ell x|^p + \sum_{\ell \in \{0,\dots,L\} \setminus \mathcal{L}} |y_\ell|^p + \mathcal{E}(x), \tag{2.47}$$

$$\text{where } \mathcal{E}(x) = \#(\{0, \dots, L\} \setminus \mathcal{L}) \cdot \Delta_g^{10p} \cdot |x|^p .$$

Since $L \leq \Delta_g^{-1}$, it follows that $\mathcal{E}(x) \leq \Delta_g^p \cdot |x|^p \leq \Delta^p \cdot |x|^p = |y_0 + a_0 x|^p$. Therefore, because $|y_0 + a_0 x|^p$ is a summand on both sides of (2.47), we can discard the error term $\mathcal{E}(x)$ and obtain

$$\sum_{\ell=0}^{L} |y_\ell + a_\ell x|^p \sim \sum_{\ell \in \mathcal{L}} |y_\ell + a_\ell x|^p + \sum_{\ell \in \{0,\dots,L\} \setminus \mathcal{L}} |y_\ell|^p .$$

We write this estimate in the form

$$\sum_{\ell=0}^{L} |y_\ell + a_\ell x|^p \sim \sum_{\ell \in \mathcal{L}} |\overline{y}_\ell + x|^p \cdot |a_\ell|^p + \sum_{\ell \in \{0,\dots,L\} \setminus \mathcal{L}} |y_\ell|^p , \tag{2.48}$$

$$\text{where we define } \overline{y}_\ell := \frac{y_\ell}{a_\ell}.$$

Now, we want to minimize the expression $\mathcal{T}(x) = \sum_{\ell \in \mathcal{L}} |\overline{y}_\ell + x|^p \cdot |a_\ell|^p$ up to a universal constant factor. We define

$$x^* := -\sum_{\ell \in \mathcal{L}} \overline{y}_\ell \cdot \mathrm{Prob}(\ell), \quad \text{where} \quad \mathrm{Prob}(\ell) := \Big(\sum_{\ell' \in \mathcal{L}} |a_{\ell'}|^p \Big)^{-1} \cdot |a_\ell|^p \quad \text{for } \ell \in \mathcal{L}.$$

Recall that $\mathcal{L} \neq \emptyset$. Hence, $\mathrm{Prob}(\ell)$ is a well-defined probability measure on $\mathcal{L}$. From equation (2.65) in [3], we conclude that $\mathcal{T}(x_*) \leq C \cdot \mathcal{T}(x)$ for all $x \in \mathbb{R}$. Therefore, we have

$$\sum_{\ell=0}^{L} |y_\ell + a_\ell x^*|^p \leq C' \cdot \sum_{\ell=0}^{L} |y_\ell + a_\ell x|^p \quad \text{for any } x \in \mathbb{R}.$$

Because $y_0 = 0$ and $a_0 = \Delta$, this implies that

$$\sum_{\ell=1}^{L} |y_\ell + a_\ell x^*|^p \leq C' \cdot \Big[\sum_{\ell=1}^{L} |y_\ell + a_\ell x|^p + \Delta^p |x|^p \Big],$$

as desired in the case $J = 1$ of our algorithm. Note that

$$\begin{cases} x^* = -\sum_{\ell \in \mathcal{L}} \overline{y}_\ell \cdot \mathrm{Prob}(\ell) = \sum_{\ell \in \mathcal{L} \setminus \{0\}} y_\ell \cdot b_\ell, \quad \text{where} \\ b_\ell = -\Big(\sum_{\ell' \in \mathcal{L}} |a_{\ell'}|^p \Big)^{-1} \cdot |a_\ell|^p \cdot a_\ell^{-1} \quad \text{for } \ell \in \mathcal{L} \setminus \{0\}. \end{cases}$$

It is safe to discard the $\ell = 0$ term in the sum, because by definition $y_0 = \overline{y}_0 = 0$. Note that $|a_\ell|$ and $|a_{\ell'}|$, for $\ell, \ell' \in \mathcal{L}$, belong to the interval $[\Delta_g^{10}, \Delta_g^{-1}]$. Therefore, we can compute $|a_{\ell'}|^p$ and $|a_\ell|^p$ to precision $\Delta_g^{-C}\Delta_\epsilon$; moreover, we can compute the expression $(\cdots)^{-1}$ –in the formula for b_ℓ– with precision $\Delta_g^{-C}\Delta_\epsilon$. Thus, we can compute the coefficients b_ℓ, for each $\ell \in \mathcal{L} \setminus \{0\}$, with precision $\Delta_g^{-C}\Delta_\epsilon$. Furthermore, note that each $|b_\ell|$ is bounded by Δ_g^{-C} for a universal constant $C \geq 1$.

All the remaining coefficients b_ℓ, for $\ell \in \{1, \ldots, L\} \setminus \mathcal{L}$, are defined to be 0. Thus, $x^* = \sum_{\ell=1}^{L} y_\ell \cdot b_\ell$, and b_ℓ can be computed with the desired parameters. Thus, we have established the case $J = 1$ of our algorithm.

For the general case, we use induction on J.

Let $J \geq 2$, and let $1 < p < \infty$ and assume that we are given an $L \times J$ matrix $A = (a_{\ell j})_{1 \leq \ell \leq L,\, 1 \leq j \leq J}$. We assume that the numbers $a_{\ell j}$ are specified with parameters $(\Delta_g, \Delta_\epsilon)$.

Let real numbers $y_1, \ldots, y_L$ be given. We have

$$\sum_{\ell=1}^{L} \Big| y_\ell + \sum_{j=1}^{J} a_{\ell j} x_j \Big|^p = \sum_{\ell=1}^{L} \Big| \widehat{y}_\ell + \sum_{j=1}^{J-1} a_{\ell j} x_j \Big|^p \quad ((x_1, \ldots, x_J) \in \mathbb{R}^J), \tag{2.49}$$

using new variables

$$\widehat{y}_\ell = y_\ell + a_{\ell J} \cdot x_J \quad \text{for } 1 \leq \ell \leq L. \tag{2.50}$$

By applying the algorithm OPTIMIZE VIA MATRIX recursively to $1 < p < \infty$ and the submatrix $(a_{\ell j})_{1 \leq \ell \leq L,\, 1 \leq j \leq J-1}$, we compute a matrix $(\widehat{b}_{j\ell})_{1 \leq j \leq J-1,\, 1 \leq \ell \leq L}$ such that the following holds.

- We compute the numbers $\widehat{b}_{j\ell}$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ for a universal constant C.
- Let $\widehat{y}_1, \ldots, \widehat{y}_L$ be given, and set

$$\widehat{x}_j = \sum_{\ell=1}^{L} \widehat{b}_{j\ell}\, \widehat{y}_\ell \quad \text{for } 1 \leq j \leq J-1. \tag{2.51}$$

Then, for any real numbers $x_1, \ldots, x_{J-1}$, we have

$$\sum_{\ell=1}^{L} |\widehat{y}_\ell + \sum_{j=1}^{J-1} a_{\ell j}\, \widehat{x}_j|^p \leq C \cdot \Big[\sum_{\ell=1}^{L} |\widehat{y}_\ell + \sum_{j=1}^{J-1} a_{\ell j}\, x_j|^p + \Delta^p \sum_{j=1}^{J-1} |x_j|^p \Big]. \tag{2.52}$$

Using (2.49)–(2.52), we draw the following conclusion.

Let real numbers $y_1, \dots, y_L$ be given, and let $x_1, \dots, x_J$ be arbitrary. We define $\widehat{y}_1, \dots, \widehat{y}_L$ by (2.50), next define $\widehat{x}_1, \dots, \widehat{x}_{J-1}$ by (2.51), and finally set

$$\widehat{x}_J = x_J. \tag{2.53}$$

Then

$$\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \widehat{x}_j|^p \le C \cdot \Big[\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j} x_j|^p + \Delta^p \sum_{j=1}^{J-1} |x_j|^p \Big],$$

hence

$$\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \widehat{x}_j|^p + \Delta^p \cdot |\widehat{x}_J|^p \le C \cdot \Big[\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j} x_j|^p + \Delta^p \sum_{j=1}^{J} |x_j|^p \Big], \tag{2.54}$$

and moreover

$$\widehat{x}_j = \sum_{\ell=1}^{L} \widehat{b}_{j\ell} \cdot (y_\ell + a_{\ell J}\, \widehat{x}_J) \quad \text{for } j = 1, \dots, J-1. \tag{2.55}$$

Thus,

$$\widehat{x}_j = \sum_{\ell=1}^{L} \widehat{b}_{j\ell}\, y_\ell + g_j\, \widehat{x}_J, \quad \text{where} \tag{2.56}$$

$$g_j := \sum_{\ell=1}^{L} \widehat{b}_{j\ell}\, a_{\ell J} \quad \text{for } j = 1, \dots, J-1. \tag{2.57}$$

We compute the numbers g_j with parameters $(\Delta_g^{C'}, \Delta_g^{-C'}\Delta_\epsilon)$ using work at most CL. This is possible because $L \le \Delta_g^{-1}$ and because of parameters with which $\widehat{b}_{j\ell}$ and $a_{\ell j}$ are specified. In the above discussion, the numbers $x_1, \dots, x_J$ are arbitrary, the numbers $\widehat{x}_1, \dots, \widehat{x}_{J-1}$ are defined from $\widehat{x}_J$ by (2.55), and $\widehat{x}_J = x_J$.

Next, note that

$$\begin{aligned} y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \widehat{x}_j &= y_\ell + \sum_{j=1}^{J-1} a_{\ell j} \Big[\sum_{\ell'=1}^{L} \widehat{b}_{j\ell'} y_{\ell'} + g_j \widehat{x}_J \Big] + a_{\ell J} \widehat{x}_J \\ &= \Big\{ y_\ell + \sum_{j=1}^{J-1} a_{\ell j} \sum_{\ell'=1}^{L} \widehat{b}_{j\ell'} y_{\ell'} \Big\} + \Big\{ a_{\ell J} + \sum_{j=1}^{J-1} a_{\ell j} g_j \Big\} \widehat{x}_J =: y_\ell^{\text{ouch}} + h_\ell \cdot \widehat{x}_J. \end{aligned}$$

Here,

$$y_\ell^{\text{ouch}} = y_\ell + \sum_{j=1}^{J-1} a_{\ell j} \sum_{\ell'=1}^{L} \widehat{b}_{j\ell'}\, y_{\ell'}, \tag{2.58}$$

and

$$h_\ell = a_{\ell J} + \sum_{j=1}^{J-1} a_{\ell j}\, g_j \quad \text{for } \ell = 1, \dots, L. \tag{2.59}$$

Thus,

$$\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \widehat{x}_j|^p = \sum_{\ell=1}^{L} |y_\ell^{\text{ouch}} + h_\ell\, \widehat{x}_J|^p. \tag{2.60}$$

Here, (2.60) holds whenever $\widehat{x}_1, \dots, \widehat{x}_{J-1}$ are determined from $\widehat{x}_J$ via (2.56).

We compute the numbers h_ℓ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, using work at most CL.

Note that it is too expensive to compute y_ℓ^{ouch} for all ℓ $(1 \le \ell \le L)$; that computation would require $\sim L^2 J$ work. However, the y_ℓ^{ouch} defined above are independent of our choice of $\widehat{x}_J$.

Applying the known case $J = 1$ of our algorithm OPTIMIZE VIA MATRIX, we compute from the h_ℓ a vector of coefficients γ_ℓ $(1 \le \ell \le L)$, for which the following holds.

- We compute the numbers γ_ℓ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ for a universal constant C.
- Let

$$\check{x}_J = \sum_{\ell=1}^{L} \gamma_\ell\, y_\ell^{\text{ouch}}. \tag{2.61}$$

  Then

$$\sum_{\ell=1}^{L} |y_\ell^{\text{ouch}} + h_\ell\, \check{x}_J|^p \le C \cdot \Big[\sum_{\ell=1}^{L} |y_\ell^{\text{ouch}} + h_\ell\, \widehat{x}_J|^p + \Delta^p \cdot |\widehat{x}_J|^p \Big]$$

  for any real number $\widehat{x}_J$.

We thus learn the following.

Let $\check{x}_1, \dots, \check{x}_{J-1}$ be defined from $\check{x}_J$ as in (2.56), i.e.,

$$\check{x}_j := \sum_{\ell=1}^{L} \widehat{b}_{j\ell}\, y_\ell + g_j\, \check{x}_J \quad \text{for } j = 1, \dots, J-1. \tag{2.62}$$

Let $\widehat{x}_J$ be any real number, and let $\widehat{x}_1, \dots, \widehat{x}_{J-1}$ be determined from $\widehat{x}_J$ by (2.56). Then

$$\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \check{x}_j|^p \le C \cdot \Big[\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \widehat{x}_j|^p + \Delta^p \cdot |\widehat{x}_J|^p \Big] \tag{2.63}$$

(See (2.60).)

From (2.54) and (2.63), we see that

$$\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, \check{x}_j|^p \le C \cdot \Big[\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j}\, x_j|^p + \Delta^p \sum_{j=1}^{J} |x_j|^p \Big]. \tag{2.64}$$

Here, $\check{x}_1, \dots, \check{x}_J$ are computed from (2.61),(2.62); and $x_1, \dots, x_J$ are arbitrary.

We produce efficient formulas for the $\check{x}_j$. Putting (2.58) into (2.61), we find that

$$\check{x}_J = \sum_{\ell=1}^{L} \gamma_\ell \cdot \Big\{ y_\ell + \sum_{j=1}^{J-1} a_{\ell j} \sum_{\ell'=1}^{L} \widehat{b}_{j\ell'} y_{\ell'} \Big\} = \sum_{\ell=1}^{L} \gamma_\ell \cdot y_\ell + \sum_{\ell'=1}^{L} \sum_{j=1}^{J-1} \Big[\sum_{\ell=1}^{L} \gamma_\ell a_{\ell j} \Big] \widehat{b}_{j\ell'} y_{\ell'}$$

$$= \sum_{\ell=1}^{L} \Big\{ \gamma_\ell + \sum_{j=1}^{J-1} \Big[\sum_{\ell'=1}^{L} \gamma_{\ell'} \, a_{\ell' j} \Big] \widehat{b}_{j\ell} \Big\} \cdot y_\ell .$$

Therefore, setting

$$\Delta_j = \sum_{\ell=1}^{L} \gamma_\ell \, a_{\ell j} \quad \text{for } j = 1, \ldots, J-1 \tag{2.65}$$

and

$$b_{J\ell}^{\#\#} = \gamma_\ell + \sum_{j=1}^{J-1} \Delta_j \, \widehat{b}_{j\ell} \quad \text{for } \ell = 1, \ldots, L \tag{2.66}$$

we find that

$$\check{x}_J = \sum_{\ell=1}^{L} b_{J\ell}^{\#\#} \, y_\ell . \tag{2.67}$$

Substituting (2.67) into (2.62), we find that

$$\check{x}_j = \sum_{\ell=1}^{L} \Big\{ \widehat{b}_{j\ell} + g_j \, b_{J\ell}^{\#\#} \Big\} \, y_\ell \quad \text{for } j = 1, \ldots, J-1.$$

Thus, setting

$$b_{j\ell}^{\#\#} = \widehat{b}_{j\ell} + g_j \, b_{J\ell}^{\#\#} \quad \text{for } j = 1, \ldots, J-1, \quad \ell = 1, \ldots, L \tag{2.68}$$

we have

$$\check{x}_j = \sum_{\ell=1}^{L} b_{j\ell}^{\#\#} \, y_\ell \quad \text{for } j = 1, \ldots, J-1. \tag{2.69}$$

Recalling (2.67), we see that (2.69) holds for $j = 1, \ldots, J$. Thus, with $\check{x}_1, \ldots, \check{x}_J$ defined by (2.69), we have

$$\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j} \, \check{x}_j|^p \le C \cdot \Big[\sum_{\ell=1}^{L} |y_\ell + \sum_{j=1}^{J} a_{\ell j} x_j|^p + \Delta^p \sum_{j=1}^{J} |x_j|^p \Big]$$

for any real numbers $x_1, \ldots, x_J$. (See (2.64).)

So the matrix $B = (b_{j\ell}^{\#\#})_{1 \le j \le J, \, 1 \le \ell \le L}$ is as promised in our algorithm.

We make a few additional remarks on the computation of $(b_{j\ell}^{\#\#})_{1 \le j \le J, \, 1 \le \ell \le L}$.

- Recall the numbers γ_ℓ, $a_{\ell j}$, and $\widehat{b}_{j\ell}$ are given with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$. Also recall that $L \le \Delta_g^{-1}$.

- Thus, the numbers Δ_j $(1 \le j \le J-1)$ in (2.65) can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.
- Consequently, the numbers $b_{j\ell}^{\#\#}$ $(1 \le \ell \le L)$ in (2.66) can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.
- Recall that the numbers g_j $(1 \le j \le J-1)$ in (2.57) can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.
- Therefore, the numbers $b_{j\ell}^{\#\#}$ $(1 \le j \le J-1,\ 1 \le \ell \le L)$ in (2.68) can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.
- Thus, the matrix $B = (b_{j\ell}^{\#\#})$ can be computed to the accuracy promised in the algorithm. □

2.8. Statement of main technical results

We next state a modified version of the main technical results for $\mathcal{A}$ (see Section 3 in [3]) that accounts for the rounding errors that may arise in our computation.

We define a norm $|P| := \big(\sum_{\alpha\in\mathcal{M}} |\partial^\alpha P(0)|^p\big)^{1/p}$ for $P \in \mathcal{P}$. Thus, $|P|$ denotes the ℓ^p-norm of the vector $(\partial^\alpha P(0))_{\alpha\in\mathcal{M}}$.

We fix an integer $\overline{S} \ge 1$.

We are given a finite set $E \subset \frac{1}{32}Q^\circ$, with $Q^\circ = [0,1)^n$. We assume that $N = \#(E) \ge 2$. We additionally assume that E consists of $\overline{S}$-bit machine points. Thus,

$$|x - x'| \ge \Delta_0 \quad \text{for distinct } x, x' \in E, \tag{2.70}$$

where $\Delta_0 := 2^{-\overline{S}}$. Hence,

$$\#(E) = N \le \Delta_0^{-n}. \tag{2.71}$$

For $\Delta_1, \Delta_2 \in (0,1]$, we write $\Delta_1 \ll \Delta_2$ to indicate that $\Delta_1 \le \Delta_2^C$ for a sufficiently large universal constant C.

We introduce constants $\Delta_\epsilon^\circ := 2^{-K_1\overline{S}}$, $\Delta_g^\circ := 2^{-K_2\overline{S}}$, and $\Delta_{\text{junk}}^\circ := 2^{-K_3\overline{S}}$ as in Theorem 1. Here, K_1, K_2, K_3 are positive integers, which are assumed to be sufficiently well-separated in the sense that $K_1 \ge C \cdot K_2 \ge C^2 \cdot K_3$ for a large enough universal constant C.

For each $\mathcal{A} \subset \mathcal{M}$, we will use parameters $\Delta_\epsilon(\mathcal{A}) = \Delta_0^{K_1(\mathcal{A})}$, $\Delta_g(\mathcal{A}) = \Delta_0^{K_2(\mathcal{A})}$, and $\Delta_{\text{junk}}(\mathcal{A}) = \Delta_0^{K_3(\mathcal{A})}$ for integer exponents $K_1(\mathcal{A}) \ge K_2(\mathcal{A}) \ge K_3(\mathcal{A}) \ge 1$. We assume the exponents are chosen so that

$$\begin{aligned} \Delta_\epsilon(\mathcal{M}) &\ll \cdots \ll \Delta_\epsilon(\emptyset) \ll \Delta_\epsilon^\circ \\ &\ll \Delta_g^\circ \ll \Delta_g(\emptyset) \ll \cdots \ll \Delta_g(\mathcal{M}) \\ &\ll \Delta_{\text{junk}}(\mathcal{M}) \ll \cdots \ll \Delta_{\text{junk}}(\emptyset) \ll \Delta_{\text{junk}}^\circ \\ &\ll \Delta_0. \end{aligned} \tag{2.72}$$

In particular,

$$\begin{cases} \Delta_\epsilon(\emptyset) \le (\Delta_\epsilon^\circ)^C \\ \Delta_g^\circ \le (\Delta_g(\emptyset))^C \\ \Delta_{\text{junk}}(\emptyset) \le (\Delta_{\text{junk}}^\circ)^C \end{cases} \tag{2.73}$$

and

$$\begin{cases} \Delta_\epsilon(\mathcal{A}^-) \le \Delta_\epsilon(\mathcal{A}^+)^C \\ \Delta_g(\mathcal{A}^+) \le \Delta_g(\mathcal{A}^-)^C \\ \Delta_{\text{junk}}(\mathcal{A}^-) \le \Delta_{\text{junk}}(\mathcal{A}^+)^C \\ \Delta_\epsilon(\emptyset) \le \Delta_g(\emptyset)^C \\ \Delta_g(\mathcal{M}) \le \Delta_{\text{junk}}(\mathcal{M})^C \\ \Delta_{\text{junk}}(\emptyset) \le \Delta_0^C \end{cases} \tag{2.74}$$

for any $\mathcal{A}^+ > \mathcal{A}^-$ and for a large enough universal constant C. We refer the reader to Section 2.6 in [3] for the definition of our order relation on sets of multiindices. The conditions in (2.72), (2.73), and (2.74) are clearly consistent with one another. We will use these conditions throughout the course of the proof.

We assume throughout the course of the proof that we can perform arithmetic operations on S-bit machine numbers to precision $\Delta_{\min} = 2^{-S}$, where $S = K_{\max}\overline{S}$. Here, the parameter $K_{\max} \in \mathbb{N}$ is larger than all the exponents $K_j(\mathcal{A})$ (for all $\mathcal{A} \subset \mathcal{M}$ and $j = 1, 2, 3$).

The main technical results for $\mathcal{A}$ are as in Section 3 of [3], with the following modifications.

- We define a dyadic decomposition $\mathrm{CZ}(\mathcal{A})$ of Q°. We continue to guarantee the properties **(CZ1)**–**(CZ5)**. We additionally guarantee that

$$\delta_Q \ge \frac{1}{32} \cdot \Delta_0 \quad \text{for all } Q \in \mathrm{CZ}(\mathcal{A}). \tag{2.75}$$

  Hence, each cube in $\mathrm{CZ}(\mathcal{A})$ has $\widetilde{S}$-bit machine points as corners, where $\widetilde{S} \le \overline{S} + 100$. Thus, we can store each cube in $\mathrm{CZ}(\mathcal{A})$ on our computer using at most C units of storage (for a universal constant C). However, we will not compute all the cubes in $\mathrm{CZ}(\mathcal{A})$ for this would require too much work.

- We let $\mathrm{CZ}_{\text{main}}(\mathcal{A})$ consist of all the cubes $Q \in \mathrm{CZ}(\mathcal{A})$ such that $\frac{65}{64}Q \cap E \ne \emptyset$. For each $Q \in \mathrm{CZ}_{\text{main}}(\mathcal{A})$, we will compute $\Omega(Q, \mathcal{A})$, $\Xi(Q, \mathcal{A})$, and $T_{(Q,\mathcal{A})}$ as in the three bullet points below.

- The assists $\omega \in \Omega(Q, \mathcal{A})$ are to be given in short form with parameters $(\Delta_g(\mathcal{A}), \Delta_\epsilon(\mathcal{A}))$.

- The functionals $\xi \in \Xi(Q, \mathcal{A}))$ are to be given in short form with parameters $(\Delta_g(\mathcal{A}), \Delta_\epsilon(\mathcal{A}))$ in terms of the assists $\Omega(Q, \mathcal{A})$.

We define

$$M_{(Q,\mathcal{A})}(f,P) := \Big(\sum_{\xi \in \Xi(Q,\mathcal{A})} |\xi(f,P)|^p \Big)^{1/p}.$$

For each $(f,P) \in \mathbb{X}(\frac{65}{64}Q \cap E) \oplus \mathcal{P}$, we guarantee that

$$c \cdot \|(f,P)\|_{(1+a(\mathcal{A}))Q} \leq M_{(Q,\mathcal{A})}(f,P) \leq C \cdot \Big[\|(f,P)\|_{\frac{65}{64}Q} + \Delta_{\text{junk}}(\mathcal{A}) \cdot |P| \Big].$$

- The operators $T_{(Q,\mathcal{A})}$ map $\mathbb{X}(\frac{65}{64}Q \cap E) \oplus \mathcal{P}$ into $\mathbb{X}$.

  (E1) $T_{(Q,\mathcal{A})}(f,P) = f$ on $(1 + a(\mathcal{A}))Q \cap E$ for each (f,P).

  (E2) $\|T_{(Q,\mathcal{A})}(f,P)\|^p_{\mathbb{X}((1+a(\mathcal{A}))Q)} + \delta_Q^{-mp} \|T_{(Q,\mathcal{A})}(f,P) - P\|^p_{L^p((1+a(\mathcal{A}))Q)} \leq C \left[M_{(Q,\mathcal{A})}(f,P) \right]^p$ for each (f,P).

  (E3) $T_{(Q,\mathcal{A})}$ has $\Omega(Q,\mathcal{A})$-assisted depth at most C.

- The only modification to the algorithm CZ-ORACLE is as follows:
  We assume that the query $\underline{x} \in Q^\circ$ is an S-bit machine point. We compute a list of all the cubes $Q \in CZ(\mathcal{A}^-)$ such that $\underline{x} \in \frac{65}{64}Q$.

  (Recall that $S = K_{\max}\overline{S}$ is the maximum bit length of a machine number representable on our computer.)

- The algorithm COMPUTE MAIN-CUBES is unchanged. We compute and store all the cubes in $CZ_{\text{main}}(\mathcal{A})$.

- The only modifications to the algorithm COMPUTE FUNCTIONALS are as follows.
  The functionals $\omega \in \Omega(Q,\mathcal{A})$ are computed in short form with parameters $(\Delta_g(\mathcal{A}), \Delta_\epsilon(\mathcal{A}))$. The functionals $\xi \in \Xi(Q,\mathcal{A})$ are computed in short form with parameters $(\Delta_g(\mathcal{A}), \Delta_\epsilon(\mathcal{A}))$ in terms of the assists $\Omega(Q,\mathcal{A})$.

- The only modifications to the algorithm COMPUTE EXTENSION OPERATORS are as follows.
  Let $\underline{x} \in Q^\circ$ be an S-bit machine point, and let $\alpha \in \mathcal{M}$. We compute the linear functional $(f,P) \mapsto \partial^\alpha(T_{(Q,\mathcal{A})}(f,P))(\underline{x})$ in short form with parameters $(\Delta_g(\mathcal{A}), \Delta_\epsilon(\mathcal{A}))$ in terms of the assists $\Omega(Q,\mathcal{A})$. This requires work at most $C \log N$, as before.

- All the constants $c_*(\mathcal{A}), S(\mathcal{A}), \epsilon_1(\mathcal{A}), \epsilon_2(\mathcal{A}), a(\mathcal{A}), c, C$ depend on m, n, p, and $\mathcal{A}$. The constant $S(\mathcal{A}) \geq 1$ is an integer. We further assume that $a(\mathcal{A})$ is an integer power of 2. (This is a new assumption in the finite-precision case.)

- We perform the above computations using one-time work at most $CN \log N$ and storage at most CN.

2.9. Algorithms for dyadic cubes

We make the following assumptions.

- We are given machine numbers $\Delta_\epsilon = 2^{-K_1 \overline{S}}$ and $\Delta_g = 2^{-K_2 \overline{S}}$, for integers $K_1, K_2 \geq 1$.
- We assume that our computer can perform arithmetic operations on S-bit machine numbers with precision $\Delta_{\min} = 2^{-S}$, where $S = K_{\max} \cdot \overline{S}$.
- We assume that $\Delta_{\min} \leq \Delta_\epsilon^C$, $\Delta_\epsilon \leq \Delta_g^C$, and $\Delta_g \leq 2^{-C\overline{S}}$ for a large enough universal constant C.

Whenever we refer to a machine number in this section, we mean an S-bit machine number, with S as above.

We call a dyadic cuboid $Q = \prod_{j=1}^n I_j \subset \mathbb{R}^n$ a "machine cuboid" if each I_j is an interval of the form $[a_j, b_j)$, where a_j and b_j are machine numbers. Recall that each I_j is contained in $[0, \infty)$, by definition of cuboids (see Section 4.1.1 in [3]).

Let Q and Q' be given machine cuboids. The following task can be performed using one unit of "work":

(2.76) Compute the smallest machine cuboid Q containing both Q' and Q''.

Let us explain why we charge only one unit of work to perform the task (2.76).

We suppose that a non-negative machine number x is represented in the computer by its binary digits $(x_i)_{-S \leq i \leq S}$, where

$$x = \sum_{i=-S}^{+S} x_i 2^i \quad \text{and each } x_i \in \{0, 1\}.$$

We suppose that the bit pattern $(x_i)_{-S \leq i \leq S}$ fits in a single machine word. Given two distinct non-negative machine numbers x, y with binary digits $(x_i)_{-S \leq i \leq S}$, $(y_i)_{-S \leq i \leq S}$ respectively, we return the largest i_* for which $x_{i_*} \neq y_{i_*}$. Recall that in our model of computation for finite-precision arithmetic, we assume that the computation of i_* from (x_i) and (y_i) takes one unit of "work". (See Section 2.1.) Moreover, there are computers in use today for which the computation of i_* from (x_i) and (y_i) may be accomplished by executing $O(1)$ assembly language instructions.

Note that the smallest smallest dyadic interval containing x and y has length 2^{i_*}. It follows easily that the task (2.76) may be accomplished using at most C operations. That is why we consider it reasonable to charge one unit of "work" to carry out (2.76).

Therefore, we can determine whether $Q < Q'$, $Q' < Q$, or $Q = Q'$, using $O(1)$ computer operations. We refer here to the order relation on dyadic cuboids defined in Section 4.1.1 of [3].

We should point out that the task (2.76) appears to require more than $O(1)$ operations in several standard models of computation (not used here). See the discussion of "quad trees" and "segment trees" in [1].

We will obtain versions of the algorithms in Section 4.1.2 of [3] which are adapted to our finite-precision model of computation.

A modification we will make throughout is that all the cuboids that are input data to an algorithm will be assumed to be machine cuboids, while all the cuboids that are produced as output data are guaranteed to be machine cuboids. We can clearly store a machine cuboid on our computer using $O(1)$ units of storage.

• **Modification 1.** We introduce a bit of notation relevant to the notion of DTrees and ADTrees. See the discussion in Section 4.1.2 of [3].

Recall that each node x of a DTree T is marked with a dyadic cuboid Q_x. When we speak of a DTree T in this section, it is assumed that Q_x is a machine cuboid for each $x \in T$.

Recall that each node x of an ADTree T is marked with linear functionals $\mu_1^x, \ldots, \mu_D^x$ on $\mathbb{R}^D$. We write $\mu_i^x : (\nu_1, \ldots, \nu_D) \mapsto \sum_{j=1}^D \theta_{ij}^x \nu_j$.

We will assume that $D \leq C$ for a universal constant C, in what follows.

We say that $\mu_1^x, \ldots, \mu_D^x$ are specified with parameters $(\Delta_g, \Delta_\epsilon)$ if $|\theta_{ij}^x| \leq \Delta_g^{-1}$ and if each θ_{ij}^x is specified to precision Δ_ϵ. If that's the case for each node x and if the number of nodes of the ADTree is at most Δ_g^{-1}, then we say that the ADTree T is specified with parameters $(\Delta_g, \Delta_\epsilon)$.

• All of the algorithms that involve BTrees are combinatorial in nature, hence they remain the same in our finite-precision model of computation. In particular, BTree1 and MAKE CONTROL TREE (deluxe edition and paperback edition) are unchanged.

• **Modification 2.** We make the following changes to the algorithm MAKE CONTROL TREE (HYBRID VERSION) (see Section 4.1.3 of [3]).

Assume that an ADTree T is given with parameters $(\Delta_g, \Delta_\epsilon)$, with each node x in T marked by linear functionals $\mu_1^x, \ldots, \mu_D^x$ on $\mathbb{R}^D$. Also, we are given a machine number $\Delta \in [\Delta_g, 1]$ of the form $\Delta = 2^{-K\overline{S}}$ for an integer $K \geq 1$.

Then we compute the control tree $CT(T)$, with all its markings except for the trees $BT(\xi)$ ($\xi \in CT(T)$). For each node $\xi \in CT(T)$, we compute functionals $\mu_1^\xi, \ldots, \mu_D^\xi : \mathbb{R}^D \to \mathbb{R}$ of the form

$$\mu_i^\xi : (\nu_1, \ldots, \nu_D) \mapsto \sum_{j=1}^D \theta_{ij}^\xi \nu_j.$$

The numbers θ_{ij}^ξ are computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. That is, we guarantee that $|\theta_{ij}^\xi| \leq \Delta_g^{-C}$ and each θ_{ij}^ξ is computed to precision $\Delta_g^{-C}\Delta_\epsilon$. We guarantee that for each $\xi \in CT(T)$ we have

$$c \sum_{i=1}^D |\mu_i^\xi(\nu)|^p \leq \sum_{x \in BT(\xi)} \sum_{i=1}^D |\mu_i^x(\nu)|^p + \Delta^p |\nu|^p \leq C \sum_{i=1}^D |\mu_i^\xi(\nu)|^p. \tag{2.77}$$

Recall that we denote $|\nu|^p = \sum_j |\nu_j|^p$ for $\nu = (\nu_1, \ldots, \nu_D)$.

The work and storage requirements are the same as before.

That completes the list of modifications to the hybrid version of MAKE CONTROL TREE.

To obtain this result, we apply the finite-precision version of COMPRESS NORMS (see Section 2.6) where before we used its infinite-precision counterpart. The proof of (2.77) is exactly as before.

• **Modification 3.** In the algorithm ENCAPSULATE: Assume that T is a DTree with N nodes such that each node x in T is marked with a machine cuboid Q_x. We perform $CN(1+\log N)$ one-time work in space CN after which we can answer queries. A query consists of a machine cuboid Q. The response to a query is an encapsulation S of Q, consisting of at most $C + C\log N$ nodes of $CT(T)$. The work and storage used to answer a query are at most $C + C\log N$, where C denotes a constant depending only on the dimension n.

For the explanation of the algorithm, just note that one can compare two machine cuboids to determine whether one contains the other, using at most C units of work. Thus, we can proceed as in the infinite-precision version of the algorithm ENCAPSULATE using our finite-precision computer.

• **Modification 4.** In the algorithm ADPROCESS (see Section 4.1.4 of [3]):

We assume our ADTree T is given with parameters $(\Delta_g, \Delta_\epsilon)$. We are given a machine number $\Delta \in [\Delta_g, 1]$ of the form $\Delta = 2^{-k\overline{s}}$ for an integer $k \geq 1$.

A query consists of a machine cuboid Q. The response to a query is a list of linear functionals $\mu_1^Q, \ldots, \mu_D^Q$ on $\mathbb{R}^D$ such that

$$c\sum_{i=1}^{D}|\mu_i^Q(v)|^p \leq \sum_{\substack{x\in T\\ Q_x\subset Q}}\sum_{i=1}^{D}|\mu_i^x(v)|^p + \Delta^p\log(\Delta_g^{-1})|v|^p \leq C\Big[\sum_{i=1}^{D}|\mu_i^Q(v)|^p + \Delta^p\log(\Delta_g^{-1})|v|^p\Big] \quad \text{for all } v\in\mathbb{R}^D. \tag{2.78}$$

Each μ_i^Q has the form $\mu_i^Q : (v_1, \ldots, v_D) \to \sum_{j=1}^{D}\theta_{ij}^Q v_j$. We compute each θ_{ij}^Q with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

The work and storage requirements remain the same as before. That completes the list of modifications to the statement of ADPROCESS.

We present the modifications needed in the explanation of the algorithm. We use the finite-precision versions of the algorithms MAKE CONTROL TREE (HYBRID VERSION) and COMPRESS NORMS in place of the infinite-precision counterparts. In the one-time work, we compute the control tree $CT(T)$. Each node $\xi \in CT(T)$ is marked with linear functionals $\mu_1^\xi, \ldots, \mu_D^\xi$ satisfying (2.77). We compute the functionals μ_k^ξ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

Using the algorithm ENCAPSULATE, we respond to a query as follows.

Given a machine cuboid Q, we produce a set S of at most $C + C\log N$ nodes in $CT(T)$ such that $\{x \in T : Q_x \subset Q\}$ is the disjoint union over $\xi \in S$ of $BT(\xi)$. Therefore, by the finite-precision version of MAKE CONTROL TREE (HYBRID VERSION)

(see **Modification 2** above), the expression

$$\mathfrak{E}_1 = \sum_{\substack{x\in T\\ Q_x\subset Q}} \sum_{i=1}^{D} |\mu_i^x(\nu)|^p + \#(S)\cdot\Delta^p\cdot|\nu|^p = \sum_{\xi\in S}\Big[\sum_{x\in BT(\xi)}\sum_{i=1}^{D}|\mu_i^x(\nu)|^p + \Delta^p\cdot|\nu|^p\Big]$$

differs by at most a factor of C from

$$\mathfrak{E}_2 = \sum_{\xi\in S}\sum_{i=1}^{D}|\mu_i^\xi(\nu)|^p.$$

Applying COMPRESS NORMS (finite-precision) (see Section 2.6) to the expression $\mathfrak{E}_2$, we compute linear functionals $\mu_1^Q,\dots,\mu_D^Q$ such that $\mathfrak{E}_2+\Delta^p\cdot|\nu|^p$ differs by at most a factor of C from $\sum_{i=1}^D|\mu_i^Q(\nu)|^p$. Each functional μ_i^Q is computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

Therefore, $\sum_{i=1}^D|\mu_i^Q(\nu)|^p$ differs by at most a factor of C from

$$\mathfrak{E}_1 + \Delta^p|\nu|^p = \sum_{\substack{x\in T\\ Q_x\subset Q}}\sum_{i=1}^{D}|\mu_i^x(\nu)|^p + (\#(S)+1)\cdot\Delta^p\cdot|\nu|^p$$

Note that

$$\#(S) \le C + C\log(\#(T)) \le C\log(\Delta_g^{-1}),$$

so the junk term

$$(\#(S)+1)\cdot\Delta^p\cdot|\nu|^p$$

is bounded by $C\,\Delta^p\log(\Delta_g^{-1})|\nu|^p$. That concludes the proof of (2.78).

The work and storage requirements are as promised.

• **Modification 5.** In the algorithms MAKE FOREST, FILL IN GAPS, and MAKE DTREE: All dyadic cuboids are assumed to be machine cuboids. The explanations of these algorithms require no modification.

• **Modification 6.** In COMPUTE NORMS FROM MARKED CUBOIDS:

We suppose our cuboids $Q_1,\dots,Q_N$ have corners whose coordinates are $\widetilde{S}$-bit machine numbers, with $\widetilde{S}\le C\overline{S}$ for a universal constant C. Hence, $N\le 2^{Cn\overline{S}}\le\Delta_0^{-C}$, where we set $\Delta_0 = 2^{-\overline{S}}$.

We are given a machine number $\Delta\in[\Delta_g,1]$ of the form $\Delta = 2^{-K\overline{S}}$ for an integer $K\ge 1$.

Each linear functional $\mu_\ell^{Q_i}$ is given as $\mu_\ell^{Q_i}:(\nu_1,\dots,\nu_D)\mapsto\sum_{j=1}^D\theta_{\ell j}^i\nu_j$. The numbers $\theta_{\ell j}^i$ are specified with parameters $(\Delta_g,\Delta_\epsilon)$. We assume that $\widehat{N} := \sum_{i=1}^N(L_i+1)\le\Delta_g^{-1}$.

A query consists of a dyadic cuboid Q whose corners are machine points.

The response to a query Q is a list of linear functionals $\widehat{\mu}_1^Q, \dots, \widehat{\mu}_D^Q : \mathbb{R}^D \to \mathbb{R}$ for which we guarantee the estimate

$$c \sum_{j=1}^{D} |\widehat{\mu}_j^Q(v)|^p \le \sum_{Q_i \subset Q} \sum_{j=1}^{L_i} |\mu_j^{Q_i}(v)|^p + \Delta^p \Delta_0^{-C} \log(\Delta_g^{-1}) \cdot |v|^p$$
$$\le C \Big[\sum_{j=1}^{D} |\widehat{\mu}_j^Q(v)|^p + \Delta^p \Delta_0^{-C} \log(\Delta_g^{-1}) \cdot |v|^p \Big] \quad \text{for all } v \in \mathbb{R}^D.$$

Each $\widehat{\mu}_i^Q$ has the form $\mu_i^Q : (v_1, \dots, v_D) \mapsto \sum_{j=1}^{D} \theta_{ij}^Q v_j$. The numbers θ_{ij}^Q are computed with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$.

That completes the list of modifications to the algorithm COMPUTE NORMS FROM MARKED CUBOIDS.

The explanation of the algorithm is as follows.

For each $i = 1, \dots, N$ we apply the finite-precision version of COMPRESS NORMS (see Section 2.6) to produce linear functionals $\overline{\mu}_j^{Q_i}$ on $\mathbb{R}^D$ for $1 \le j \le D$ such that

$$c \cdot \sum_{j=1}^{D} \big|\overline{\mu}_j^{Q_i}(v)\big|^p \le \sum_{j=1}^{L_i} \big|\mu_j^{Q_i}(v)\big|^p + \Delta^p \cdot |v|^p \le C \cdot \sum_{j=1}^{D} \big|\overline{\mu}_j^{Q_i}(v)\big|^p. \tag{2.79}$$

Note that each $L_i \le \Delta_g^{-1}$ by assumption, so the algorithm may be applied as stated. The functionals $\overline{\mu}_j^{Q_i}$ are given with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$.

Using the algorithm MAKE DTREE, we construct a DTree T with at most CN nodes, such that each Q_i is a node of T. We mark each Q_i in T with the list of functionals $\overline{\mu}_1^{Q_i}, \dots, \overline{\mu}_D^{Q_i}$, and we mark all the other nodes in T with a list of linear functionals that are all zero. When equipped with these markings, T forms an ADTree. Note that $\#(T) \le CN \le C\Delta_0^{-C} \le \Delta_g^{-1}$. Hence, the ADTree T is specified with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$ (recall that $\overline{\mu}_j^{Q_i}$ are specified with such parameters).

We apply the algorithm ADPROCESS to the ADTree T. Thus, given a machine cube Q, we can compute a list of linear functionals $\widehat{\mu}_1^Q, \dots, \widehat{\mu}_D^Q$ on $\mathbb{R}^D$ such that

$$c \cdot \sum_{j=1}^{D} \big|\widehat{\mu}_j^Q(v)\big|^p \le \sum_{\substack{1 \le i \le N \\ Q_i \subset Q}} \sum_{j=1}^{D} \big|\overline{\mu}_j^{Q_i}(v)\big|^p + \Delta^p \log(\Delta_g^{-C}) \cdot |v|^p$$
$$\le C \cdot \Big[\sum_{j=1}^{D} \big|\widehat{\mu}_j^Q(v)\big|^p + \Delta^p \log(\Delta_g^{-C}) \cdot |v|^p \Big].$$

Note that $\Delta \in [\Delta_g^C, 1]$, so the algorithm may be applied as stated. Using (2.79), we determine that

$$c \cdot \sum_{j=1}^{D} \big|\widehat{\mu}_j^Q(v)\big|^p \le \sum_{\substack{1 \le i \le N \\ Q_i \subset Q}} \sum_{j=1}^{L_i} \big|\mu_j^{Q_i}(v)\big|^p + \mathfrak{E}(v) \le C \cdot \Big[\sum_{j=1}^{D} \big|\widehat{\mu}_j^Q(v)\big|^p + \mathfrak{E}(v) \Big],$$

where

$$\mathfrak{E}(v) = \sum_{\substack{1 \le i \le N \\ Q_i \subset Q}} \Delta^p \cdot |v|^p + \Delta^p \log(\Delta_g^{-C}) \cdot |v|^p \,.$$

Since $N \le \Delta_0^{-C}$, we conclude that

$$\mathfrak{E}(v) \le \Delta^p \Delta_0^{-C} \log(\Delta_g^{-C}) \cdot |v|^p \le \Delta^p \Delta_0^{-C'} \log(\Delta_g^{-1}) \cdot |v|^p \,.$$

The previous estimate implies the desired condition on the functionals $\widehat{\mu}_1^Q, \ldots, \widehat{\mu}_D^Q$.

This completes the explanation of the finite-precision version of the algorithm COMPUTE NORMS FROM MARKED CUBOIDS.

• The algorithm PLACING A POINT INSIDE TARGET CUBOIDS requires minor changes in finite-precision. We assume that $Q_1, \ldots, Q_N$ are machine cuboids, and that the query $\underline{x}$ is a machine point. The response to a query $\underline{x}$ is either one of the Q_i containing $\underline{x}$, or else a promise that no such Q_i exists. The work to answer a query is at most $C \cdot (1 + \log N)$. The explanation of the algorithm requires no modification.

This concludes the description of the changes required in Section 4.1 of [3].

Aside from the following modifications, all the algorithms in Sections 4.2–4.5 of [3] are unchanged in our finite-precision model of computation.

- Each point $x \in E$ is an $\overline{S}$-bit machine point (i.e., the coordinates of x are $\overline{S}$-bit machine numbers).
- All numbers are S-bit machine numbers and all given points are S-bit machine points, where $S = K_{max}\overline{S}$.
- Each dyadic cuboid has $\widetilde{S}$-bit machine points as corners, where $\widetilde{S} \le C\overline{S}$.

2.10. CZ decompositions

We describe the modifications required in Section 4.6 of [3]. We let $\Delta_{min} = 2^{-S}$ denote the machine precision of our computer, where $S = K_{max}\overline{S}$.

We call a dyadic cube $Q = \prod_{k=1}^n I_k \subset \mathbb{R}^n$ a "machine cube" if each I_k is an interval of the form $[a_k, b_k)$, where a_k and b_k are machine numbers.

- In Sections 4.6.2 and 4.6.3 of [3], we assume we given a subset $E \subset Q^\circ = [0,1)^n$. We assume that each point in E is an $\overline{S}$-bit machine point. Hence, $|x - y| \ge \Delta_0 = 2^{-\overline{S}}$ for distinct $x, y \in E$.
- In Section 4.6.3 of [3], we assume we are given a list $\vec{\Delta} = (\Delta(x))_{x \in E}$ consisting of positive real numbers. We assume that the numbers $\Delta(x)$ are $\widetilde{S}$-bit machine numbers, where $\widetilde{S} \le C\overline{S}$ for a universal constant $C \ge 1$. Hence, $\Delta(x) \ge 2^{-\widetilde{S}} \ge \Delta_0^C$ for all $x \in E$.

We recall the definition of the Calderón–Zygmund decomposition $CZ(\vec{\Delta})$:

- $CZ(\vec{\Delta})$ consists of the maximal dyadic cubes $Q \subset Q^\circ$ such that either $\#(E \cap 3Q) \le 1$ or $\Delta(x) \ge \delta_Q$ for all $x \in E \cap 3Q$.

For all $Q \in CZ(\vec{\Delta})$, either $Q = Q^\circ$ or $\#(9Q \cap E) \geq \#(3Q^+ \cap E) \geq 2$. Furthermore, $|x - y| \geq \Delta_0$ for any distinct $x, y \in E$. Hence, $\delta_Q \geq c \cdot \Delta_0$ for any $Q \in CZ(\vec{\Delta})$. It follows that the cubes in $CZ(\vec{\Delta})$ have $\widetilde{S}$-bit machine points as corners, where $\widetilde{S} \leq C\overline{S}$.

The PLAIN VANILLA CZ-ORACLE in finite-precision operates as follows. Given an S-bit machine point $\underline{x} \in Q^\circ$, return the cube $Q \in CZ(\vec{\Delta})$ that contains $\underline{x}$. The work to answer a query is at most $C \log N$. The explanation is identical as before.

Now, suppose we are given a dyadic decomposition CZ_{old} of the unit cube Q°, satisfying the properties laid out in Section 4.6.3 of [3]. Suppose in addition that each $Q \in CZ_{old}$ is a machine cube. Suppose we have available a CZ_{old}-ORACLE that operates as follows: given an S-bit machine point $\underline{x} \in Q^\circ$, return the cube $Q \in CZ_{old}$ that contains $\underline{x}$.

We recall the definition of the Calderón–Zygmund decomposition CZ_{new}:

- CZ_{new} consists of the maximal dyadic cubes $Q \subset Q^\circ$ such that either $Q \in CZ_{old}$ or $\Delta(x) \geq \delta_Q$ for all $x \in E \cap 3Q$.

Note that CZ_{old} is a refinement of CZ_{new}. Since the cubes in CZ_{old} are machine cubes, it follows that the cubes in CZ_{new} are also machine cubes.

The GLORIFIED CZ-ORACLE in finite-precision operates as follows: A query consists of an S-bit machine point $\underline{x} \in Q^\circ$. The response to a query is a list of the cubes $Q \in CZ_{new}$ such that $\underline{x} \in \frac{65}{64}Q$. The work to answer a query is at most $C \log N$ computer operations, as well as at most C calls to the CZ_{old}-ORACLE. The explanation of the finite-precision version of the algorithm is unchanged.

This concludes the description of the finite-precision versions of the algorithms in Sections 4.6.2 and 4.6.3 of [3].

We now turn to Sections 4.6.4 and 4.6.5 of [3].

We are given a set $E \subset \frac{1}{32}Q^\circ$, with $\#(E) = N \geq 2$. We assume that E consists of $\overline{S}$-bit machine points.

We are given a locally finite collection CZ, consisting of dyadic cubes, that partitions Q° (or $\mathbb{R}^n$). We do not list all the cubes in CZ. Instead, we have available a CZ-ORACLE that operates as follows: given an S-bit machine point $\underline{x} \in Q^\circ$ (or $\underline{x} \in \mathbb{R}^n$), the oracle responds with a list of all $Q \in CZ$ such that $\underline{x} \in \frac{65}{64}Q$. We guarantee that every such Q is an $\widetilde{S}$-bit machine cube with $\widetilde{S} \leq C\overline{S}$. We charge at most $C \log N$ units of work to answer a query.

Under these assumptions, we have versions of the algorithms FIND NEIGHBORS and FIND MAIN-CUBES (see Section 4.6.4 of [3]) for our finite-precision model of computation. The explanations are unchanged. Since it will be used in the next section, we record here the statement of the latter algorithm:

ALGORITHM: FIND MAIN-CUBES (FINITE-PRECISION)

After one-time work at most $CN \log N$ in space CN, we produce the collection of cubes $CZ_{main} := \{Q \in CZ : \frac{65}{64}Q \cap E \neq \emptyset\}$. We mark each cube $Q \in CZ_{main}$ with a point $x(Q) \in \frac{65}{64}Q \cap E$.

We also have a version of COMPUTE CUTOFF FUNCTION for our finite-precision model of computation.

ALGORITHM: COMPUTE CUTOFF FUNCTION (FINITE-PRECISION)

Given machine numbers Δ_ϵ and Δ_g, which are large integer powers of $\Delta_0 = 2^{-\overline{S}}$, given a machine number $\overline{r} \in (\Delta_0, 1/64]$, given an S-bit machine point $\underline{x} \in Q^\circ$, and given a machine cube $Q \in CZ$, we compute the numbers $\frac{1}{\alpha!}\partial^\alpha\big(J_{\underline{x}}\widetilde{\theta}_Q\big)(0)$ (all $\alpha \in \mathcal{M}$) with parameters $(\Delta_g, \Delta_\epsilon)$. (See Section 4.6.5 of [3] for statement of the properties of the cutoff functions $\widetilde{\theta}_Q$.)

2.11. Starting the induction

We begin the proof of the finite-precision version of the Main Technical Results for $\mathcal{A}$. (See Section 2.8.) We follow the line of reasoning in [4], used to prove the infinite-precision version of the Main Technical Results for $\mathcal{A}$.

We proceed by induction on the collection of multiindex sets $\mathcal{A} \subset \mathcal{M}$ with respect to the total order relation $<$.

For the base case of the induction, we must prove the Main Technical Results for $\mathcal{A} = \mathcal{M}$.

Recall that $\Delta_g(\mathcal{M})$ and $\Delta_\epsilon(\mathcal{M})$ are assumed to be integer powers of $\Delta_0 = 2^{-\overline{S}}$ that satisfy $\Delta_\epsilon(\mathcal{M}) \ll \Delta_g(\mathcal{M})$ (see (2.72)). We denote $\Delta_g = \Delta_g(\mathcal{M})$ and $\Delta_\epsilon = \Delta_\epsilon(\mathcal{M})$ in the course of this section. Thus, we may assume that $\Delta_\epsilon \le \Delta_g^C$ for a sufficiently large universal constant C.

We define $CZ(\mathcal{M})$ to be the collection of the maximal dyadic cubes $Q \subset Q^\circ$ such that $\#(E \cap 3Q) \le 1$.

From (2.70), we know that

$$\delta_{\min} = \min\{|x - y| : x, y \in E,\ x \ne y\} \ge \Delta_0. \tag{2.80}$$

With at most $CN \log N$ computer operations, we can compute a machine number δ with $c \cdot \delta_{\min} < \delta < \frac{1}{100}\delta_{\min}$ using the BBD Tree (see Theorem 35 in [3]). This data structure requires no modifications in the finite-precision model of computation. We shall use the PLAIN VANILLA CZ-ORACLE (see the description in Section 2.10) with $\vec{\Delta}$ defined by $\Delta(x) := \delta$ for all $x \in E$. This yields the $CZ(\mathcal{M})$-ORACLE, as described in the Main Technical Results for $\mathcal{M}$; see Remark 47 in [3] for further explanation.

Note that $\#(E \cap 3Q^+) \ge 2$ for each $Q \in CZ(\mathcal{M})$, where Q^+ is the dyadic parent of Q. We can thus choose distinct points $x, x' \in E \cap 3Q^+$. From (2.70) we know that $|x - x'| \ge \Delta_0$. Hence, $6\delta_Q = \delta_{3Q^+} \ge \Delta_0$. In particular,

$$\delta_Q \ge \frac{1}{32}\Delta_0 \quad \text{for each } Q \in CZ(\mathcal{M}).$$

Thus the decomposition $CZ(\mathcal{M})$ satisfies the additional property (2.75) required in finite-precision.

Using the algorithm FIND MAIN-CUBES (see Section 2.10), we list all the cubes $Q \in CZ_{\text{main}}(\mathcal{M})$, and we compute a point $x(Q) \in E \cap \frac{65}{64}Q$ associated to each $Q \in CZ_{\text{main}}(\mathcal{M})$.

We define linear maps $T_{(Q,\mathcal{M})}$ and functionals ξ_Q as before, in (1.3) and (1.4) of [4]. We take the assist set $\Omega(Q, \mathcal{M})$ to be empty for each $Q \in CZ_{\text{main}}(\mathcal{M})$.

For each $Q \in CZ_{main}(\mathcal{M})$, the linear functional ξ_Q is computed in short form with parameters $(\Delta_g, \Delta_\epsilon)$. Furthermore, given an S-bit machine point $\underline{x} \in Q^\circ$ and a multiindex $\alpha \in \mathcal{M}$, we compute the linear functional $(f, P) \mapsto \partial^\alpha(T_{(Q,\mathcal{M})}(f, P))(\underline{x})$ with parameters $(\Delta_g, \Delta_\epsilon)$. This completes the description of the changes to the algorithm COMPUTE MAIN-CUBES AND COMPUTE EXTENSION OPERATORS (BASE CASE). The explanation of the algorithm is obvious once we examine the formulas for $T_{(Q,\mathcal{M})}$ and ξ_Q.

That concludes the proof of the finite-precision version of the Main Technical Results for $\mathcal{A} = \mathcal{M}$. This completes the base case of our induction. Next, we turn to the induction step.

2.12. The induction step

Let $\mathcal{A} \subsetneq \mathcal{M}$ be a given multiindex set. Let $\mathcal{A}^- < \mathcal{A}$ be maximal with respect to the order on multiindex sets. Our goal is to prove the Main Technical Results for $\mathcal{A}$. By induction we assume that the Main Technical Results for $\mathcal{A}^-$ hold. We list below a few consequences of these results. (See Section 2.8.)

We denote $\Delta_g := \Delta_g(\mathcal{A}^-)$, $\Delta_\epsilon := \Delta_\epsilon(\mathcal{A}^-)$, and $\Delta_{junk} := \Delta_{junk}(\mathcal{A}^-)$, which are the parameters arising in the Main Technical Results for $\mathcal{A}^-$. These parameters are all large integer powers of $\Delta_0 = 2^{-\overline{S}}$. These parameters are not fixed just yet. Hence, in the course of the proof of the Main Technical Results for $\mathcal{A}$ we may impose additional assumptions on the relationships between these parameters. From (2.74), we may assume estimates of the form

$$\Delta_\epsilon \le \Delta_g^C,\ \Delta_g \le \Delta_{junk}^C, \quad \text{and} \quad \Delta_{junk} \le \Delta_0^C, \quad \text{for a universal constant } C. \tag{2.81}$$

For example, the first estimate in (2.81) is derived from (2.74) as follows:

$$\Delta_\epsilon(\mathcal{A}^-) \le \Delta_\epsilon(\emptyset) \le \Delta_g(\emptyset)^C \le \Delta_g(\mathcal{A}^-)^C.$$

The other conditions can be derived in a similar fashion.

By the induction hypothesis, we have defined a dyadic decomposition $CZ(\mathcal{A}^-)$ of the unit cube Q°, which satisfies conditions **(CZ1)**–**(CZ5)** in the Main Technical Results for $\mathcal{A}^-$. Furthermore, from the finite-precision version of these results, we have

$$\delta_Q \ge \frac{1}{32} \cdot \Delta_0 \ \text{ for each } Q \in CZ(\mathcal{A}^-).$$

Recall that $CZ_{main}(\mathcal{A}^-)$ is the collection of all the cubes $Q \in CZ(\mathcal{A}^-)$ such that $\frac{65}{64}Q \cap E \neq \emptyset$.

According to the Main Technical Results for $\mathcal{A}^-$, we have computed a list of all the cubes in $CZ_{main}(\mathcal{A}^-)$. Furthermore, we have access to a $CZ(\mathcal{A}^-)$-ORACLE that operates as follows: given an S-bit machine point $\underline{x} \in Q^\circ$, we list all the cubes $Q \in CZ(\mathcal{A}^-)$ such that $\underline{x} \in \frac{65}{64}Q$, using work at most $C \log N$.

We have computed a list of assists $\Omega(Q, \mathcal{A}^-)$, and a list of assisted functionals $\Xi(Q, \mathcal{A}^-)$ for each $Q \in CZ_{main}(\mathcal{A}^-)$. Each $\omega \in \Omega(Q, \mathcal{A}^-)$ is a linear functional on $\mathbb{X}(\frac{65}{64}Q \cap E)$, and is given in short form with parameters $(\Delta_g, \Delta_\epsilon)$; each $\xi \in \Xi(Q, \mathcal{A}^-)$ is a linear functional on $\mathbb{X}(\frac{65}{64}Q \cap E) \oplus \mathcal{P}$, and is given in short

form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of the assists $\Omega(Q, \mathcal{A}^-)$. We guarantee that

$$c \cdot \|(f,P)\|_{(1+a(\mathcal{A}^-))Q} \le M_{(Q,\mathcal{A}^-)}(f,P) \le C \cdot \left[\|(f,P)\|_{\frac{65}{64}Q} + \Delta_{\text{junk}}|P|\right]$$

where

$$M_{(Q,\mathcal{A}^-)}(f,P) := \Big(\sum_{\xi \in \Xi(Q,\mathcal{A}^-)} |\xi(f,P)|^p\Big)^{1/p}.$$

We recall that $|P| = \big(\sum_{\alpha\in\mathcal{M}} |\partial^\alpha P(0)|^p\big)^{1/p}$.

We have computed a linear map $T_{(Q,\mathcal{A}^-)} : \mathbb{X}(\frac{65}{64}Q \cap E) \oplus \mathcal{P} \to \mathbb{X}$ for each $Q \in CZ_{\text{main}}(\mathcal{A}^-)$ in the following sense: given an S-bit machine point $\underline{x} \in Q^\circ$ and a multiindex $\alpha \in \mathcal{M}$, we compute the linear functional

$$(f,P) \mapsto \partial^\alpha(T_{(Q,\mathcal{A}^-)}(f,P))(\underline{x})$$

in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of the assists $\Omega(Q, \mathcal{A}^-)$. This computation requires work at most $C \log N$.

The above computations are carried out using one-time work at most $CN \log N$ in space at most CN, thanks to the induction hypothesis.

The finite-precision version of the algorithm APPROXIMATE OLD TRACE NORM is as follows.

ALGORITHM: APPROXIMATE OLD TRACE NORM (FINITE-PRECISION)

For each $Q \in CZ_{\text{main}}(\mathcal{A}^-)$ we compute linear functionals $\xi_1^Q, \dots, \xi_D^Q$ on $\mathcal{P}$, such that

$$c \cdot \sum_{i=1}^{D} |\xi_i^Q(P)|^p \le \sum_{\xi\in\Xi(Q,\mathcal{A}^-)} |\xi(0,P)|^p + \Delta_g^p \cdot |P|^p \le C \cdot \sum_{i=1}^{D} |\xi_i^Q(P)|^p.$$

The functionals ξ_i^Q have the form $\xi_i^Q : (P) \mapsto \sum_{\alpha\in\mathcal{M}} \theta_{i\alpha}^Q \cdot \frac{1}{\alpha!}\partial^\alpha P(0)$. The numbers $\theta_{i\alpha}^Q$ are computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

Explanation. The explanation proceeds just as in infinite-precision. We simply apply the finite-precision version of the algorithm COMPRESS NORMS (with $\Delta = \Delta_g$) instead of the infinite-precision version of this algorithm. See Section 2.6. □

That completes the description of the main technical results for $\mathcal{A}^-$.

We now begin the proof of the main technical results for $\mathcal{A}$.

2.12.1. The non-monotonic case. Section 1.3.1 of [4] requires no change in finite-precision. For a non-monotonic set $\mathcal{A} \subsetneq \mathcal{M}$, we can simply define $CZ(\mathcal{A}) = CZ(\mathcal{A}^-)$, and

$$\Omega(Q,\mathcal{A}) = \Omega(Q,\mathcal{A}^-),\ \Xi(Q,\mathcal{A}) = \Xi(Q,\mathcal{A}^-),\ \text{ and}$$
$$T_{(Q,\mathcal{A})} = T_{(Q,\mathcal{A}^-)} \ \text{ for each } Q \in CZ_{\text{main}}(\mathcal{A}) = CZ_{\text{main}}(\mathcal{A}^-).$$

As before, the Main Technical Results for $\mathcal{A}$ (finite-precision) follow from the Main Technical Results for $\mathcal{A}^-$ (finite-precision).

We can compute everything to the desired precision provided that $\Delta_\epsilon(\mathcal{A}) \geq \Delta_\epsilon(\mathcal{A}^-)$, $\Delta_g(\mathcal{A}) \leq \Delta_g(\mathcal{A}^-)$, and $\Delta_{\text{junk}}(\mathcal{A}) \geq \Delta_{\text{junk}}(\mathcal{A}^-)$. These conditions are allowed in view of the assumptions in (2.74),

This proves the Main Technical Results for a nonmonotonic set $\mathcal{A}$.

2.12.2. The monotonic case. Henceforth, we assume that $\mathcal{A}$ is a monotonic set.

The statement and proof of Proposition 1 of [4] is the same in finite-precision, except for the following changes.

- Given a query consisting of an S-bit machine point $\underline{x} \in \mathbb{R}^n$ such that $|\underline{x}| \leq 2^{\overline{S}}$, the $\overline{CZ}(\mathcal{A}^-)$-ORACLE returns a list of the cubes $Q \in \overline{CZ}(\mathcal{A}^-)$ such that $\underline{x} \in \frac{65}{64}Q$. The work required to answer a query is at most $C \log N$.

One can check that the explanation for the $\overline{CZ}(\mathcal{A}^-)$-ORACLE given in Proposition 1 of [4] applies equally well in the finite-precision setting (under the additional hypotheses on $\underline{x}$ stated above). Indeed, since $|\underline{x}| \leq 2^{\overline{S}}$, we see that each of the dyadic cubes that is relevant to the explanation of the algorithm is contained in the rectangular box $[-2^{C\overline{S}}, 2^{C\overline{S}}]^n$ and has sidelength in the interval $[2^{-C\overline{S}}, 2^{C\overline{S}}]$ for a universal constant C. Therefore, the relevant dyadic cubes have $C\overline{S}$-bit machine points as corners. We may assume that $C\overline{S} \leq S$, and therefore all the relevant dyadic cubes are S-bit machine cubes and can be processed on our finite-precision computer, which allows the previous explanation to apply in the current setting.

This concludes the description of changes needed in Section 1.3.2 of [4].

2.12.3. Keystone cubes. Section 1.3.3 of [4] is unchanged in finite-precision. We define integer constants S_0, S_1, S_2 as in (1.17) of [4]. The KEYSTONE-ORACLE is unchanged. The explanation follows just as before from the MAIN KEYSTONE CUBE ALGORITHM and the algorithm LIST ALL KEYSTONE CUBES.

2.13. An approximation to the sigma

Given a polynomial $P \in \mathcal{P}$, we define

$$|P| := \Big(\sum_{\alpha \in \mathcal{M}} |\partial^\alpha P(0)|^p \Big)^{1/p}.$$

Recall that the parameters $\Delta_\epsilon(\mathcal{A}^-)$, $\Delta_g(\mathcal{A}^-)$, and $\Delta_{\text{junk}}(\mathcal{A}^-)$, are denoted by Δ_ϵ, Δ_g, and Δ_{junk}, respectively.

We denote $\Delta_{\text{new}} = \Delta_{\text{junk}}(\mathcal{A})$, which is the constant arising in the Main Technical Results for $\mathcal{A}$. We assume that Δ_{new} satisfies

$$\begin{cases} \Delta_\epsilon \leq \Delta_g \leq \Delta_{\text{junk}} \leq \Delta_{\text{new}}^5 \leq \Delta_{\text{new}} \leq \Delta_0^C, \\ C \cdot \Delta_{\text{new}} \leq 1 \quad \text{for a large enough universal constant } C. \end{cases} \tag{2.82}$$

The conditions in (2.82) are easily derived from (2.74).

We have the following estimates from the finite-precision version of the Main Technical Results for $\mathcal{A}^-$. Let $a = a(\mathcal{A}^-)$ denote the geometric constant arising therein. Then for each $Q \in CZ_{\text{main}}(\mathcal{A}^-)$, the functional

$$M_{(Q,\mathcal{A}^-)}(f,R) = \Big(\sum_{\xi \in \Xi(Q,\mathcal{A}^-)} |\xi(f,R)|^p \Big)^{1/p} \tag{2.83}$$

satisfies

$$c \cdot \|(f,R)\|_{(1+a)Q} \le M_{(Q,\mathcal{A}^-)}(f,R) \le C \cdot \Big[\|(f,R)\|_{\frac{65}{64}Q} + \Delta_{\text{junk}}|R| \Big]. \tag{2.84}$$

Estimate (2.84) holds additionally for all $Q \in \overline{CZ}(\mathcal{A}^-) \setminus CZ_{\text{main}}(\mathcal{A}^-)$, because then by definition $\Xi(Q,\mathcal{A}^-) = \emptyset$ and so $M_{(Q,\mathcal{A}^-)}(f,R) = 0$; also, $\|(f,R)\|_{(1+a)Q} = \|(f,R)\|_{\frac{65}{64}Q} = 0$, since $\frac{65}{64}Q \cap E = \emptyset$. Thus, (2.84) holds for all $Q \in \overline{CZ}(\mathcal{A}^-)$.

We set

$$\mathcal{I}(Q^\#) := \big\{ Q \in \overline{CZ}(\mathcal{A}^-) : Q \cap S_0 Q^\# \neq \emptyset \big\}.$$

(Recall that $S_0 = S(\mathcal{A}^-)$.)

Lemma 9 of [4] is unchanged in finite-precision. Similarly, the conditions (1.24) and (1.25) in [4] continue to hold.

The finite-precision version of the algorithm MAKE NEW ASSISTS AND ASSIGN KEYSTONE JETS is as follows.

ALGORITHM: MAKE NEW ASSISTS AND ASSIGN KEYSTONE JETS (FIN-PREC)

For each keystone cube $Q^\#$, we compute a list of new assists $\Omega^{\text{new}}(Q^\#)$ and we compute an $\Omega^{\text{new}}(Q^\#)$-assisted bounded depth linear map $R^\#_{Q^\#} : \mathbb{X}(S_1 Q^\# \cap E) \oplus \mathcal{P} \to \mathcal{P}$.

Each of the new assists $\omega \in \Omega^{\text{new}}(Q^\#)$ is given in the form

$$\omega : f \mapsto \sum_{x \in S} \lambda_x \cdot f(x).$$

Here, the set $S \subset E$ may depend on ω. The coefficients λ_x are computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. The sum of $\text{depth}(\omega) = \#(S)$ over all the new assists $\omega \in \Omega^{\text{new}}(Q^\#)$ and over all keystone cubes $Q^\#$, is bounded by CN.

Similarly, the maps $(f,P) \mapsto R^\# = R^\#_{Q^\#}(f,P)$ are such that

$$(f,P) \mapsto \partial^\alpha R^\#(0) \quad (\text{for any } \alpha \in \mathcal{A}) \text{ has the form}$$
$$\sum_{x \in S} \lambda_x \cdot f(x) + \sum_{\omega \in \Omega'} \mu_\omega \cdot \omega(f) + \sum_{\beta \in \mathcal{M}} \theta_\beta \frac{1}{\beta!} \partial^\beta P(0).$$

Here, the subsets $S \subset E$ and $\Omega' \subset \Omega$ may depend on $Q^\#$, and the coefficients $\lambda_x, \mu_\omega, \theta_\beta$ are computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. The number $\#(S) + \#(\Omega') + \#(\mathcal{M})$ is bounded by a universal constant C.

The polynomial $R^\#$ satisfies the following properties.

- $\partial^\alpha(R^\# - P) \equiv 0$ for all $\alpha \in \mathcal{A}$. (This condition is natural because $\mathcal{A}$ is monotonic; see (1.1) in [4].)
- Let $R \in \mathcal{P}$ with $\partial^\alpha(R - P) \equiv 0$ for all $\alpha \in \mathcal{A}$. Then

$$\sum_{Q \in \mathcal{I}(Q^\#)} \sum_{\xi \in \Xi(Q, \mathcal{A}^-)} |\xi(f, R^\#)|^p \le C \sum_{Q \in \mathcal{I}(Q^\#)} \sum_{\xi \in \Xi(Q, \mathcal{A}^-)} \left[|\xi(f, R)|^p + \Delta_g^p |R|^p \right]. \tag{2.85}$$

Explanation. As before, we define coordinates on V_P, which is the space of all polynomials $R \in \mathcal{P}$ such that $\partial^\alpha [R - P] \equiv 0$ for all $\alpha \in \mathcal{A}$. The coordinate map $w \mapsto R_w$ is given by

$$R_w(x) = \sum_{\alpha \in \mathcal{A}} \frac{1}{\alpha!} \partial^\alpha P(0) \cdot x^\alpha + \sum_{j=1}^{k} \frac{1}{\alpha_j!} w_j \cdot x^{\alpha_j} \quad \text{for } w = (w_1, \dots, w_k) \in \mathbb{R}^k,$$

where $\mathcal{M} \setminus \mathcal{A} = \{\alpha_1, \dots, \alpha_k\}$. Note that

$$|R_w|^p = \sum_{j=1}^{k} |w_j|^p + \sum_{\beta \in \mathcal{A}} |\partial^\beta P(0)|^p \ge \sum_{j=1}^{k} |w_j|^p. \tag{2.86}$$

We compute a list $\mathfrak{L}$ of all the $Q \in CZ_{\text{main}}(\mathcal{A}^-)$ such that $Q \cap S_0 Q^\# \neq \emptyset$. We produce this list by the same method used in infinite-precision (recall that $S_0 = S(\mathcal{A}^-) \in \mathbb{N}$ is a universal constant, as stated in the Main Technical Results for $\mathcal{A}^-$.)

From the Main Technical Results for $\mathcal{A}^-$ (finite-precision), we can compute a list of the functionals $\xi_\ell : (f, R_w) \mapsto \xi(f, R_w)$, with $1 \le \ell \le L$, where ξ is an arbitrary element of the list $\Xi(Q, \mathcal{A}^-)$ for some $Q \in \mathfrak{L}$. Each ξ_ℓ is expressed in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of the assists $\Omega(Q, \mathcal{A}^-)$. This expression is given in (1.28) of [4], where

- The numbers $\check{\mu}_{\ell j}$ are specified with parameters $(\Delta_g, \Delta_\epsilon)$;
- The functionals λ_ℓ, $\check{\lambda}_\ell$, and the coefficients $\mu_{\ell a}$ are specified with parameters $(\Delta_g, \Delta_\epsilon)$.
- We have $L \le CN \le \Delta_g^{-C}$ for a large enough universal constant C. (Recall that $N \le \Delta_0^{-n} \le \Delta_g^{-1}$; see (2.71) and (2.81).)

We process the functionals $w \mapsto \xi_\ell(f, R_w)$, with f and $(\partial^\alpha P(0))_{\alpha \in \mathcal{A}}$ held fixed, using the algorithm OPTIMIZE VIA MATRIX (finite-precision), where we set $\Delta = \Delta_g$ (see Section 2.7). Thus, we can compute (see below) numbers $b_{j\ell}$, such that, for

$$\begin{cases} \omega_j^{\text{new}}(f) = \sum_{\ell=1}^{L} b_{j\ell} \left[\lambda_\ell(f) + \sum_{a=1}^{I_\ell} \mu_{\ell a} \omega_{\ell a}(f) \right] \\ \lambda_j^{\text{new}}((\partial^\alpha P(0))_{\alpha \in \mathcal{A}}) = \sum_{\ell=1}^{L} b_{j\ell} \cdot \check{\lambda}_\ell((\partial^\alpha P(0))_{\alpha \in \mathcal{A}}), \end{cases} \tag{2.87}$$

we have

$$\sum_{Q\in\mathcal{I}(Q^{\#})}\sum_{\xi\in\Xi(Q,\mathcal{A}^-)}|\xi(f,R_{w^*})|^p$$

$$\leq C\sum_{Q\in\mathcal{I}(Q^{\#})}\sum_{\xi\in\Xi(Q,\mathcal{A}^-)}\Big[|\xi(f,R_w)|^p+\Delta_g^p\sum_{j=1}^{k}|w_j|^p\Big], \tag{2.88}$$

for all $w=(w_1,\dots,w_k)\in\mathbb{R}^k$, where

$$\begin{aligned}&w^*=(w_1^*,\dots,w_k^*),\ \text{with}\\ &w_j^*=\omega_j^{new}(f)+\lambda_j^{new}((\partial^\alpha P(0))_{\alpha\in\mathcal{A}})\quad\text{for all } 1\leq j\leq k.\end{aligned} \tag{2.89}$$

This is a consequence of the finite-precision version of the algorithm OPTIMIZE VIA MATRIX. We compute the numbers $b_{j\ell}$ with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$.

Recall that $w\mapsto R_w$ parametrizes the space V_P of all polynomials $R\in\mathcal{P}$ with $\partial^\alpha[R-P]\equiv 0$. Thus, using (2.86) and (2.88) we see that

$$\sum_{Q\in\mathcal{I}(Q^{\#})}\sum_{\xi\in\Xi(Q,\mathcal{A}^-)}|\xi(f,R_{w^*})|^p\leq C\cdot\Big[\sum_{Q\in\mathcal{I}(Q^{\#})}\sum_{\xi\in\Xi(Q,\mathcal{A}^-)}|\xi(f,R)|^p+\Delta_g^p\cdot|R|^p\Big],$$

for any polynomial $R\in V_P$. We can thus set $R^{\#}_{Q^{\#}}=R_{w^*}$ and we obtain the estimate (2.85).

We now produce a numerically accurate formula for the new assists ω_j^{new} ($j=1,\dots,k$) and for the functionals $\lambda_j^{new}((\partial^\alpha P(0))_{\alpha\in\mathcal{A}})$. We examine the relevant definitions.

In the expression for λ_j^{new} in (2.87), the numbers $b_{j\ell}$ are given with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$, the functionals $\check{\lambda}_\ell$ are given with parameters $(\Delta_g,\Delta_\epsilon)$, and $L\leq\Delta_g^{-C}$. Thus, we can compute the functionals λ_j^{new} with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$.

We will review our computation of a short form representation of each $\omega_j^{new}(f)$ in (2.87), following the infinite-precision text. We need to document roundoff errors at each stage of the computation, and estimate the size of the relevant numbers.

We write $\omega_j^{new}=\omega_j^{new,1}+\omega_j^{new,2}$, with $\omega_j^{new,1}$ and $\omega_j^{new,2}$ defined via (1.33) and (1.34) of [4], respectively.

We first review the computation of $\omega_j^{new,1}$:

- The numbers $c_\ell(x)$ ($x\in S_\ell\subset E$) are given with parameters $(\Delta_g,\Delta_\epsilon)$, since each λ_ℓ is given with the same parameters by assumption. The weights $d_j(x)$ are computed by evaluating the sum $d_j(x)=\sum_\ell b_{j\ell}\cdot c_\ell(x)$. Each term $b_{j\ell}\cdot c_\ell(x)$ is computed to precision $\Delta_g^{-C}\Delta_\epsilon$. Hence, each weight $d_j(x)$ is computed to precision $L\Delta_g^{-C}\Delta_\epsilon\leq CN\Delta_g^{-C}\Delta_\epsilon\leq\Delta_g^{-C'}\Delta_\epsilon$; we compute each $d_j(x)$ by sorting, just as before. Moreover, each $d_j(x)$ satisfies $|d_j(x)|\leq L\Delta_g^{-C}\leq\Delta_g^{-C'}$. Therefore, we can compute each $d_j(x)$ with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$. The bounds on the work and storage required by this computation are the same as before.

We can thus express the functionals $\omega_j^{\mathrm{new},1}$ in the form in (1.33) of [4], where the coefficients $d_j(x)$ are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. Thus, by definition, we can compute the functional $\omega_j^{\mathrm{new},1}$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

We now review the computation of $\omega_j^{\mathrm{new},2}$.

The coefficients $q_{j\omega}$ in (1.34) of [4] are computed using $q_{j\omega} = \sum_{(\ell,a)} b_{j\ell} \cdot \mu_{\ell a}$. The numbers $b_{j\ell}$ are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, the numbers $\mu_{\ell a}$ are given with parameters $(\Delta_g, \Delta_\epsilon)$, and the number of terms in the sum is bounded by $CN \le \Delta_g^{-C}$. Hence, the numbers $q_{\ell\omega}$ can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

We finally express $\omega_j^{\mathrm{new},2}$ in the form in (1.34) of [4]. As before, the coefficients $k_j(x)$, which we compute by sorting, are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

We have shown how to compute the new assists $\omega_j^{\mathrm{new}} = \omega_j^{\mathrm{new},1} + \omega_2^{\mathrm{new},j}$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. We have seen that computation follows by the same method as in infinite-precision, and by making careful note of the roundoff errors and the size of numbers involved in the computation, we verified that the computation could be carried out on our finite-precision computer.

The functionals $(f, P) \mapsto \partial^\alpha \big[R^\#_{Q^\#}(f, P)\big](0)$ can be computed in short form in terms of the assists ω_j^{new} $(j = 1, \dots, k)$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, as desired. (See (1.30) of [4].)

This completes the explanation of the algorithm MAKE NEW ASSISTS AND ASSIGN KEYSTONE JETS (finite-precision). □

For each $(f, R) \in \mathbb{X}(S_1 Q^\# \cap E) \oplus \mathcal{P}$ we set

$$\big[M^\#_{Q^\#}(f, R)\big]^p := \sum_{Q \in \mathcal{I}(Q^\#)} \sum_{\xi \in \Xi(Q, \mathcal{A}^-)} |\xi(f, R)|^p = \sum_{Q \in \mathcal{I}(Q^\#)} \big[M_{(Q,\mathcal{A}^-)}(f, R)\big]^p . \tag{2.90}$$

Let $P \in \mathcal{P}$. From the previous algorithm, we see that the polynomial $R^\# = R^\#_{Q^\#}(f, P)$ satisfies

$$\begin{cases} \partial^\alpha (R^\# - P) \equiv 0 \ \text{ for all } \alpha \in \mathcal{A}, \\ M^\#_{Q^\#}(f, R^\#) \le C \cdot \big[M^\#_{Q^\#}(f, R) + \Delta_{\mathrm{junk}} |R|\big], \end{cases} \tag{2.91}$$

for any polynomial $R \in \mathcal{P}$ such that $\partial^\alpha [R - P] \equiv 0$ for all $\alpha \in \mathcal{A}$. (Recall that $\Delta_g \le \Delta_{\mathrm{junk}}$.)

Instead of Lemma 10 of [4], we have the following result.

Lemma 3. *Let $Q^\#$ be a keystone cube. Then*

$$c \cdot \|(f, R)\|_{S_0 Q^\#} \le M^\#_{Q^\#}(f, R) \le C \cdot \big[\|(f, R)\|_{S_1 Q^\#} + \Delta_{\mathrm{junk}} |R|\big] .$$

for all $(f, R) \in \mathbb{X}(S_1 Q^\# \cap E) \oplus \mathcal{P}$. Here, $c > 0$ and $C \ge 1$ are universal constants.

Proof. We prove the first inequality $c\|(f, R)\|_{S_0 Q^\#} \le M^\#_{Q^\#}(f, R)$ by the same argument as before. We use the approximation (2.84) in place of (1.20) of [4].

To prove the second inequality, we start from (2.84), which implies

$$\left[M^{\#}_{Q^{\#}}(f,R)\right]^p = \sum_{Q\in\mathcal{I}(Q^{\#})} \left[M_{(Q,\mathcal{A}^-)}(f,R)\right]^p \le C \sum_{Q\in\mathcal{I}(Q^{\#})} \left[\|(f,R)\|^p_{\frac{65}{64}Q} + \Delta^p_{\text{junk}}|R|^p\right].$$

Since $\frac{65}{64}Q \subset S_1Q^{\#}$ and $\delta_{Q^{\#}} \le \delta_Q \le C\delta_{Q^{\#}}$ for each $Q \in \mathcal{I}(Q^{\#})$, and since $\#\left[\mathcal{I}(Q^{\#})\right] \le C$, by Lemma 14 of [3], we can estimate the above line by

$$C \cdot \left[\|(f,R)\|^p_{S_1Q^{\#}} + \Delta^p_{\text{junk}}|R|^p\right] \le C' \cdot \left[\|(f,R)\|_{S_1Q^{\#}} + \Delta_{\text{junk}}|R|\right]^p.$$

This completes the proof of Lemma 3. □

Proposition 2 of [4] requires a few changes in finite-precision. Here is the modified statement:

Proposition 1. *Let $\widehat{Q}$ be a dyadic subcube of Q°, such that $3\widehat{Q}$ is tagged with $(\mathcal{A},\epsilon)$. Assume also that $Q^{\#} \in CZ(\mathcal{A}^-)$ is a keystone cube, and that $S_1Q^{\#} \subseteq \frac{65}{64}\widehat{Q}$.*

If $H \in \mathbb{X}$, $H = f$ on $E \cap S_1Q^{\#}$, and $\partial^\alpha H(x_{Q^{\#}}) = \partial^\alpha P(x_{Q^{\#}})$ for all $\alpha \in \mathcal{A}$, then

$$\delta^{-m}_{Q^{\#}}\|H - R^{\#}_{Q^{\#}}\|_{L^p(S_1Q^{\#})} \le C \cdot \left[\|H\|_{\mathbb{X}(S_1Q^{\#})} + \Delta_{\text{junk}}|J_{x_{Q^{\#}}}H|\right]. \tag{2.92}$$

Here, $C \ge 1$ is a universal constant; and $R^{\#}_{Q^{\#}} = R^{\#}_{Q^{\#}}(f,P)$. (See the algorithm MAKE NEW ASSISTS AND ASSIGN KEYSTONE JETS.)

Proof. The argument that (1.47),(1.48) of [4] imply (1.49) of [4] is unchanged in the finite-precision setting. We restate the result here: If $\partial^\alpha R(x_{Q^{\#}}) = 0$ for all $\alpha \in \mathcal{A}$, then

$$R \in \sigma(S_0Q^{\#}) \implies |\partial^\beta R(x_{Q^{\#}})| \le W\delta^{m-n/p-|\beta|}_{Q^{\#}} \quad \text{for all } \beta \in \mathcal{M}, \tag{2.93}$$

where $W = W(m,n,p)$.

We need only examine and fix the last paragraph in the proof of Proposition 2 of [4]. We modify the text that begins after the sentence "We now prove the main assertion in Proposition 2". The revised text is as follows.

Suppose that $H \in \mathbb{X}$ satisfies $H = f$ on $E \cap S_1Q^{\#}$ and $\partial^\alpha H(x_{Q^{\#}}) = \partial^\alpha P(x_{Q^{\#}})$ for $\alpha \in \mathcal{A}$. Then $\partial^\alpha(J_{x_{Q^{\#}}}H - P) \equiv 0$ for all $\alpha \in \mathcal{A}$. (Recall, $\mathcal{A}$ is monotonic.) We apply the estimate in (2.91), and then we apply Lemma 3. Therefore,

$$\begin{aligned} M^{\#}_{Q^{\#}}(f,R^{\#}_{Q^{\#}}) &\le C\left[M^{\#}_{Q^{\#}}(f,J_{x_{Q^{\#}}}H) + \Delta_{\text{junk}}|J_{x_{Q^{\#}}}H|\right] \\ &\le C'\left[\|(f,J_{x_{Q^{\#}}}H)\|_{S_1Q^{\#}} + \Delta_{\text{junk}}|J_{x_{Q^{\#}}}H|\right] \le C''\left[\|H\|_{\mathbb{X}(S_1Q^{\#})} + \Delta_{\text{junk}}|J_{x_{Q^{\#}}}H|\right]. \end{aligned}$$

Thus,

$$M^{\#}_{Q^{\#}}(0,R^{\#}_{Q^{\#}} - J_{x_{Q^{\#}}}H) \le C\left[\|H\|_{\mathbb{X}(S_1Q^{\#})} + \Delta_{\text{junk}}|J_{x_{Q^{\#}}}H|\right].$$

Lemma 3 implies that

$$\|(0,R^{\#}_{Q^{\#}} - J_{x_{Q^{\#}}}H)\|_{S_0Q^{\#}} \le C\left[\|H\|_{\mathbb{X}(S_1Q^{\#})} + \Delta_{\text{junk}}|J_{x_{Q^{\#}}}H|\right],$$

hence

$$R_{Q^\#} - J_{x_{Q^\#}} H \in C\big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |J_{x_{Q^\#}} H|\big] \cdot \sigma(S_0 Q^\#).$$

Since $\partial^\alpha (R^\#_{Q^\#} - J_{x_{Q^\#}} H) \equiv 0$ for $\alpha \in \mathcal{A}$, we can apply (2.93) to show that

$$\big|\partial^\beta \big[J_{x_{Q^\#}} H - R^\#_{Q^\#}\big](x_{Q^\#})\big| \le C \cdot \delta_{Q^\#}^{m-n/p-|\beta|} \cdot \big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |J_{x_{Q^\#}} H|\big] \tag{2.94}$$

for all $\beta \in \mathcal{M}$.

Hence, by the Sobolev inequality we have

$$\delta_{Q^\#}^{-m} \|H - R^\#_{Q^\#}\|_{L^p(S_1 Q^\#)} \le C\big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |J_{x_{Q^\#}} H|\big]$$

This is the desired estimate. (See (2.92).) This completes the proof of Proposition 1. □

We derive a few consequences of Proposition 1. First note

$$\begin{aligned} |J_{x_{Q^\#}} H - R^\#_{Q^\#}| &= \Big(\sum_{\alpha \in \mathcal{M}} \big|\partial^\alpha (J_{x_{Q^\#}} H - R^\#_{Q^\#})(0)\big|^p \Big)^{1/p} \\ &\le C \cdot \sum_{\beta \in \mathcal{M}} |\partial^\beta (J_{x_{Q^\#}} H - R^\#_{Q^\#})(0)| \le C' \sum_{\beta \in \mathcal{M}} |\partial^\beta (J_{x_{Q^\#}} H - R^\#_{Q^\#})(x_{Q^\#})|. \end{aligned}$$

where the last inequality is due to the fact that $|x_{Q^\#}| \le C$ (see Lemma 7 of [3]). Also, recall that $\delta_{Q^\#} \le 1$. Thus, we can use (2.94) to obtain

$$|J_{x_{Q^\#}} H - R^\#_{Q^\#}| \le C''\big[\|H\|_{\mathbb{X}(S_1 Q)} + \Delta_{\text{junk}} |J_{x_{Q^\#}} H|\big].$$

Hence,

$$\begin{aligned} |J_{x_{Q^\#}} H| &\le |R^\#_{Q^\#}| + C'' \cdot \big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |J_{x_{Q^\#}} H|\big] \\ &\Longrightarrow |J_{x_{Q^\#}} H| \le 2 \cdot |R^\#_{Q^\#}| + 2C \cdot \|H\|_{\mathbb{X}(S_1 Q^\#)}, \end{aligned}$$

where we have used that $C \cdot \Delta_{\text{junk}} \le 1/2$. By using the previous estimate in (2.92), we have

$$\begin{aligned} \delta_{Q^\#}^{-m} \|H - R^\#_{Q^\#}\|_{L^p(S_1 Q^\#)} &\le C'\big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |R^\#_{Q^\#}| + \Delta_{\text{junk}} \|H\|_{\mathbb{X}(S_1 Q^\#)}\big] \\ &\le C''\big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |R^\#_{Q^\#}|\big]. \end{aligned}$$

In summary, we have proven the following result.

Lemma 4. *Under the assumptions of Proposition 1, we have*

$$\delta_{Q^\#}^{-m} \|H - R^\#_{Q^\#}\|_{L^p(S_1 Q^\#)} \le C''\big[\|H\|_{\mathbb{X}(S_1 Q^\#)} + \Delta_{\text{junk}} |R^\#_{Q^\#}|\big]. \tag{2.95}$$

We will prove one more lemma before returning to the main line of our argument.

Lemma 5. *Under the assumptions of Proposition 1, we have*

$$|R^\# - P| \le C \cdot \big[\|(f, P)\|_{S_1 Q^\#} + \Delta_{\text{junk}} |P|\big]$$

where $R^\# = R^\#_{Q^\#}(f, P)$ *for a keystone cube* $Q^\#$.

Proof. Let $Q^{\#}$ be a keystone cube, and denote $R^{\#} = R^{\#}_{Q^{\#}}(f, P)$.

Since $\sigma(S_0 Q^{\#})$ is the unit ball of the norm $\|(0,\cdot)\|_{S_0 Q^{\#}}$ on $\mathcal{P}$, we can use (2.93) to show the following: If $\widetilde{R} \in \mathcal{P}$ satisfies $\partial^{\alpha}\widetilde{R} \equiv 0$ for all $\alpha \in \mathcal{A}$, then for any $\beta \in \mathcal{M}$ we have

$$|\partial^{\beta}\widetilde{R}(x_{Q^{\#}})| \leq C \cdot (\delta_{Q^{\#}})^{m-n/p-|\beta|} \cdot \|(0,\widetilde{R})\|_{S_0 Q^{\#}} \leq C \cdot \|(0,\widetilde{R})\|_{S_0 Q^{\#}}.$$

(Here, we have used that $\delta_{Q^{\#}} \leq 1$.) From the above estimate and Lemma 3, we deduce that

$$|\widetilde{R}| = \Big(\sum_{\alpha \in \mathcal{M}} \Big|\partial^{\alpha}\widetilde{R}(0)\Big|^p \Big)^{1/p} \leq C \cdot \sum_{\beta \in \mathcal{M}} \big|\partial^{\beta}\widetilde{R}(x_{Q^{\#}})\big| \leq C' \cdot M^{\#}_{Q^{\#}}(0,\widetilde{R}).$$

Taking $R = P$ in (2.91), we see that the polynomial $R^{\#} = R^{\#}_{Q^{\#}}(f, P)$ satisfies

$$\begin{cases} \partial^{\alpha}(R^{\#} - P) \equiv 0 \quad \text{for all } \alpha \in \mathcal{A} \\ M^{\#}_{Q^{\#}}(f, R^{\#}) \leq C \cdot \big[M^{\#}_{Q^{\#}}(f, P) + \Delta_{\text{junk}} \cdot |P|\big]. \end{cases}$$

Thus, the $\mathcal{A}$-derivatives of $\widetilde{R} = R^{\#} - P$ vanish, so we may apply the previous estimate to give

$$\begin{aligned} |R^{\#} - P| &\leq C' \cdot M^{\#}_{Q^{\#}}(0, R^{\#} - P) \leq C'' \cdot \big[M^{\#}_{Q^{\#}}(f, R^{\#}) + M^{\#}_{Q^{\#}}(f, P)\big] \\ &\leq C''' \cdot \big[M^{\#}_{Q^{\#}}(f, P) + \Delta_{\text{junk}}|P|\big] \leq \overline{C} \cdot \big[\|(f, P)\|_{S_1 Q^{\#}} + \Delta_{\text{junk}}|P|\big]. \end{aligned}$$

In the last estimate, we use Lemma 3. This completes the proof of Lemma 5. □

In Section 1.4.2 of [4], all of the marked cubes are assumed to be $\widetilde{S}$-bit machine cubes, with $\widetilde{S} \leq C\overline{S}$. All the functionals are to be given in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. This concludes the description of the changes to Section 1.4.2 of [4].

Recall the notion of a testing cube (see Section 1.4.3 in [4]): A dyadic cube $\widehat{Q} \subset Q^{\circ}$ is a testing cube if it can be written as a disjoint union of cubes in $CZ(\mathcal{A}^-)$.

Remark 1. Recall that each cube Q in $CZ(\mathcal{A}^-)$ satisfies $\delta_Q \geq c \cdot \Delta_0$ with $\Delta_0 = 2^{-\overline{S}}$ for a universal constant $c > 0$, by the Main Technical Results for $\mathcal{A}^-$. Hence, every testing cube $\widehat{Q}$ has $\widetilde{S}$-bit machine points as corners, where $\widetilde{S} \leq C\overline{S}$ for a universal constant C.

Recall, in (1.62) of [4], we introduce a parameter t_G, which is an integer power of 2. Furthermore, we assume that

$$t_G = 2^{-\widetilde{S}} \text{ for an integer } \widetilde{S} \text{ with } 1 \leq \widetilde{S} \leq C\overline{S}. \tag{2.96}$$

In finite-precision, we must make a slight change to Lemma 11 of [4]. We will need to assume that the constant a_{new} is picked to satisfy

$$a_{\text{new}} = 2^{-\widetilde{S}} \text{ for an integer } \widetilde{S} \text{ with } 1 \leq \widetilde{S} \leq C\overline{S}. \tag{2.97}$$

To see that this is possible, we examine the proof of the lemma. Observe that it suffices to choose $a_{new} = a \cdot t_G/512$, where $a = a(\mathcal{A}^-)$ is as in the Main Technical Results for $\mathcal{A}^-$. Recall that a is a universal constant and an integer power of 2. Because t_G satisfies (2.96), the constant a_{new} satisfies (2.97).

We are finished describing the changes required in Section 1.4.3 of [4].

2.13.1. Testing functionals. We continue on to Section 1.4.4 of [4].

Recall that Δ_{new} is a machine number that satisfies (2.82). We will make use of the condition

$$\Delta_{new} \le c(\epsilon, t_G), \tag{2.98}$$

where $c(\epsilon, t_G) \in (0,1)$ is a small constant depending on m, n, p, t_G, and ϵ. We later choose ϵ and t_G to depend only on m, n, and p - hence, (2.98) is consistent with our previous assumptions (2.74), (2.81), and (2.82).

We assume we are given a testing cube $\widehat{Q} \subset Q^\circ$.

For $Q \in CZ(\mathcal{A}^-)$ with $Q \subset (1 + 100 t_G)\widehat{Q}$, we define the map

$$(f, P) \mapsto R_Q^{\widehat{Q}} := \begin{cases} P, & \delta_Q \ge t_G \delta_{\widehat{Q}}, \\ R^{\#}_{\mathcal{K}(Q)}(f, P), & \delta_Q < t_G \delta_{\widehat{Q}}, \end{cases} \tag{2.99}$$

for any $(f, P) \in \mathbb{X}(\frac{65}{64}\widehat{Q} \cap E) \oplus \mathcal{P}$. Recall that the mapping $R^{\#}_{Q^\#} : \mathbb{X}(S_1 Q^\# \cap E) \oplus \mathcal{P} \to \mathcal{P}$ is defined in the algorithm MAKE NEW ASSISTS AND ASSIGN KEYSTONE JETS (finite-precision version); see Section 2.13. Recall that the mapping $Q \mapsto \mathcal{K}(Q)$ satisfies $S_1 \mathcal{K}(Q) \subset CQ$; hence, if $\delta_Q < t_G \delta_{\widehat{Q}}$ and t_G is sufficiently small, then $S_1 \mathcal{K}(Q) \subset \frac{65}{64}\widehat{Q}$ (see the proof of (1.64) in [4] for more details), and hence the mapping $R_Q^{\widehat{Q}}$ is well-defined on $\mathbb{X}(\frac{65}{64}\widehat{Q} \cap E) \oplus \mathcal{P}$.

- Recalling the precision with which we compute the maps $R^{\#}_{Q^\#}$, we see that $(f, P) \mapsto \partial^\alpha \big[R_Q^{\widehat{Q}}(f, P)\big](0)$ has the form

$$\sum_{x \in E} \lambda_x f(x) + \sum_{\omega} \mu_\omega \, \omega(f) + \sum_{\beta \in \mathcal{M}} \theta_\beta \cdot \frac{1}{\beta!} \partial^\beta P(0), \tag{2.100}$$

  where the possibly nonzero coefficients $\lambda_x, \mu_\omega, \theta_\beta$ are computed with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$. The number of possibly nonzero coefficients is at most a universal constant.

- Recalling the precision with which we compute each ξ in $\Xi(Q, \mathcal{A}^-)$, we see that $(f, P) \mapsto \xi(f, R_Q^{\widehat{Q}})$ has the form

$$\sum_{x \in E} \widetilde{\lambda}_x f(x) + \sum_{\omega} \widetilde{\mu}_\omega \, \omega(f) + \sum_{\beta \in \mathcal{M}} \widetilde{\theta}_\beta \cdot \frac{1}{\beta!} \partial^\beta P(0), \tag{2.101}$$

  where the possibly nonzero coefficients $\widetilde{\lambda}_x, \widetilde{\mu}_\omega, \widetilde{\theta}_\beta$ are computed with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$. The number of possibly nonzero coefficients is at most a universal constant.

We will need to modify the definition of $M_{\widehat{Q}}(f, P)$ in (1.65)–(1.68) of [4].

We define $\left[M_{\widehat{Q}}(f, P)\right]^p$ to be the sum of the terms **(I)**–**(IV)** (see (1.65)–(1.68) of [4]) plus the sum of the terms

$$\textbf{(V)} = \Delta_{\text{new}}^{2p} \sum_{x \in \frac{65}{64}\widehat{Q} \cap E} |f(x) - P(x)|^p, \quad \text{and} \tag{2.102}$$

$$\textbf{(VI)} = \Delta_{\text{new}}^{p} |P|^p = \Delta_{\text{new}}^{p} \sum_{\beta \in \mathcal{M}} |\partial^\beta P(0)|^p.$$

Recall that Δ_{new} is a machine number satisfying (2.82).

Each of the linear functionals arising in **(I)**–**(IV)** and in **(V)**–**(VI)**, will be computed with precision $(\Delta_g^C, \Delta_\epsilon\, \Delta_g^{-C})$. In Section 2.13.2, we give further explanation of this remark, and we analyze the work required to compute all the functionals.

As in the infinite-precision text, we define

$$\overline{\overline{\sigma}}(\widehat{Q}) := \left\{P \in \mathcal{P} : M_{\widehat{Q}}(0, P) \le 1\right\}.$$

We replace the algorithm APPROXIMATE NEW TRACE NORM from the infinite-precision text with the finite-precision version below.

APPROXIMATE NEW TRACE NORM (FINITE-PRECISION)

We are given a machine number $t_G > 0$ as in (2.96).

We perform one-time work at most $C(t_G) N \log N$ in space $C(t_G) N$, after which we can answer queries as follows.

A query consists of a testing cube $\widehat{Q}$. The response to the query $\widehat{Q}$ is a list $\mu_1^{\widehat{Q}}, \ldots, \mu_D^{\widehat{Q}}$ of linear functionals on $\mathcal{P}$ such that

$$c(t_G) \cdot \sum_{i=1}^{D} |\mu_i^{\widehat{Q}}(P)|^p \le \left[M_{\widehat{Q}}(0, P)\right]^p \le C(t_G) \cdot \Big[\sum_{i=1}^{D} |\mu_i^{\widehat{Q}}(P)|^p\Big]. \tag{2.103}$$

The functionals $\mu_1^{\widehat{Q}}, \ldots, \mu_D^{\widehat{Q}}$ have the form

$$P \mapsto \sum_{\beta \in \mathcal{M}} \text{coeff}_\beta \frac{1}{\beta!} \partial^\beta P(0)$$

where the coefficients coeff_β are computed with parameters $(\Delta_g^C, \Delta_g^{-C} \Delta_\epsilon)$.

We define a quadratic form on $\mathcal{P}$ by

$$q_{\widehat{Q}}(P) := \sum_{i=1}^{D} |\mu_i^{\widehat{Q}}(P)|^2. \tag{2.104}$$

This quadratic form satisfies

$$c(t_G) \cdot \left[M_{\widehat{Q}}(0, P)\right]^2 \le q_{\widehat{Q}}(P) \le C(t_G) \cdot \left[M_{\widehat{Q}}(0, P)\right]^2. \tag{2.105}$$

In particular,

$$\{q_{\widehat{Q}} \le c(t_G)\} \subset \overline{\overline{\sigma}}(\widehat{Q}) \subset \{q_{\widehat{Q}} \le C(t_G)\}. \tag{2.106}$$

The quadratic form $q_{\widehat{Q}}$ is given in the form

$$q_{\widehat{Q}}(P) = \sum_{\alpha,\beta\in\mathcal{M}} q_{\alpha\beta}\cdot\frac{1}{\alpha!}\partial^{\alpha}P(0)\cdot\frac{1}{\beta!}\partial^{\beta}P(0).$$

The $q_{\alpha\beta}$ form a symmetric matrix. For each $\alpha,\beta\in\mathcal{M}$ we compute the number $q_{\alpha\beta}$ with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$.

The work required to answer a query is at most $C(t_G)\log N$.

Here, $c(t_g)>0$ and $C(t_g)\geq 1$ are constants depending on t_G, m, n, and p.

Explanation. The explanation is more or less the same as that in [4]. We make sure to use the finite-precision versions of the algorithms APPROXIMATE OLD TRACE NORM, COMPUTE NORMS FROM MARKED CUBOIDS, and COMPRESS NORMS, where before we used the infinite-precision versions. Below, we make a few additional comments on discrepancies that arise.

Given a keystone cube $Q^{\#}$, the polynomial map $P\mapsto R^{\#}_{Q^{\#}}(0,P)$ in (1.74) of [4] is given with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$; indeed, the functionals $\overline{\lambda}_{(Q^{\#},\beta)}$ are given with such parameters, as in the statement of the algorithm MAKE NEW ASSISTS AND ASSIGN KEYSTONE JETS (finite-precision).

We apply a marking procedure, just as in the explanation in [4]. We provide details below.

- For each $Q\in CZ_{\mathrm{main}}(\mathcal{A}^-)$ and $i=1,\dots,D$, we mark Q with the functional
$$\xi_{(Q,i)}(P)=\xi_i^Q(R^{\#}_{\mathcal{K}(Q)}(0,P)).$$
Note that each functional $R\mapsto\xi_i^Q(R)$ is given with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$; see the statement of the algorithm APPROXIMATE OLD TRACE NORM (finite-precision). Also, the polynomial map $P\mapsto R^{\#}_{\mathcal{K}(Q)}(0,P)$ is given with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$. We can stably compute the composition of a linear functional on $\mathbb{R}^D$ with a linear map on $\mathbb{R}^D$, hence we can compute each $\xi_{(Q,i)}$ with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$, for a possibly larger constant C.
- For each $(Q',Q'')\in BD(\mathcal{A}^-)$ and $\beta\in\mathcal{M}$, we mark Q' with the functional
$$\xi_{(Q',Q'',\beta)}(P)=\delta_{Q'}^{n/p-m+|\beta|}\cdot\partial^{\beta}\{R^{\#}_{\mathcal{K}(Q')}(0,P)-R^{\#}_{\mathcal{K}(Q'')}(0,P)\}(x_{Q'}).$$
This functional is given with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$ because the polynomial maps $P\mapsto R^{\#}_{\mathcal{K}(Q')}(0,P)$ and $P\mapsto R^{\#}_{\mathcal{K}(Q'')}(0,P)$ are given with the same parameters, because $\delta_{Q'}$ is a machine number with $c\cdot\Delta_0\leq\delta_{Q'}\leq 1$ (this is a consequence of the Main Technical Results for $\mathcal{A}^-$; see (2.75)), and because $x_{Q'}$ is an $\widetilde{S}$-bit machine point with $\widetilde{S}\leq C\overline{S}$.

Each of the functionals ξ that is associated to a marked cube has the form

$$P\mapsto\xi(P)=\sum_{\beta\in\mathcal{M}}\mathrm{coeff}_\beta\cdot\partial^{\beta}P(0),$$

where the coefficients coeff_β are specified with parameters $(\Delta_g^C,\Delta_g^{-C}\Delta_\epsilon)$. We perform one-time work for the algorithm COMPUTE NORMS FROM MARKED CUBES

(finite-precision) on the marked cubes described in the above bullet points. (See **Modification 6** in Section 2.9.) All of the marked cubes belong to $\mathrm{CZ}(\mathcal{A}^-)$. By assumption, all cubes in $\mathrm{CZ}(\mathcal{A}^-)$ have sidelength in the interval $[c \cdot \Delta_0, 1]$, where $\Delta_0 = 2^{-\overline{S}}$. Hence, the marked cubes have $\widetilde{S}$-bit machine points as corners, where $\widetilde{S} \leq C\overline{S}$. Thus, the finite-precision version of the algorithm applies.

This concludes the description of the one-time work. Next, we explain the query work for the algorithm APPROXIMATE NEW TRACE NORM.

We are given a testing cube $\widehat{Q}$. As in in the infinite-precision text, we partition $(1 + t_G)\widehat{Q}$ into dyadic cubes $Q_1, \ldots, Q_L \subset \mathbb{R}^n$ such that $\delta_{Q_\ell} = (t_G/4)\delta_{\widehat{Q}}$. Hence, $\delta_{Q_\ell} \geq 2^{-C\overline{S}}$ for a universal constant C, for each $\ell = 1, \ldots, L$, thanks to (2.96) and Remark 1. Consequently, each Q_ℓ is a $\widetilde{S}$-bit machine cube with $\widetilde{S} \leq C\overline{S}$. Moreover, note that $L \leq C(t_G)$.

We apply the query algorithm in COMPUTE NORMS FROM MARKED CUBOIDS (finite-precision) with $\Delta = \Delta_g$ for each cube Q_ℓ $(1 \leq \ell \leq L)$. We refer the reader to **Modification 6** in Section 2.9 for a statement of the algorithm. According to this, with work at most $C(t_G)\log N$ we compute linear functionals $\mu_1^{Q_\ell}, \ldots, \mu_D^{Q_\ell}$ such that

$$c\sum_{k=1}^{D} |\mu_k^{Q_\ell}(P)|^p \leq \sum_{\substack{Q \in \mathrm{CZ}(\mathcal{A}^-) \\ \text{lin. func. } \xi}} \big\{ |\xi(P)|^p : Q \subset Q_\ell,\ Q \text{ marked with } \xi \big\} + \Delta_g^{p/2}|P|^p$$

$$\leq C\Big[\sum_{k=1}^{D} |\mu_k^{Q_\ell}(P)|^p + \Delta_g^{p/2}|P|^p \Big], \tag{2.107}$$

where each $\mu_k^{Q_\ell}$ is given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. Here, we use the estimate $\Delta_g^p \Delta_0^{-C} \log(\Delta_g^{-1}) \leq \Delta_g^{p/2}$, which follows because $\Delta_g \ll \Delta_0$.

We sum (2.107) from $\ell = 1, \ldots, L$. The sum of the junk terms is equal to $L\Delta_g^{p/2}|P|^p \leq C(t_G)\Delta_g^{p/2}|P|^p$. Hence, as in infinite-precision, in analogy with (1.77) of [4], we obtain the estimate

$$c(t_G) \sum_{\ell=1}^{L} \sum_{k=1}^{D} |\mu_k^{Q_\ell}(P)|^p \leq [\mathfrak{S}_1 + \mathfrak{S}_2] + \Delta_g^{p/2}|P|^p$$

$$\leq C(t_G)\Big[\sum_{\ell=1}^{L} \sum_{k=1}^{D} |\mu_k^{Q_\ell}(P)|^p + \Delta_g^{p/2}|P|^p \Big]. \tag{2.108}$$

For the definition of the terms $\mathfrak{S}_1$ and $\mathfrak{S}_2$, see (1.77) in [4].

To compute a list of the functionals in $(\mathbf{F_1})$–$(\mathbf{F_6})$, we follow the explanation in [4] that proceeds (1.77). In the above text, we have described how to compute the functionals in $(\mathbf{F_1})$–$(\mathbf{F_4})$; all these functional are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. Additionally, the maps $P \mapsto R_Q^{\widehat{Q}}(0, P)$ are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$; see (2.99). Hence, we can compute the functionals in $(\mathbf{F_5})$ and $(\mathbf{F_6})$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

Therefore, the functionals listed in $(\mathbf{F_1})$–$(\mathbf{F_6})$ are given in the form

$$P \mapsto \sum_{\beta \in \mathcal{M}} d_\beta \cdot \frac{1}{\beta!} \partial^\beta P(0),$$

where the numbers d_β are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

In addition, we consider the functionals

$$(\mathbf{F_7}) \qquad \boxed{\lambda_\beta(P) := \Delta_{new} \cdot \partial^\beta P(0)} \qquad \text{for } \beta \in \mathcal{M}.$$

Recall that $\Delta_{new} \leq 1$ is a machine number. Thus, the functionals in $(\mathbf{F_7})$ are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

We define $[X(P)]^p$ to be the sum of p-th powers of all functionals in $(\mathbf{F_1})$–$(\mathbf{F_7})$.

Note that $\sum_{\beta \in \mathcal{M}} |\lambda_\beta(P)|^p = \Delta_{new}^p |P|^p =$ **(VI)** (see (2.102)). Hence, we have

$$[X(P)]^p = [X_{old}(P)]^p + \mathbf{(VI)},$$

where $[X_{old}(P)]^p$ is the sum of p-th powers of all functionals in $(\mathbf{F_1})$–$(\mathbf{F_6})$.

We have

$$\begin{aligned} (2.109) \qquad c(t_G) \cdot [X_{old}(P)]^p &\leq [\text{sum of terms } \mathbf{(I)}\text{–}\mathbf{(IV)} \text{ with } f \equiv 0] + \Delta_g^{p/2} |P|^p \\ &\leq C(t_G) \cdot \left[[X_{old}(P)]^p + \Delta_g^{p/2} |P|^p \right]. \end{aligned}$$

To obtain the above estimate, we reason as in the two paragraphs containing and following (1.78) of [4], making sure to use estimate (2.108) in place of (1.77) of [4]. (Recall that X_{old} here corresponds to X in the notation of [4].)

Consider the term **(V)** (see (2.102)) that arises in the definition of $M_{\widehat{Q}}(0, P)$. When $f \equiv 0$ in **(V)**, we have

$$\mathbf{(V)}|_{f=0} = \Delta_{new}^{2p} \sum_{x \in \frac{65}{64}\widehat{Q} \cap E} |P(x)|^p \leq C\, \Delta_{new}^{2p} N \sum_{\beta \in \mathcal{M}} \left|\partial^\beta P(0)\right|^p \leq \Delta_{new}^p \sum_{\beta \in \mathcal{M}} \left|\partial^\beta P(0)\right|^p.$$

Here, we use the estimates $N \leq \Delta_0^{-n} \leq \Delta_{new}^{-p/10}$ (see (2.71) and (2.82)) and $C\,\Delta_{new}^{19p/10} \leq \Delta_{new}^p$ (see (2.82)). Hence, we have $\mathbf{(V)}|_{f=0} \leq$ **(VI)**. Thus, up to constant factors, $\left[M_{\widehat{Q}}(0, P)\right]^p$ is equivalent to the sum of the terms **(I)**–**(IV)** and **(VI)** (with $f \equiv 0$).

Therefore, by adding the term $\mathbf{(VI)} + \Delta_g^{p/2}|P|^p$ to the chain of inequalities (2.109), we learn that

$$\begin{aligned} c(t_G) \cdot \left\{ \left[X_{old}(P)\right]^p + \mathbf{(VI)} + \Delta_g^{p/2}|P|^p \right\} &\leq \left[M_{\widehat{Q}}(0, P)\right]^p + \Delta_g^{p/2} |P|^p \\ &\leq C(t_G) \cdot \left\{ [X_{old}(P)]^p + \mathbf{(VI)} + \Delta_g^{p/2} |P|^p \right\}. \end{aligned}$$

Note that the middle term above is comparable to $\left[M_{\widehat{Q}}(0, P)\right]^p$, since

$$\left[M_{\widehat{Q}}(0, P)\right]^p \geq \mathbf{(VI)} = \Delta_{new}^p |P|^p \geq \Delta_g^{p/2} \cdot |P|^p.$$

Here, we use that $\Delta_g \le \Delta_{\text{new}}^2$ (see (2.82)). Since

$$[X(P)]^p = [X_{\text{old}}(P)]^p + \mathbf{(VI)},$$

we conclude that

$$c(t_G)\cdot\left\{\left[X(P)\right]^p + \Delta_g^{p/2}\,|P|^p\right\} \le \left[M_{\widehat{Q}}(0,P)\right]^p \le C(t_G)\cdot\left\{\left[X(P)\right]^p + \Delta_g^{p/2}\,|P|^p\right\}.$$

Recall that $\left[X(P)\right]^p$ is the sum of the pth powers of the functionals in $(\mathbf{F_1})$–$(\mathbf{F_7})$. Processing the functionals in $(\mathbf{F_1})$–$(\mathbf{F_7})$ using COMPRESS NORMS (finite-precision), we compute functionals $\mu_1^{\widehat{Q}},\dots,\mu_D^{\widehat{Q}}$ on $\mathcal{P}$ such that

$$c\cdot\sum_{i=1}^{D}|\mu_i^{\widehat{Q}}(P)|^p \le \left[X(P)\right]^p + \Delta_g^{p/2}\,|P|^p \le C\cdot\sum_{i=1}^{D}|\mu_i^{\widehat{Q}}(P)|^p.$$

The functionals $\mu_i^{\widehat{Q}}$ are given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

The previous two estimates establish (2.103).

Moreover, properties (2.105) and (2.106) are immediate from the definition of $q_{\widehat{Q}}$ in (2.104) and the equivalence of the ℓ_p and ℓ_2 norms on a finite-dimensional space. Each of the $\mu_i^{\widehat{Q}}$ is given with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, hence the coefficients $q_{\alpha\beta}$ of the quadratic form $q_{\widehat{Q}}$ can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ (for a possibly larger constant C).

This concludes the explanation of the query algorithm. It is easy to check that the query work at most $C(t_G)\log N$. □

2.13.2. Supporting data. We assume we are given a testing cube $\widehat{Q} \subset Q^\circ$.

We explain the main modifications to Section 1.4.5 of [4] needed here.

• **Modification 1.** As part of the supporting data for $\widehat{Q}$, we include a list of all the points $x \in \frac{65}{64}\widehat{Q} \cap E$, in addition to all the other data, namely (SD1)–(SD5). We call this the *modified supporting data* for $\widehat{Q}$.

The list $\Omega(\widehat{Q})$ of the new assist functionals is defined as in (1.79) of [4].

• **Modification 2.** The algorithm COMPUTE NEW ASSISTS operates as follows. Given a testing cube $\widehat{Q}$, and given the supporting data for $\widehat{Q}$, we compute a list of all the functionals in $\Omega(\widehat{Q})$. We compute a short form of each $\omega \in \Omega(\widehat{Q})$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

• **Modification 3.** We make only minor changes to the algorithm COMPUTE SUPPORTING MAP. The linear maps $R_Q^{\widehat{Q}}$ are to be computed in short form in terms of the assists $\Omega(\widehat{Q})$ with precision $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. The explanation of the algorithm is unchanged.

• **Modification 4.** The algorithm COMPUTE NEW ASSISTED FUNCTIONALS is replaced with the finite-precision version below.

ALGORITHM: COMPUTE NEW ASSISTED FUNCTIONALS (FINITE-PRECISION)

Given a testing cube $\widehat{Q}$ and its modified supporting data, we define

$$\mathfrak{W}_2^{\mathrm{fin}}(\widehat{Q}) := \mathfrak{W}_2(\widehat{Q}) + C(t_G) \cdot \#\Big(\frac{65}{64}\widehat{Q} \cap E\Big) \tag{2.110}$$

and

$$\mathfrak{S}_2^{\mathrm{fin}}(\widehat{Q}) := \mathfrak{S}_2(\widehat{Q}) + C(t_G) \cdot \#\Big(\frac{65}{64}\widehat{Q} \cap E\Big), \tag{2.111}$$

where $\mathfrak{W}_2(\widehat{Q})$, $\mathfrak{S}_2(\widehat{Q})$ are defined as in (1.83), (1.84) of [4], respectively.

We compute a list $\Xi(\widehat{Q})$ of functionals on $\mathbb{X}(E) \oplus \mathcal{P}$, such that

$$\big[M_{\widehat{Q}}(f,P)\big]^p = \sum_{\xi \in \Xi(\widehat{Q})} |\xi(f,P)|^p.$$

Each functional ξ in $\Xi(\widehat{Q})$ is given in short form in terms of assists $\Omega(\widehat{Q})$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

This computation requires work at most $\mathfrak{W}_2^{\mathrm{fin}}(\widehat{Q})$ in space $\mathfrak{S}_2^{\mathrm{fin}}(\widehat{Q})$.

Explanation. We include in the list $\Xi(\widehat{Q})$ all the same functionals as before, namely, the "assisted functionals" in (1.85)–(1.88) of [4], as well as additional functionals described in the next paragraph. Each "assisted functional" is given in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of the assists $\Omega(\widehat{Q})$. That is because all the functionals ξ and the maps $R_Q^{\widehat{Q}}$, $R_{Q'}^{\widehat{Q}}$, $R_{Q''}^{\widehat{Q}}$, $R_{Q_{sp}}^{\widehat{Q}}$ that are relevant to (1.85)–(1.88) of [4], have been computed in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. See (2.100) and (2.101).

We also include in the list $\Xi(\widehat{Q})$ the additional functionals

$$\lambda_x(f,P) := \Delta_{\mathrm{new}}^2 \cdot (f(x) - P(x)) \quad \text{for each } x \in \frac{65}{64}\widehat{Q} \cap E$$

and

$$\lambda_\beta(f,P) := \Delta_{\mathrm{new}} \cdot \partial^\beta P(0) \quad \text{for each } \beta \in \mathcal{M}.$$

These are the new functionals needed in finite-precision. That completes the definition of $\widehat{\Xi}(\widehat{Q})$. Note that $\Delta_{\mathrm{new}} \le 1 \le \Delta_g^{-1}$. Hence, each functional λ_x and λ_β can be expressed in short form (without assists) using coefficients that are bounded in magnitude by Δ_g^{-C}. Moreover, each coefficient can be computed to precision Δ_ϵ, which is within the precision available to our computer. (Recall: the computer works with precision $\Delta_{\min} \ll \Delta_\epsilon$.) Hence, the functionals λ_x and λ_β can be computed in short form with parameters $(\Delta_g^C, \Delta_\epsilon)$ (without assists).

The sum of $|\xi(f,P)|^p$ over all ξ in $\Xi(\widehat{Q})$ is equal to $\big[M_{\widehat{Q}}(f,P)\big]^p$, by definition.

The additional term $\#(\frac{65}{64}\widehat{Q} \cap E)$ in (2.110) and (2.111) accounts for the additional work and space, respectively, needed to compute the functionals λ_x, λ_β.

This completes the explanation of the algorithm. □

As before, given a testing cube $\widehat{Q}$, the covering cubes are defined by

$$\mathcal{I}_{\text{cov}}(\widehat{Q}) := \{Q \in CZ(\mathcal{A}^-) : Q \subset (1 + t_G)\widehat{Q}\}.$$

As before, we define a family of cutoff functions $\theta_Q^{\widehat{Q}}$ (for $Q \in \mathcal{I}_{\text{cov}}(\widehat{Q})$) that satisfy (1.92)–(1.94) of [4].

The finite-precision version of the algorithm COMPUTE POU is as follows.

COMPUTE POU (FINITE-PRECISION)

After one-time work at most $CN \log N$ in space CN, we can answer queries as follows.

A query consists of a testing cube $\widehat{Q}$ and an S-bit machine point $\underline{x} \in Q^\circ$.

Notice that $\widehat{Q}$ has $\widetilde{S}$-bit machine points as corners, with $\widetilde{S} \le C\overline{S}$ for a universal constant C, so we can safely process $\widehat{Q}$ on our finite-precision computer. (See Remark 1.)

We respond to the query with a list of all the cubes $Q_1, \dots, Q_L \in \mathcal{I}_{\text{cov}}(\widehat{Q})$ (with $Q_1, \dots, Q_L$ all distinct) such that $\underline{x} \in \frac{65}{64} Q_\ell$. Futhermore, we compute the numbers $\frac{1}{\alpha!}\partial^\alpha J_{\underline{x}} \theta_{Q_\ell}^{\widehat{Q}}(0)$ (for all $\ell = 1, \dots, L$ and $\alpha \in \mathcal{M}$) with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

To answer a query requires work and storage at most $C \log N$.

Explanation. The explanation is the same as in the infinite-precision case. □

The definition of the local extension operator $T_{\widehat{Q}}(f, P)$ given in (1.95)–(1.98) of [4] is unchanged. As before, COMPUTE POU, COMPUTE SUPPORTING MAP, and the Main Technical Results for $\mathcal{A}^-$ (finite-precision versions) yield the algorithm COMPUTE NEW EXTENSION OPERATOR (finite-precision), with the following modification:

• **Modification 5:** In finite-precision, we assume $\underline{x} \in Q^\circ$ is an S-bit machine point and $\beta \in \mathcal{M}$. We compute the functional $(f, P) \mapsto \partial^\beta(J_{\underline{x}} T_{\widehat{Q}}(f, P))(0)$ which has the form

$$\sum_{\ell=1}^{L} \gamma_\ell \cdot \omega_\ell(f) + \sum_{j=1}^{J} \lambda_j \cdot f(x_j) + \sum_{\gamma \in \mathcal{M}} \theta_\gamma \cdot \frac{1}{\gamma!} \partial^\gamma P(0),$$

where each ω_ℓ is in $\Omega(\widehat{Q})$ and each x_j is in $E \cap \frac{65}{64}\widehat{Q}$; the real numbers γ_ℓ, λ_j, and θ_γ are computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$; and $L+J+\#(\mathcal{M}) \le C$, for a universal constant C. Namely, we compute the functional $(f, P) \mapsto \partial^\beta(J_{\underline{x}} T_{\widehat{Q}}(f, P))(0)$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of assists $\Omega(\widehat{Q})$. The explanation of the algorithm is the same as before.

2.14. Inequalities for testing functionals

Let $a_{\text{new}} = a_{\text{new}}(t_G)$ be the constant from Lemma 11 of [4]. We recall that $a_{\text{new}} = 2^{-\widetilde{S}}$ for an integer $\widetilde{S}$ with $1 \le \widetilde{S} \le C\overline{S}$. (See (2.97).)

First, in Proposition 3 of [4], we state and prove some properties of the extension operator $(f, P) \mapsto T_{\widehat{Q}}(f, P)$ defined in (1.98) of [4]. The assertion and proof of the proposition are unchanged in the current setting.

Next, we prove a few estimates that show that the testing functional $M_{\widehat{Q}}(f,P)$ well-approximates the trace seminorm associated to a dilate of the testing cube $\widehat{Q}$. Such estimates were stated before in Proposition 4 of [4] (the conditional and unconditional inequalities). In the present setting, the statement and proof of the corresponding estimates will need to be modified. The next result contains the relevant estimates.

Proposition 2. *Let $\widehat{Q}$ be a testing cube, and let $(f,P) \in \mathbb{X}(\frac{65}{64}\widehat{Q} \cap E) \oplus \mathcal{P}$. Then the following estimates hold.*

(Unconditional inequality) $\|(f,P)\|_{(1+a_{new})\widehat{Q}} \leq C(t_G) \cdot M_{\widehat{Q}}(f,P)$.

(Conditional inequality) *If $3\widehat{Q}$ is tagged with $(\mathcal{A},\epsilon)$, then*

$$M_{\widehat{Q}}(f,P) \leq C(t_G) \cdot (1/\epsilon) \cdot \left[\|(f,P)\|_{\frac{65}{64}\widehat{Q}} + \Delta_{new} \cdot |P| \right].$$

As in [4], the unconditional inequality is a direct consequence of Proposition 3 of [4] (which still holds in the present setting).

To prove the conditional inequality, we first state and prove a finite-precision version of Lemma 12 of [4]:

Lemma 6. *Suppose that the testing cube $\widehat{Q}$ is η-simple for some $\eta \geq t_G$. Then*

$$M_{\widehat{Q}}(f,P) \leq C(t_G) \cdot \left[\|(f,P)\|_{\frac{65}{64}\widehat{Q}} + \Delta_{new} \cdot |P| \right],$$

where $C(t_G)$ depends only on m, n, p, and t_G.

Proof. The proof is more or less the same as the proof of Lemma 12 in [4]. If $\widehat{Q}$ is η-simple ($\eta \geq t_G$), then the terms **(II)**, **(III)**, **(IV)** vanish, as explained in the outset of the proof of the lemma. This leaves us with the original term **(I)**, and the new terms **(V)**, and **(VI)**. Recall, the definition of **(I)** is given in equation (1.65) of [4], and the definitions of **(V)** and **(VI)** are given in (2.102).

An arbitrary summand in the term **(I)** has the form $\left[M_{(Q,\mathcal{A}^-)}(f,P)\right]^p$, for $Q \in CZ_{\text{main}}(\mathcal{A}^-)$ with $Q \subset (1+t_G)\widehat{Q}$. From (2.84), we have

$$\left[M_{(Q,\mathcal{A}^-)}(f,P)\right]^p \leq C \cdot \left[\|(f,P)\|^p_{\frac{65}{64}Q} + \Delta^p_{\text{junk}}|P|^p \right].$$

From Lemma 11 of [4] and $Q \subset (1+t_G)\widehat{Q}$ we conclude that $\frac{65}{64}Q \subset \frac{65}{64}\widehat{Q}$. Due to the fact that $\widehat{Q}$ is η-simple with $\eta \geq t_G$, we know $\delta_Q \geq t_G \delta_{\widehat{Q}}$. From Lemma 5 of [4], we therefore conclude that $\|(f,P)\|_{\frac{65}{64}Q} \leq C(t_G)\|(f,P)\|_{\frac{65}{64}\widehat{Q}}$. So,

$$\left[M_{(Q,\mathcal{A}^-)}(f,P)\right]^p \leq C(t_G) \cdot \left[\|(f,P)\|^p_{\frac{65}{64}\widehat{Q}} + \Delta^p_{\text{junk}}|P|^p \right].$$

The number of cubes Q relevant to the sum in **(I)** is at most $C(t_G)$. Therefore,

$$\textbf{(I)} \leq C(t_G) \cdot \left[\|(f,P)\|^p_{\frac{65}{64}\widehat{Q}} + \Delta^p_{\text{junk}}|P|^p \right].$$

The term **(VI)** is equal to $\Delta^p_{\text{new}}|P|^p$.

It remains to estimate the term **(V)**. Recall that $\#(\frac{65}{64}\widehat{Q} \cap E) \leq N \leq \Delta_0^{-n}$ (see (2.71)). Hence,

$$\mathbf{(V)} = \Delta_{\text{new}}^{2p} \sum_{x \in \frac{65}{64}\widehat{Q} \cap E} |f(x) - P(x)|^p \leq \Delta_{\text{new}}^{2p}\, \Delta_0^{-C}\, \|(f,P)\|^p_{\frac{65}{64}\widehat{Q}} \tag{2.112}$$

$$\leq \|(f,P)\|^p_{\frac{65}{64}\widehat{Q}}. \tag{2.113}$$

We explain below the previous two estimates.

We deduce the estimate in (2.112) by picking a function $\widetilde{F}$ that satisfies

$$\|\widetilde{F}\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})} + \delta_{\widehat{Q}}^{-mp}\|\widetilde{F} - P\|_{L^p(\frac{65}{64}\widehat{Q})} \leq 2 \cdot \|(f,P)\|^p_{\frac{65}{64}\widehat{Q}}$$

and $\widetilde{F} = f$ on $E \cap \frac{65}{64}\widehat{Q}$. For each $x \in E \cap \frac{65}{64}\widehat{Q}$, we apply estimate (2.4) from Lemma 10 of [3] to the function $F = \widetilde{F} - P$. Hence, we have

$$|f(x) - P(x)| = \left|\widetilde{F}(x) - P(x)\right| \leq C \cdot \delta_{\widehat{Q}}^{-n/p}\|\widetilde{F} - P\|_{L^p(\frac{65}{64}\widehat{Q})} + \delta_{\widehat{Q}}^{m-n/p}\|\widetilde{F}\|_{\mathbb{X}(\frac{65}{64}\widehat{Q})}.$$

Recall that $\delta_{\widehat{Q}} \geq c\Delta_0$, since $\widehat{Q}$ is a testing cube. Hence, we have $|f(x) - P(x)| \leq C\,\Delta_0^{-C}|(f,P)|_{\frac{65}{64}\widehat{Q}}$. Summing over $x \in E \cap \frac{65}{64}\widehat{Q}$ and using the fact that $\#(E \cap \frac{65}{64}\widehat{Q}) \leq \Delta_0^{-C}$, we obtain the stated estimate.

We deduce the estimate in (2.113) from the estimate $\Delta_{\text{new}} \leq \Delta_0^{C/(2p)}$; see (2.82). Therefore,

$$\left[M_{\widehat{Q}}(f,P)\right]^p = \mathbf{(I)} + \mathbf{(V)} + \mathbf{(VI)} \leq C(t_G) \cdot \left[\|(f,P)\|^p_{\frac{65}{64}\widehat{Q}} + \Delta^p_{\text{new}}|P|^p + \Delta^p_{\text{junk}}|P|^p\right].$$

Since $\Delta_{\text{junk}} \leq \Delta_{\text{new}}$ (see (2.82)), the above estimate implies the conclusion of Lemma 6. □

Lemma 6 implies the conditional inequality if $\widehat{Q}$ is η-simple for

$$\eta = \min\{c_*(\mathcal{A}^-), [100S(\mathcal{A}^-)]^{-1}\}.$$

So we may assume that $\widehat{Q}$ is not η-simple, for η as above (as in (1.107) of [4]).

Both Proposition 5 and 6 of [4] hold in the present setting, without change. The proofs are as before.

We now prove the conditional inequality. We describe how the estimates from before will need to be changed in the present setting.

On the right-hand side of (1.142) of [4] we add the terms **(V)** and **(VI)**. Note that

$$\begin{aligned}\mathbf{(V)} + \mathbf{(VI)} &= \Delta_{\text{new}}^{2p} \sum_{x \in \frac{65}{64}\widehat{Q} \cap E} |f(x) - P(x)|^p + \Delta^p_{\text{new}}|P|^p \\ &\leq \|(f,P)\|^p_{\frac{65}{64}\widehat{Q}} + \Delta^p_{\text{new}}|P|^p \quad \text{(thanks to (2.112))} \\ &\leq [\text{RHS of the conditional inequality}]^p.\end{aligned}$$

Therefore, the extra terms do not hurt.

Our inductive assumption now states that

$$M_{(Q,\mathcal{A}^-)}(f, R^{\widehat{Q}}_Q) \le C \cdot \Big[\|(f, R^{\widehat{Q}}_Q)\|_{\frac{65}{64}Q} + \Delta_{\text{junk}} |R^{\widehat{Q}}_Q| \Big]$$
$$\le C \cdot \Big[\|H\|_{\mathbb{X}(\frac{65}{64}Q)} + \delta_Q^{-m} \|H - R^{\widehat{Q}}_Q\|_{L^p(\frac{65}{64}Q)} + \Delta_{\text{junk}} |R^{\widehat{Q}}_Q| \Big].$$

Hence, in place of (1.143) of [4], we now have

$$\begin{aligned} \big[M_{\widehat{Q}}(f,P)\big]^p \le C(t_G) \cdot \Big[& \|H\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})} + \delta_{\widehat{Q}}^{-mp} \|H - P\|^p_{L^p(\frac{65}{64}\widehat{Q})} \\ & + \sum_{Q \subset (1+100t_G)\widehat{Q}} \Big[\|H\|^p_{\mathbb{X}(\frac{65}{64}Q)} + \delta_Q^{-mp} \|H - R^{\widehat{Q}}_Q\|^p_{L^p(\frac{65}{64}Q)} \Big] \\ & + \Delta^p_{\text{junk}} \sum_{Q \subset (1+100t_G)\widehat{Q}} |R^{\widehat{Q}}_Q|^p \quad (=: \mathfrak{S}) \\ & + \big[\text{RHS of conditional inequality}\big]^p \Big]. \end{aligned} \tag{2.114}$$

The third and fourth lines contain new terms not present in the original estimate.

We will now estimate the extra term $\mathfrak{S}$ in the third line of (2.114)

We write $\mathfrak{S} = \mathfrak{S}_1 + \mathfrak{S}_2$, with

$$\mathfrak{S}_1 = \Delta^p_{\text{junk}} \sum_{\substack{Q \subset (1+100t_G)\widehat{Q} \\ \delta_Q \ge t_G \delta_{\widehat{Q}}}} |R^{\widehat{Q}}_Q|^p, \quad \text{and} \quad \mathfrak{S}_2 = \Delta^p_{\text{junk}} \sum_{\substack{Q \subset (1+100t_G)\widehat{Q} \\ \delta_Q < t_G \delta_{\widehat{Q}}}} |R^{\widehat{Q}}_Q|^p.$$

We estimate the term $\mathfrak{S}_1$. The number of cubes in $CZ(\mathcal{A}^-)$ is at most Δ_0^{-C}. Moreover, by definition, $R^{\widehat{Q}}_Q = P$ for each $Q \in CZ(\mathcal{A}^-)$ relevant to the $\mathfrak{S}_1$ (see (2.99)). Hence,

$$\begin{aligned} \mathfrak{S}_1 &\le \Delta^p_{\text{junk}} \cdot \Delta_0^{-C} \cdot |P|^p \\ &\le \Delta^p_{\text{new}} \cdot |P|^p \le \big[\text{RHS of conditional inequality}\big]^p. \end{aligned} \tag{2.115}$$

(Here, we use that $\Delta_{\text{junk}} \le \Delta^2_{\text{new}} \le \Delta_{\text{new}} \cdot \Delta_0^{C/p}$; see (2.82).)

We next estimate the term $\mathfrak{S}_2$. Let $Q \in CZ(\mathcal{A}^-)$ satisfy $Q \subset (1 + 100t_G)\widehat{Q}$ and $\delta_Q < t_G \delta_{\widehat{Q}}$. Then the keystone cube $Q^\# = \mathcal{K}(Q)$ associated to Q satisfies $S_1 Q^\# \subset CQ \subset (1 + Ct_G)\widehat{Q} \subset \frac{65}{64}\widehat{Q}$, where the first inclusion is a property of the map $\mathcal{K}$ (see the statement of the KEYSTONE-ORACLE), the second inclusion is due to $\delta_Q < t_G \delta_{\widehat{Q}}$, and the last inclusion follows if we take t_G sufficiently small. By definition (2.99) we have $R^{\widehat{Q}}_Q = R^\#_{Q^\#}$. Moreover, since the number of cubes in $CZ(\mathcal{A}^-)$ is at most Δ_0^{-C}, we have

$$\mathfrak{S}_2 \le \max_{Q^\# \text{ keystone}} \Big\{ \Delta^p_{\text{junk}} \Delta_0^{-C} |R^\#_{Q^\#}|^p : S_1 Q^\# \subset \frac{65}{64}\widehat{Q} \Big\}. \tag{2.116}$$

Let $Q^{\#}$ be a keystone cube with $S_1 Q^{\#} \subset \frac{65}{64}\widehat{Q}$. Lemma 5 states that

$$|R^{\#}_{Q^{\#}} - P| \leq C \cdot \left[\|(f,P)\|_{S_1 Q^{\#}} + \Delta_{\text{junk}} |P| \right].$$

Hence,

$$|R^{\#}_{Q^{\#}}|^p \leq C \cdot \left[\|(f,P)\|^p_{S_1 Q^{\#}} + |P|^p \right]. \tag{2.117}$$

We define

$$\widetilde{F} = P + \sum_{x \in S_1 Q^{\#} \cap E} \theta_x \cdot (f(x) - P(x))$$

where $\theta_x(y)$ $(x \in E)$ are cutoff functions satisfying (a) $\theta_x \equiv 1$ on a neighborhood of x, (b) θ_x is supported on a ball $B(x, c\Delta_0)$ for a small universal constant c, and (c) $\|\partial^\alpha \theta_x\|_{L^\infty} \leq \Delta_0^{-C}$ for all $|\alpha| \leq m$. Indeed note that we may take $\theta_x(y) = \theta(y - x)$ for a fixed cutoff function θ supported on a small ball about the origin. From (a) and (b) we deduce that $\theta_x(z) \equiv 0$ for any $z \in E \setminus \{x\}$, since $|x - y| \geq \Delta_0$ for distinct points $x, y \in E$. Thus, we have $\widetilde{F}(x) = f(x)$ for each $x \in E$. Moreover,

$$\|\widetilde{F}\|^p_{X(S_1 Q^{\#})} \leq \Delta_0^{-C} \sum_{x \in S_1 Q^{\#} \cap E} |f(x) - P(x)|^p$$

and

$$\|\widetilde{F} - P\|^p_{L^p(S_1 Q^{\#})} \leq \Delta_0^{-C} \sum_{x \in S_1 Q^{\#} \cap E} |f(x) - P(x)|^p.$$

Hence, by definition of the trace seminorm, $\|(f,P)\|^p_{S_1 Q^{\#}} \leq \Delta_0^{-C} \sum_x |f(x) - P(x)|^p$. Thus, estimate (2.117) implies that

$$|R^{\#}_{Q^{\#}}|^p \leq C \Big[\Delta_0^{-C'} \sum_{x \in S_1 Q^{\#} \cap E} |f(x) - P(x)|^p + |P|^p \Big]. \tag{2.118}$$

Therefore, returning to (2.116), we have

$$\mathfrak{S}_2 \leq \Delta^p_{\text{junk}} \Delta_0^{-C''} \Big[\sum_{x \in \frac{65}{64}\widehat{Q} \cap E} |f(x) - P(x)|^p + |P|^p \Big].$$

Since $\Delta^p_{\text{junk}} \Delta_0^{-C''} \leq \Delta^{4p}_{\text{new}} \cdot \left[\Delta^p_{\text{new}} \Delta_0^{-C''} \right] \leq \Delta^{4p}_{\text{new}}$ (see (2.82)), we conclude that

$$\mathfrak{S}_2 \leq \Delta^{4p}_{\text{new}} \sum_{x \in \frac{65}{64}\widehat{Q} \cap E} |f(x) - P(x)|^p + \Delta^{4p}_{\text{new}} |P|^p \leq \Delta^{2p}_{\text{new}} \left[M_{\widehat{Q}}(f,P) \right]^p$$

$$\leq \frac{1}{2C(t_G)} \left[M_{\widehat{Q}}(f,P) \right]^p,$$

where $C(t_G)$ is the constant in (2.114). To obtain the previous estimates, we make sure to choose

$$\Delta^{2p}_{\text{new}} \leq \frac{1}{2C(t_G)}$$

(see (2.98)).

This completes our estimation of the term $\mathfrak{S}_2$.

In the estimate (2.114), we consider the term $C(t_G)\cdot\mathfrak{S} = C(t_G)\cdot\mathfrak{S}_1 + C(t_G)\cdot\mathfrak{S}_2$ on the right-hand side, and note that $C(t_G)\cdot\mathfrak{S}_2$ is irrelevant since it is bounded by a half of the left-hand side; moreover, the term $C(t_G)\cdot\mathfrak{S}_1$ is bounded from above by $C(t_G)\cdot$[RHS of conditional inequality]^p, thanks to (2.115). Therefore, in place of (2.114), we have the simpler estimate:

$$\begin{aligned} M_{\widehat{Q}}(f,P)^p \le C(t_G)\cdot\Big[&\|H\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})} + \delta_{\widehat{Q}}^{-mp}\|H-P\|^p_{L^p(\frac{65}{64}\widehat{Q})} \\ &+ \sum_{Q\subset(1+100t_G)\widehat{Q}} \Big[\|H\|^p_{\mathbb{X}(\frac{65}{64}Q)} + \delta_Q^{-mp}\|H-R^{\widehat{Q}}_Q\|^p_{L^p(\frac{65}{64}Q)}\Big] \\ &+ [\text{RHS of conditional inequality}]^p\Big]. \end{aligned} \tag{2.119}$$

The difference between the estimates (2.119) and (1.143) of [4] is that the right-hand side of (2.119) contains an extra term: [RHS of conditional inequality]^p.

The estimates in **Stage II** are unchanged. Using these estimates in (2.119), we obtain

$$\begin{aligned} M_{\widehat{Q}}(f,P)^p &\le C(t_G)\cdot\Big(\|H\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})} + \delta_{\widehat{Q}}^{-mp}\|H-P\|^p_{L^p(\frac{65}{64}\widehat{Q})} \\ &\quad + \sum_{\substack{Q\subset(1+100t_G)\widehat{Q}\\ \delta_Q<t_G\delta_{\widehat{Q}}}} \delta_Q^{-mp}\|H-R^{\#}_{\mathcal{K}(Q)}\|^p_{L^p(\frac{65}{64}Q)} \\ &\quad + \big[\text{RHS of conditional inequality}\big]^p\Big) \\ &\le C(t_G)\cdot\Big(\|H\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})} + \delta_{\widehat{Q}}^{-mp}\|H-P\|^p_{L^p(\frac{65}{64}\widehat{Q})} \\ &\quad + \sum_{\substack{Q^{\#}\ \text{keystone}\\ S_1Q^{\#}\subset\frac{65}{64}\widehat{Q}}} (\delta_{Q^{\#}})^{-mp}\|H-R^{\#}_{Q^{\#}}\|^p_{L^p(S_1Q^{\#})} \\ &\quad + \big[\text{RHS of conditional inequality}\big]^p\Big). \end{aligned} \tag{2.120}$$

The difference between the estimates (2.120) and (1.144) of [4] is the extra term $\big[\text{RHS of conditional inequality}\big]^p$ that appears in (2.120).

We now examine the estimates in **Stage III**.

In place of the inequality (1.145) of [4], which reads

$$\delta_{Q^{\#}}^{-mp}\|H-R^{\#}_{Q^{\#}}\|^p_{L^p(S_1Q^{\#})} \le C\|H\|^p_{\mathbb{X}(S_1Q^{\#})},$$

we now apply (2.95) which reads

$$\delta_{Q^{\#}}^{-mp}\|H-R^{\#}_{Q^{\#}}\|^p_{L^p(S_1Q^{\#})} \le C\big[\|H\|^p_{\mathbb{X}(S_1Q^{\#})} + \Delta^p_{\text{junk}}|R^{\#}_{Q^{\#}}|^p\big].$$

Thus, from (2.118) we have

$$\delta_{Q^\#}^{-mp}\|H-R^\#_{Q^\#}\|^p_{L^p(S_1Q^\#)} \le C\|H\|^p_{\mathbb{X}(S_1Q^\#)} + C\,\Delta_{\text{junk}}^p\Big[\Delta_0^{-C'}\sum_{x\in S_1Q^\#\cap E}|f(x)-P(x)|^p+|P|^p\Big].$$

There are at most $CN \le \Delta_0^{-C}$ keystone cubes in $CZ(\mathcal{A}^-)$. Hence, since the collection $\{S_1Q^\# : Q^\# \text{ keystone}\}$ has bounded overlap, we have

$$\sum_{\substack{Q^\#\ \text{keystone}\\ S_1Q^\#\subset\frac{65}{64}\widehat{Q}}}\delta_{Q^\#}^{-mp}\|H-R^\#_{Q^\#}\|^p_{L^p(S_1Q^\#)} \le C\,\|H\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})} + C\,\Delta_{\text{junk}}^p\,\Delta_0^{-C}\Big[\sum_{x\in\frac{65}{64}\widehat{Q}\cap E}|f(x)-P(x)|^p+|P|^p\Big].$$

We have $C\,\Delta_{\text{junk}}^p\Delta_0^{-C} \le C\,\Delta_{\text{new}}^{3p} \le \frac{1}{2C(t_G)}\Delta_{\text{new}}^{2p}$, due to assumptions (2.82) and (2.98). Thus, the term inside the curly braces in the above estimate is bounded by

$$\frac{1}{2C(t_G)}\,[\ \textbf{(V)} + \ \textbf{(VI)}\ ] \le \frac{1}{2C(t_G)}M_{\widehat{Q}}(f,P)^p.$$

We put the previous estimates into (2.120) to obtain

$$M_{\widehat{Q}}(f,P)^p \le C(t_G)\Big[\|H\|^p_{\mathbb{X}(\frac{65}{64}\widehat{Q})}+\delta_{\widehat{Q}}^{-mp}\|H-P\|^p_{L^p(\frac{65}{64}\widehat{Q})}+\frac{1}{2C(t_G)}M_{\widehat{Q}}(f,P)^p\Big] + \big[\text{RHS of conditional inequality}\big]^p.$$

Thus, from the third bullet point in Proposition 5 of [4], we deduce that

$$M_{\widehat{Q}}(f,P)^p \le C(t_G)\Lambda^{(2D+1)p}\|(f,P)\|^p_{\frac{65}{64}\widehat{Q}} + \big[\text{RHS of conditional inequality}\big]^p.$$

Since $\Lambda^{2D+1} \le 1/\epsilon$, this estimate implies the conditional inequality in Proposition 2. This completes the proof of Proposition 2.

We fix $t_G > 0$ to be a universal constant, small enough so that the preceding results hold. We define the universal constant $a(\mathcal{A}) = a_{\text{new}}$, with a_{new} as in Lemma 11 of [4].

For a moment, we fix $\epsilon = \epsilon_0$ in Proposition 2 for a small universal constant ϵ_0. This implies the following result.

Proposition 3. *There exist universal constants $\epsilon_0 > 0$ and $C \ge 1$ such that the following estimates hold.*

(Unconditional inequality) $\|(f,P)\|_{(1+a(\mathcal{A}))\widehat{Q}} \le C\cdot M_{\widehat{Q}}(f,P)$.

(Conditional inequality) *If $3\widehat{Q}$ is tagged with $(\mathcal{A},\epsilon_0)$, then*

$$M_{\widehat{Q}}(f,P) \le C\left[\|(f,P)\|_{\frac{65}{64}\widehat{Q}}+\Delta_{new}\cdot|P|\right].$$

We no longer fix $\epsilon = \epsilon_0$. Once again, we assume that ϵ is a small parameter, less than a small enough universal constant.

We assume that

$$\Delta_{\text{new}} \le c(\epsilon), \tag{2.121}$$

for a small enough constant $c(\epsilon)$, depending only on ϵ, m, n, and p.

We will need new proofs of Propositions 8, 9, and 10 of Section 1.5 of [4]. We recall these results below (see Propositions 4, 5, and 6) and give the new proofs.

Proposition 4. *Let $\widehat{Q}$ be a testing cube. If*

$$\left[\#\left(\frac{65}{64}\widehat{Q} \cap E\right) \le 1 \text{ or } \overline{\overline{\sigma}}(\widehat{Q}) \text{ has an } (\mathcal{A}', x_{\widehat{Q}}, \epsilon, \delta_{\widehat{Q}})\text{-basis for some } \mathcal{A}' \le \mathcal{A}\right],$$

then $(1 + a(\mathcal{A}))\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^{\kappa})$. Otherwise, no cube containing $3\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^{1/\kappa})$. Here, $\kappa > 0$ is a universal constant.

Proof. If $\#(\frac{65}{64}\widehat{Q} \cap E) \le 1$, then $(1 + a(\mathcal{A}))\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon)$.

Suppose $\overline{\overline{\sigma}}(\widehat{Q})$ has an $(\mathcal{A}', x_{\widehat{Q}}, \epsilon, \delta_{\widehat{Q}})$-basis with $\mathcal{A}' \le \mathcal{A}$. Call this basis $(P_\alpha)_{\alpha \in \mathcal{A}'}$. Then

- $P_\alpha \in \epsilon \cdot \delta_{\widehat{Q}}^{|\alpha| + n/p - m} \cdot \overline{\overline{\sigma}}(\widehat{Q})$ for all $\alpha \in \mathcal{A}'$.
- $\partial^\beta P_\alpha(x_{\widehat{Q}}) = \delta_{\alpha\beta}$ for all $\alpha, \beta \in \mathcal{A}'$.
- $|\partial^\beta P_\alpha(x_{\widehat{Q}})| \le \epsilon \cdot \delta_{\widehat{Q}}^{|\alpha| - |\beta|}$ for all $\alpha \in \mathcal{A}'$, $\beta \in \mathcal{M}$, $\beta > \alpha$.

Since $\overline{\overline{\sigma}}(\widehat{Q}) = \{P : M_{\widehat{Q}}(0, P) \le 1\}$, we have $M_{\widehat{Q}}(0, P_\alpha) \le \epsilon \delta_{\widehat{Q}}^{|\alpha| + n/p - m}$ for $\alpha \in \mathcal{A}'$. So, the unconditional inequality gives

$$\|(0, P_\alpha)\|_{(1 + a(\mathcal{A}))\widehat{Q}} \le C' \epsilon \delta_{\widehat{Q}}^{|\alpha| + n/p - m} \quad \text{for all } \alpha \in \mathcal{A}'.$$

Thus,

$$P_\alpha \in C' \epsilon \delta_{\widehat{Q}}^{|\alpha| + n/p - m} \sigma((1 + a(\mathcal{A}))\widehat{Q}) \quad \text{for all } \alpha \in \mathcal{A}'.$$

Thus, combined with the second and third bullet points above, we have that $(P_\alpha)_{\alpha \in \mathcal{A}'}$ is an $(\mathcal{A}', x_{\widehat{Q}}, C'\epsilon, \delta_{\widehat{Q}})$-basis for $\sigma((1 + a(\mathcal{A}))\widehat{Q})$. Hence, $(P_\alpha)_{\alpha \in \mathcal{A}'}$ is an $(\mathcal{A}', x_{\widehat{Q}}, \epsilon^{\kappa}, \delta_{(1 + a(\mathcal{A}))\widehat{Q}})$-basis for $\sigma((1 + a(\mathcal{A}))\widehat{Q})$, for a small enough universal constant κ. Since $\mathcal{A}' \le \mathcal{A}$, it follows that $(1 + a(\mathcal{A}))\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^{\kappa})$, as claimed. That proves the first half of Proposition 4.

On the other hand, suppose $Q \supset 3\widehat{Q}$ and suppose Q is tagged with $(\mathcal{A}, \epsilon^{1/\kappa'})$, for a small enough universal constant $\kappa' > 0$, to be chosen below (not any previous κ'). Then $3\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^{\kappa/\kappa'})$ for some universal constant $\kappa > 0$, thanks to Lemma 28 of [3]. Hence, as long as ϵ is small enough so that $\epsilon^{\kappa/\kappa'} \le \epsilon_0$, the conditional inequality applies:

$$M_{\widehat{Q}}(0, P) \le C\left[\|(0, P)\|_{\frac{65}{64}\widehat{Q}} + \Delta_{\text{new}}|P|\right] \quad \text{for any } P \in \mathcal{P}.$$

By Lemma 28 of [3], we also know $\frac{65}{64}\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^{\kappa/\kappa'})$. So either $\#(\frac{65}{64}Q \cap E) \leq 1$ (in which case we have finished the proof of Proposition 4) or else $\sigma(\frac{65}{64}Q)$ has an $(\mathcal{A}', x_{\widehat{Q}}, \epsilon^{\kappa/\kappa'}, \delta_{\widehat{Q}})$-basis for some $\mathcal{A}' \leq \mathcal{A}$.

In the latter case, Lemma 25 of [3] gives an $(\mathcal{A}'', x_{\widehat{Q}}, \epsilon^{\overline{\kappa}/\kappa'}, \delta_{\widehat{Q}}, \Lambda)$-basis, with

$$\mathcal{A}'' \leq \mathcal{A}' \leq \mathcal{A},\ \widetilde{\kappa} \leq \overline{\kappa} \leq \widetilde{\widetilde{\kappa}},\ \text{with } \widetilde{\kappa}, \widetilde{\widetilde{\kappa}} > 0 \text{ universal constants independent of } \kappa',$$
$$\text{and } \epsilon^{\overline{\kappa}/\kappa'} \Lambda^{100D} \leq \epsilon^{\overline{\kappa}/2\kappa'}.$$

Call this basis $(P_\alpha)_{\alpha \in \mathcal{A}''}$. Then

- $P_\alpha \in \epsilon^{\overline{\kappa}/\kappa'} \cdot \delta_{\widehat{Q}}^{|\alpha|+n/p-m} \cdot \sigma(\frac{65}{64}Q)$ for all $\alpha \in \mathcal{A}''$.
- $\partial^\beta P_\alpha(x_{\widehat{Q}}) = \delta_{\alpha\beta}$ for all $\alpha, \beta \in \mathcal{A}''$.
- $|\partial^\beta P_\alpha(x_{\widehat{Q}})| \leq \epsilon^{\overline{\kappa}/\kappa'} \cdot \delta_{\widehat{Q}}^{|\alpha|-|\beta|}$ for all $\alpha \in \mathcal{A}''$, $\beta \in \mathcal{M}$, $\beta > \alpha$.
- $|\partial^\beta P_\alpha(x_{\widehat{Q}})| \leq \Lambda \cdot \delta_{\widehat{Q}}^{|\alpha|-|\beta|}$ for all $\alpha \in \mathcal{A}''$, $\beta \in \mathcal{M}$.

We deduce a few conclusions from the above bullet points. The first bullet point implies that $\|(0, P_\alpha)\|_{\frac{65}{64}\widehat{Q}} \leq \epsilon^{\overline{\kappa}/\kappa'} \delta_{\widehat{Q}}^{|\alpha|+n/p-m}$; the last bullet point implies that $|\partial^\beta P_\alpha(x_{\widehat{Q}})| \leq \Lambda \cdot \Delta_0^{-C}$ for all $\alpha \in \mathcal{A}''$, $\beta \in \mathcal{M}$ (since $\Delta_0 \leq \delta_{\widehat{Q}} \leq 1$), hence

$$|P_\alpha| = \Big(\sum_{\beta \in \mathcal{M}} |\partial^\beta P_\alpha(0)|^p \Big)^{1/p} \leq C \cdot \Big(\sum_{\beta \in \mathcal{M}} |\partial^\beta P_\alpha(x_{\widehat{Q}})|^p \Big)^{1/p} \quad (\text{since } |x_{\widehat{Q}}| \leq C)$$
$$\leq C' \Lambda \cdot \Delta_0^{-C}.$$

Hence, the (known) conditional inequality implies the estimate

$$M_{\widehat{Q}}(0, P_\alpha) \leq C \cdot \Big[\|(0, P_\alpha)\|_{\frac{65}{64}\widehat{Q}} + \Delta_{\text{new}} \cdot |P_\alpha| \Big]$$
$$\leq C\epsilon^{\overline{\kappa}/\kappa'} \delta_{\widehat{Q}}^{|\alpha|+n/p-m} + C\Lambda\Delta_0^{-C}\Delta_{\text{new}} \leq C'\epsilon^{\overline{\kappa}/\kappa'} \delta_{\widehat{Q}}^{|\alpha|+n/p-m}, \tag{2.122}$$

for $\alpha \in \mathcal{A}''$, where we have used that $\delta_{\widehat{Q}} \leq 1$ and $|\alpha| + n/p - m < 0$, and

$$\Lambda \cdot \Delta_0^{-C}\Delta_{\text{new}} \overset{(2.82)}{\leq} \Lambda \cdot \Delta_{\text{new}}^{1/2} \overset{(2.121)}{\leq} \Lambda \cdot \epsilon^{2\cdot\widetilde{\widetilde{\kappa}}/\kappa'} \leq \epsilon^{\overline{\kappa}/\kappa'}.$$

Here, in the last inequality, we use that $\Lambda \leq \Lambda^{100D} \leq \epsilon^{-\overline{\kappa}/2\kappa'}$, where $\overline{\kappa} \leq \widetilde{\widetilde{\kappa}}$.

Now, the estimate (2.122) implies that

$$P_\alpha \in C'\epsilon^{\overline{\kappa}/\kappa'} \delta_{\widehat{Q}}^{|\alpha|+n/p-m} \cdot \overline{\overline{\sigma}}(\widehat{Q}) \quad \text{for all } \alpha \in \mathcal{A}''.$$

This estimate, together with the second and third bullet points above, shows that

$$(P_\alpha)_{\alpha \in \mathcal{A}''} \text{ is an } (\mathcal{A}'', x_{\widehat{Q}}, C\epsilon^{\overline{\kappa}/\kappa'}, \delta_{\widehat{Q}})\text{-basis for } \overline{\overline{\sigma}}(\widehat{Q}).$$

We ensure that $C\epsilon^{\overline{\kappa}/\kappa'} \leq \epsilon$ by choosing κ' to be a small enough universal constant. Hence, $\overline{\overline{\sigma}}(\widehat{Q})$ has an $(\mathcal{A}'', x_{\widehat{Q}}, \epsilon, \delta_{\widehat{Q}})$-basis with $\mathcal{A}'' \leq \mathcal{A}' \leq \mathcal{A}$. This completes the proof of Proposition 4. □

Proposition 5. . *Suppose* $\widehat{Q}_1 \subset \widehat{Q}_2$ *are testing cubes with* $\#(3\widehat{Q}_2 \cap E) \geq 2$, *and* $(1+a(\mathcal{A}))\widehat{Q}_1 \cap E = 3\widehat{Q}_2 \cap E$. *Suppose* $\overline{\overline{\sigma}}(\widehat{Q}_1)$ *has an* $(\mathcal{A}', x_{\widehat{Q}_1}, \epsilon, \delta_{\widehat{Q}_2})$*-basis. Then* $3\widehat{Q}_2$ *is tagged with* $(\mathcal{A}', \epsilon^\kappa)$ *for a universal constant* κ.

Proof. By Lemma 27 of [3], $\overline{\overline{\sigma}}(\widehat{Q}_1)$ has an $(\mathcal{A}'', x_{\widehat{Q}_2}, \epsilon^\kappa, \delta_{\widehat{Q}_2})$-basis, with $\mathcal{A}'' \leq \mathcal{A}'$, for some universal constant κ. Call that basis $(P_\alpha)_{\alpha\in\mathcal{A}''}$. Then for each $\alpha \in \mathcal{A}''$, we have

- $P_\alpha \in \epsilon^\kappa \cdot \delta_{\widehat{Q}_2}^{|\alpha|+n/p-m} \cdot \overline{\overline{\sigma}}(\widehat{Q}_1)$.
- $\partial^\beta P_\alpha(x_{\widehat{Q}_2}) = \delta_{\alpha\beta}$ for all $\beta \in \mathcal{A}''$.
- $|\partial^\beta P_\alpha(x_{\widehat{Q}_2})| \leq \epsilon^\kappa \cdot \delta_{\widehat{Q}_2}^{|\alpha|-|\beta|}$ for $\beta \in \mathcal{M}$, $\beta > \alpha$.

The first condition here gives $M_{\widehat{Q}_1}(0, P_\alpha) \leq \epsilon^\kappa \delta_{\widehat{Q}_2}^{|\alpha|+n/p-m}$. So, by the unconditional inequality,

$$\|(0, P_\alpha)\|_{(1+a(\mathcal{A}))\widehat{Q}_1} \leq C\,\epsilon^\kappa\,\delta_{\widehat{Q}_2}^{|\alpha|+n/p-m}.$$

Hence, $P_\alpha \in C\epsilon^\kappa \delta_{\widehat{Q}_2}^{|\alpha|+n/p-m}\sigma((1+a(\mathcal{A}))\widehat{Q}_1)$. By Lemma 15 of [3], we know $\sigma(3\widehat{Q}_2)$ is comparable to $\sigma((1+a(\mathcal{A}))\widehat{Q}_1) + \mathcal{B}(x_{\widehat{Q}_2}, \delta_{\widehat{Q}_2})$, so

$$\sigma((1+a(\mathcal{A}))\widehat{Q}_1) \subset C\,\sigma(3\widehat{Q}_2).$$

Thus, $P_\alpha \in C\epsilon^\kappa \delta_{\widehat{Q}_2}^{|\alpha|+n/p-m}\sigma(3\widehat{Q}_2)$ for all $\alpha \in \mathcal{A}''$. With the second and third bullet points above, this shows that $\sigma(3\widehat{Q}_2)$ has an $(\mathcal{A}'', x_{\widehat{Q}_2}, C\epsilon^\kappa\delta_{\widehat{Q}_2})$-basis, with $\mathcal{A}'' \leq \mathcal{A}'$. Therefore, $3\widehat{Q}_2$ is tagged with $(\mathcal{A}', \epsilon^{\kappa/2})$, if ϵ is less than a small enough universal constant. This completes the proof of Proposition 5. □

Corollary 1 of [4] is a direct consequence of Proposition 5, just as before.

Proposition 6. *Suppose that* $\widehat{Q}_1 \subset \widehat{Q}_2$ *are testing cubes,* $\#(3\widehat{Q}_2 \cap E) \geq 2$, *and* $(1+a(\mathcal{A}))\widehat{Q}_1 \cap E = 3\widehat{Q}_2 \cap E$. *Suppose* $3\widehat{Q}_2$ *is tagged with* $(\mathcal{A}, \epsilon)$. *Then* $\overline{\overline{\sigma}}(\widehat{Q}_1)$ *has an* $(\mathcal{A}', x_{\widehat{Q}_1}, \epsilon^{\kappa'}, \delta_{\widehat{Q}_2})$*-basis for some* $\mathcal{A}' \leq \mathcal{A}$ *and for some universal constant* κ'.

Proof. Since $3\widehat{Q}_1 \subset 3\widehat{Q}_2$ and $3\widehat{Q}_2$ is tagged with $(\mathcal{A}, \epsilon)$, Lemma 28 of [3] shows that $3\widehat{Q}_1$ is tagged with $(\mathcal{A}, \epsilon^\kappa)$ for a universal constant κ. Hence, the Conditional Inequality holds for $\widehat{Q}_1$. Hence,

$$M_{\widehat{Q}_1}(0, P) \leq C\left[\|(0, P)\|_{\frac{65}{64}\widehat{Q}_1} + \Delta_{new}|P|\right] \quad \text{for } P \in \mathcal{P}. \tag{2.123}$$

Now, since $\frac{65}{64}\widehat{Q}_1 \cap E = 3\widehat{Q}_2 \cap E$ and $\frac{65}{64}\widehat{Q}_1 \subset 3\widehat{Q}_2$, we know from Lemma 15 of [3] that

$$\sigma(3\widehat{Q}_2) \subset C \cdot \left[\sigma\Big(\frac{65}{64}\widehat{Q}_1\Big) + \mathcal{B}\big(x_{\widehat{Q}_2}, \delta_{3\widehat{Q}_2}\big)\right].$$

(We have $\mathcal{B}(x_{\widehat{Q}_1}, \delta_{3\widehat{Q}_2}) \subset C\mathcal{B}(x_{\widehat{Q}_2}, \delta_{3\widehat{Q}_2})$, because $|x_{\widehat{Q}_1} - x_{\widehat{Q}_2}| \leq \delta_{\widehat{Q}_2}$. Hence, the above inclusion follows from Lemma 15 of [3].)

Recall that $3\widehat{Q}_2$ is tagged with $(\mathcal{A}, \epsilon)$ and $\#(3\widehat{Q}_2 \cap E) \geq 2$. Hence, $\sigma(3\widehat{Q}_2)$ has an $(\mathcal{A}', x_{\widehat{Q}_2}, \epsilon, \delta_{3\widehat{Q}_2})$-basis, for some $\mathcal{A}' \leq \mathcal{A}$. By Lemma 25 of [3], there exist a multiindex set $\mathcal{A}'' \leq \mathcal{A}' \leq \mathcal{A}$ and numbers $\Lambda \geq 1$, $\kappa_1 \leq \overline{\kappa} \leq \kappa_2$, such that

$$\sigma(3\widehat{Q}_2) \text{ has an } (\mathcal{A}'', x_{\widehat{Q}_2}, \epsilon^{\overline{\kappa}}, \delta_{3\widehat{Q}_2}, \Lambda)\text{-basis, where } \epsilon^{\overline{\kappa}}\Lambda^{100D} \leq \epsilon^{\overline{\kappa}/2},$$

for some universal constants $\kappa_1, \kappa_2 \in (0, 1]$. Therefore,

$$\sigma\Big(\frac{65}{64}\widehat{Q}_1\Big) + \mathcal{B}(x_{\widehat{Q}_2}, \delta_{3\widehat{Q}_2}) \text{ has an } (\mathcal{A}'', x_{\widehat{Q}_2}, C\epsilon^{\overline{\kappa}}, \delta_{3\widehat{Q}_2}, \Lambda)\text{-basis.}$$

From Lemma 23 of [3], we see that $\sigma\big(\frac{65}{64}\widehat{Q}_1\big)$ has an $(\mathcal{A}'', x_{\widehat{Q}_2}, C'\epsilon^{\overline{\kappa}}\Lambda, \delta_{3\widehat{Q}_2}, C\Lambda)$-basis. Here, $C'\epsilon^{\overline{\kappa}}\Lambda \leq C'\epsilon^{\overline{\kappa}/2} \leq \epsilon^{\overline{\kappa}/4}$, for sufficiently small ϵ. Let $(P_\alpha)_{\alpha\in\mathcal{A}}$ be that basis. Thus, for each $\alpha \in \mathcal{A}''$,

- $P_\alpha \in \epsilon^{\overline{\kappa}/4} \cdot \delta_{\widehat{Q}_2}^{|\alpha|+n/p-m} \cdot \sigma(\frac{65}{64}\widehat{Q}_1)$.
- $\partial^\beta P_\alpha(x_{\widehat{Q}_2}) = \delta_{\alpha\beta}$ for all $\beta \in \mathcal{A}''$.
- $|\partial^\beta P_\alpha(x_{\widehat{Q}_2})| \leq \epsilon^{\overline{\kappa}/4} \cdot \delta_{\widehat{Q}_2}^{|\alpha|-|\beta|}$ for all $\beta \in \mathcal{M}$, $\beta > \alpha$.
- $|\partial^\beta P_\alpha(x_{\widehat{Q}_2})| \leq C\Lambda \cdot \delta_{\widehat{Q}_2}^{|\alpha|-|\beta|}$ for all $\beta \in \mathcal{M}$.

The first and fourth bullet points imply that $\|(0, P_\alpha)\|_{\frac{65}{64}\widehat{Q}_1} \leq C\epsilon^{\overline{\kappa}/4}\delta_{\widehat{Q}_2}^{|\alpha|+n/p-m}$ and $|P_\alpha| \leq C\Lambda\Delta_0^{-C}$, hence (2.123) gives

$$M_{\widehat{Q}_1}(0, P_\alpha) \leq C'\epsilon^{\overline{\kappa}/4}\delta_{\widehat{Q}_2}^{|\alpha|+n/p-m} + C\,\Lambda\,\Delta_0^{-C}\,\Delta_{\text{new}} \leq C'\epsilon^{\overline{\kappa}/4}\,\delta_{\widehat{Q}_2}^{|\alpha|+n/p-m}, \tag{2.124}$$

which implies that

$$P_\alpha \in C'\epsilon^{\overline{\kappa}/4}\,\delta_{\widehat{Q}_2}^{|\alpha|+n/p-m} \cdot \overline{\overline{\sigma}}(\widehat{Q}_1) \quad \text{for } \alpha \in \mathcal{A}''. \tag{2.125}$$

Here, to prove (2.124), we use that $\delta_{\widehat{Q}_2} \leq 1$ and $|\alpha| + n/p - m < 0$, and

$$\Lambda \cdot \Delta_0^{-C}\Delta_{\text{new}} \overset{(2.82)}{\leq} \Lambda \cdot \Delta_{\text{new}}^{1/2} \overset{(2.121)}{\leq} \Lambda \cdot \epsilon^{2\kappa_2} \leq \epsilon^{\overline{\kappa}}.$$

Here, in the last inequality, we use that $\Lambda \leq \Lambda^{100D} \leq \epsilon^{-\overline{\kappa}/2}$, where $\overline{\kappa} \leq \kappa_2$.

With the second and third bullet points, (2.125) shows that $(P_\alpha)_{\alpha\in\mathcal{A}''}$ forms an $(\mathcal{A}'', x_{\widehat{Q}_2}, C\epsilon^{\overline{\kappa}/4}, \delta_{\widehat{Q}_2})$-basis for $\overline{\overline{\sigma}}(\widehat{Q}_1)$. Hence, by Lemma 26 of [3], it follows that $\overline{\overline{\sigma}}(\widehat{Q}_1)$ has an $(\mathcal{A}''', x_{\widehat{Q}_1}, \epsilon^{\kappa'}, \delta_{\widehat{Q}_2})$-basis for some $\mathcal{A}''' \leq \mathcal{A}'' \leq \mathcal{A}' \leq \mathcal{A}$ and for a small enough universal constant κ'. This completes the proof of Proposition 6. □

The statements and proofs of Propositions 11 and 12 of [4] are unchanged.

The statement of the algorithm OPTIMIZE BASIS requires modification.

ALGORITHM: OPTIMIZE BASIS (FINITE-PRECISION)

We perform one time work at most $CN \log N$ in space CN, after which we can answer queries as follows.

A query consists of a testing cube $\widehat{Q}$ and a set $\mathcal{A} \subset \mathcal{M}$

We respond to the query $(\widehat{Q}, \mathcal{A})$ by producing the following.

- A collection of machine intervals I_ℓ ($1 \le \ell \le \ell_{\max}$). The intervals I_ℓ are pairwise disjoint, the union of the I_ℓ is $[\Delta_g, \Delta_g^{-1}]$, and $\ell_{\max} \le C$.
- A list of non-negative machine numbers a_ℓ ($\ell = 1, \dots, \ell_{\max}$). The numbers a_ℓ are bounded in magnitude by Δ_g^{-C}.
- A list of numbers λ_ℓ. Each λ_ℓ has the form $\mu_\ell + \nu_\ell/p$, with $\mu_\ell, \nu_\ell \in \mathbb{Z}$ and $|\mu_\ell|, |\nu_\ell| \le C$.
- Let $\eta^{(\widehat{Q},\mathcal{A})}(\delta) := a_\ell\, \delta^{\lambda_\ell}$ for $\delta \in I_\ell$. Then we have:

  (A1) For each $\delta \in [\Delta_g, \Delta_g^{-1}]$ there exists $\mathcal{A}' \le \mathcal{A}$ such that $\overline{\overline{\sigma}}(\widehat{Q})$ has an $(\mathcal{A}', x_{\widehat{Q}}, \eta^{1/2}, \delta)$-basis for all $\eta > C \cdot \eta^{(\widehat{Q},\mathcal{A})}(\delta)$.

  (A2) For each $\delta \in [\Delta_g, \Delta_g^{-1}]$ and any $\mathcal{A}' \le \mathcal{A}$, $\overline{\overline{\sigma}}(\widehat{Q})$ *does not* have an $(\mathcal{A}', x_{\widehat{Q}}, \eta^{1/2}, \delta)$-basis for any $\eta < c \cdot \eta^{(\widehat{Q},\mathcal{A})}(\delta)$.

  (A3) Moreover, $c \cdot \eta^{(\widehat{Q},\mathcal{A})}(\delta_1) \le \eta^{(\widehat{Q},\mathcal{A})}(\delta_2) \le C \cdot \eta^{(\widehat{Q},\mathcal{A})}(\delta_1)$ whenever $\frac{1}{10}\delta_1 \le \delta_2 \le 10\delta_1$ and $\delta_1, \delta_2 \in [\Delta_g, \Delta_g^{-1}]$.

  (A4) Also, $\eta^{(\widehat{Q},\mathcal{A})}(\delta) \ge \Delta_g^C$, for all $\delta \in [\Delta_g, \Delta_g^{-1}]$.
- To answer a query requires work at most $C \log N$.

Explanation. Recall that we can perform arithmetic operations to within precision Δ_ϵ.

We denote $\mathbb{Z}[1/p] = \{\lambda = k + \ell \cdot \frac{1}{p} : k, \ell \in \mathbb{Z}\}$. If $\lambda = k + \ell \cdot \frac{1}{p} \in \mathbb{Z}[1/p]$, with k and ℓ bounded by a universal constant, then we say that λ is a machine element of $\mathbb{Z}[1/p]$. Such a λ can be stored on our computer using at most C units of storage.

Recall that we defined $|P|_x = \left(\sum_{\alpha \in \mathcal{M}} |\partial^\alpha P(x)|^p\right)^{1/p}$ for $P \in \mathcal{P}$ and $x \in \mathbb{R}^n$, and $|P| = |P|_0$. The vectors $(\partial^\alpha P(x))_{\alpha \in \mathcal{M}}$ and $(\partial^\alpha P(0))_{\alpha \in \mathcal{M}}$ are related by multiplication against an invertible matrix $A(x) = (A_{\alpha\beta}(x))_{\alpha,\beta \in \mathcal{M}}$. This is a consequence of Taylor's formula. Note that the operator norm of the matrix $A(x)$ is bounded by a universal constant if $|x| \le 1$. Thus,

$$C^{-1}\, |P|_x \le |P| \le C\, |P|_x \quad \text{for } P \in \mathcal{P} \text{ and } |x| \le 1. \tag{2.126}$$

Using APPROXIMATE NEW TRACE NORM (see Section 2.13.1), we compute a quadratic form $q_{\widehat{Q}}$ on $\mathcal{P}$ such that $\{q_{\widehat{Q}} \le c\} \subset \overline{\overline{\sigma}}(\widehat{Q}) \subset \{q_{\widehat{Q}} \le C\}$, where $c > 0$ and $C \ge 1$ are universal constants.[2] The quadratic form $q_{\widehat{Q}}$ is given in the form

$$q_{\widehat{Q}}(P) = \sum_{\alpha,\beta \in \mathcal{M}} \widetilde{q}_{\alpha\beta} \cdot \frac{1}{\alpha!}\partial^\alpha P(0) \cdot \frac{1}{\beta!}\partial^\beta P(0),$$

where we compute the numbers $\widetilde{q}_{\alpha\beta}$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. Using a linear change of basis, we write

$$q_{\widehat{Q}}(P) = \sum_{\alpha,\beta \in \mathcal{M}} q_{\alpha\beta} \cdot \frac{1}{\alpha!}\partial^\alpha P(x_{\widehat{Q}}) \cdot \frac{1}{\beta!} \cdot \partial^\beta P(x_{\widehat{Q}}).$$

[2] Recall that by now we have fixed t_G to be a universal constant; hence, the constants $c(t_G)$ and $C(t_G)$ in APPROXIMATE NEW TRACE NORM are now universal constants c, C.

Each $q_{\alpha\beta}$ is a linear combination of all the numbers $\tilde{q}_{\alpha\beta}$. Thus, we can compute each $q_{\alpha\beta}$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

From the conditions in the algorithm APPROXIMATE NEW TRACE NORM (finite-precision), we know that $q_{\widehat{Q}}(P) \geq c \cdot (M_{\widehat{Q}}(0,P))^2$. Furthermore, the term $\mathbf{(VI)} = \Delta_{\text{new}}^p \cdot |P|^p$ is a summand in $\left[M_{\widehat{Q}}(0,P)\right]^p$, hence $M_{\widehat{Q}}(0,P) \geq \Delta_{\text{new}} \cdot |P|$ (see (2.102)). Hence, using (2.126), we see that

$$q_{\widehat{Q}}(P) \geq c'\Delta_{\text{new}}^2 |P|^2 \geq c''\Delta_{\text{new}}^2 |P|_{x_{\widehat{Q}}}^2.$$

(Note that $|x_{\widehat{Q}}| \leq 1$, since $\widehat{Q} \subset Q^\circ = [0,1)^n$.)

Therefore, the matrix $(q_{\alpha\beta})$ satisfies

$$(q_{\alpha\beta}) \geq c\,\Delta_{\text{new}}^2 \cdot (\delta_{\alpha\beta}) \geq \Delta_g \cdot (\delta_{\alpha\beta}).$$

This means that we can apply the finite-precision version of FIT BASIS TO CONVEX BODY to the matrix $(q_{\alpha\beta})$ and the convex body $\overline{\overline{\sigma}}(\widehat{Q})$ (see Section 2.5). We can therefore compute a piecewise monomial function $\eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$ for each $\mathcal{A}' \leq \mathcal{A}$. We guarantee that

- For any $\delta \in [\Delta_g, \Delta_g^{-1}]$,
  - **(P1)** $\overline{\overline{\sigma}}(\widehat{Q})$ has an $(\mathcal{A}', x_{\widehat{Q}}, \eta^{1/2}, \delta)$-basis, for any $\eta > C \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$,
  - **(P2)** $\overline{\overline{\sigma}}(\widehat{Q})$ does not have an $(\mathcal{A}', x_{\widehat{Q}}, \eta^{1/2}, \delta)$-basis, for any $\eta < c \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$.
- **(P3)** Moreover, $c \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta_1) \leq \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta_2) \leq C \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta_1)$, whenever $\frac{1}{10}\delta_1 \leq \delta_2 \leq 10\delta_1$, for $\delta_1, \delta_2 \in [\Delta_g, \Delta_g^{-1}]$.
- **(P4)** Also, $\eta_*^{(\widehat{Q},\mathcal{A}')}(\delta) \geq \Delta_g^C$ for any $\delta \in [\Delta_g, \Delta_g^{-1}]$.
- The function $\eta_*^{(\widehat{Q},\mathcal{A}')} : [\Delta_g, \Delta_g^{-1}] \to \mathbb{R}$ is given in the form

$$\eta_*^{(\widehat{Q},\mathcal{A}')}(\delta) = a_{\ell,\mathcal{A}'} \cdot \delta^{\lambda_{\ell,\mathcal{A}'}} \quad \text{for } \delta \in I_{\ell,\mathcal{A}'}.$$

  To represent $\eta_*^{(\widehat{Q},\mathcal{A}')}$ we store the following data: pairwise disjoint machine intervals $I_{\ell,\mathcal{A}'}$ $(1 \leq \ell \leq \ell_{\max}(\mathcal{A}'))$ that form a partition of $[\Delta_g, \Delta_g^{-1}]$; machine numbers $a_{\ell,\mathcal{A}'} \in [\Delta_g^C, \Delta_g^{-C}]$; and exponents $\lambda_{\ell,\mathcal{A}'}$ that are machine elements of $\mathbb{Z}[1/p]$. We guarantee that $\ell_{\max}(\mathcal{A}') \leq C$ for each $\mathcal{A}' \leq \mathcal{A}$.

By computing all the nonempty intersections of the intervals $I_{\ell,\mathcal{A}'}$, we write each $\eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$ in the form

$$\eta_*^{(\widehat{Q},\mathcal{A}')}(\delta) = c_{\ell,\mathcal{A}'} \cdot \delta^{\gamma_{\ell,\mathcal{A}'}} \quad \text{for } \delta \in I_\ell \;\; (\ell = 1, 2, \ldots, \ell_{\max}).$$

Here, we compute the following: machine intervals I_ℓ $(1 \leq \ell \leq \ell_{\max})$ that partition $[\Delta_g, \Delta_g^{-1}]$; machine numbers $c_{\ell,\mathcal{A}'} \in [\Delta_g^C, \Delta_g^{-C}]$; and exponents $\gamma_{\ell,\mathcal{A}'}$ that are machine elements in $\mathbb{Z}[1/p]$. Moreover, $\ell_{\max} \leq C$.

We define

$$\eta(\delta) := \min_{\mathcal{A}' \le \mathcal{A}} \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta). \tag{2.127}$$

We will compute a piecewise-monomial approximation to the function $\eta(\delta)$ using the following procedure.

PROCEDURE: PROCESS MONOMIALS

Assume that we are given the following: A machine interval $I \subset [\Delta_g, \Delta_g^{-1}]$, machine numbers $a_1, a_2 \in [\Delta_g^C, \Delta_g^{-C}]$, and machine elements $\gamma_1, \gamma_2 \in \mathbb{Z}[1/p]$. We define monomial functions $m_1(\delta) = a_1\delta^{\gamma_1}$ and $m_2(\delta) = a_2\delta^{\gamma_2}$. Then we produce one of three outcomes:

1. We guarantee that $m_1(\delta) \le m_2(\delta) + \Delta_\epsilon^{1/2}$ for all $\delta \in I$.
2. We guarantee that $m_2(\delta) \le m_1(\delta) + \Delta_\epsilon^{1/2}$ for all $\delta \in I$.
3. We compute a machine number $\delta_* \in I$, and distinct indices $j, k \in \{1, 2\}$, such that
$$\begin{cases} m_j(\delta) \le m_k(\delta) + \Delta_\epsilon^{1/2} & \text{for } \delta \in I \cap (0, \delta_*], \\ m_k(\delta) \le m_j(\delta) + \Delta_\epsilon^{1/2} & \text{for } \delta \in I \cap [\delta_*, \infty). \end{cases}$$

This computation requires work and storage at most C.

Explanation. If $\gamma_1 = \gamma_2$ then outcome (1) occurs if $a_1 \le a_2$, and outcome (2) occurs if $a_1 > a_2$. Thus, we can respond in the case when $\gamma_1 = \gamma_2$

Assume instead that $\gamma_1 \ne \gamma_2$. In this case we have $c_0 \le |\gamma_1 - \gamma_2| \le C_0$ for universal constants c_0 and C_0, since γ_1 and γ_2 are machine elements in $\mathbb{Z}[1/p]$.[3]

We define a monomial function $m(\delta) := m_1(\delta)/m_2(\delta) = a{\cdot}\delta^\gamma$, where $a = a_1/a_2$ and $\gamma = \gamma_1 - \gamma_2$. The unique solution to the equation $m(\delta) = 1$ is given by

$$\delta_{\mathrm{sol}} := a^{1/\gamma}. \tag{2.128}$$

Since monomial functions are monotonic, we have either

(a) $m(\delta) < 1$ for $\Delta_g \le \delta < \delta_{\mathrm{sol}}$, and $m(\delta) > 1$ for $\delta_{\mathrm{sol}} < \delta \le \Delta_g^{-1}$; or

(b) $m(\delta) > 1$ for $\Delta_g \le \delta < \delta_{\mathrm{sol}}$, and $m(\delta) < 1$ for $\delta_{\mathrm{sol}} < \delta \le \Delta_g^{-1}$.

We know that (a) holds if $\gamma > 0$, and (b) holds if $\gamma < 0$. We can determine which case occurs because the rational number γ is given to exact precision.

Since $a \in [\Delta_g^C, \Delta_g^{-C}]$ and $c_0 \le |\gamma| \le C_0$, due to the numerical stability of exponentiation we can compute a machine number δ_* such that

$$\delta_* \in [\Delta_g^C, \Delta_g^{-C}] \text{ and } |\delta_* - \delta_{\mathrm{sol}}| \le \Delta_g^{-C}\Delta_\epsilon \quad \text{(see (2.128))}. \tag{2.129}$$

From (2.129) and the Lipschitz continuity of $m(\delta)$, we have $|m(\delta) - 1| \le \Delta_g^{-C}\Delta_\epsilon$ for all δ in the interval between δ_* and δ_{sol}.

[3]The constant c_0 here depends only on m, n, and p, but it may depend sensitively on the approximation of $1/p$ by rationals with low denominators.

Therefore, in case (a) we have $m(\delta) = m_1(\delta)/m_2(\delta) \le 1 + \Delta_g^{-C}\Delta_\epsilon$ for all $\Delta_g \le \delta \le \delta_*$, and $m(\delta) = m_1(\delta)/m_2(\delta) \ge 1 - \Delta_g^{-C}\Delta_\epsilon$ for all $\delta_* \le \delta \le \Delta_g^{-1}$. Note that both $m_1(\delta)$ and $m_2(\delta)$ are in the range $[\Delta_g^C, \Delta_g^{-C}]$ if $\delta \in [\Delta_g, \Delta_g^{-1}]$. Thus, in case (a) we determine that

$$m_1(\delta) \le m_2(\delta)\cdot(1+\Delta_g^{-C}\Delta_\epsilon) \le m_2(\delta)+\Delta_g^{-2C}\Delta_\epsilon \le m_2(\delta)+\Delta_\epsilon^{1/2} \quad \text{for } \Delta_g \le \delta \le \delta_*,$$

and similarly, $m_1(\delta) \ge m_2(\delta) - \Delta_\epsilon^{1/2}$ for $\delta_* \le \delta \le \Delta_g^{-1}$. Thus, we can respond as follows:

- If δ_* is to the left of the interval I then outcome (2) occurs.
- If δ_* is to the right of the interval I then outcome (1) occurs.
- If δ_* belongs to the interval I then outcome (3) occurs with $j = 1$ and $k = 2$.

Similarly, in case (b) we determine that $m_1(\delta) \ge m_2(\delta) - \Delta_\epsilon^{1/2}$ for all $\Delta_g \le \delta \le \delta_*$, and similarly, $m_1(\delta) \le m_2(\delta) + \Delta_\epsilon^{1/2}$ for all $\delta_* \le \delta \le \Delta_g^{-1}$. Thus, we can respond as follows:

- If δ_* is to the left of the interval I then outcome (1) occurs.
- If δ_* is to the right of the interval I then outcome (2) occurs.
- If δ_* belongs to the interval I then outcome (3) occurs with $j = 2$ and $k = 1$.

This completes the explanation of procedure PROCESS MONOMIALS. Clearly, the work and storage of this algorithm are at most C for a universal constant C. □

We return to the setting before the above procedure.

Fix $\ell \in \{1, \dots, \ell_{\max}\}$. Applying the procedure PROCESS MONOMIALS, for each pair $(\mathcal{A}', \mathcal{A}'')$ such that $\mathcal{A}' \le \mathcal{A}$ and $\mathcal{A}'' \le \mathcal{A}$, we produce one of three outcomes.

In outcome (1), we guarantee that $\eta_*^{(\widehat{Q},\mathcal{A}')} \le \eta_*^{(\widehat{Q},\mathcal{A}'')} + \Delta_\epsilon^{1/2}$, uniformly on the interval I_ℓ.

In outcome (2), we guarantee that $\eta_*^{(\widehat{Q},\mathcal{A}'')} \le \eta_*^{(\widehat{Q},\mathcal{A}')} + \Delta_\epsilon^{1/2}$, uniformly on the interval I_ℓ.

In outcome (3), we divide the interval I_ℓ at the point $\delta_{\ell,\mathcal{A}',\mathcal{A}''} = \delta_* \in I_\ell$ to obtain *split subintervals* $I_\ell^- = I_\ell \cap (0, \delta_*]$ and $I_\ell^+ = I_\ell \cap (\delta_*, \infty)$. (A subinterval may contain only a single point or be empty.) We guarantee that $\eta_*^{(\widehat{Q},\mathcal{A}')} \le \eta_*^{(\widehat{Q},\mathcal{A}'')} + \Delta_\epsilon^{1/2}$ on one of the split subintervals, and $\eta_*^{(\widehat{Q},\mathcal{A}'')} \le \eta_*^{(\widehat{Q},\mathcal{A}')} + \Delta_\epsilon^{1/2}$ on the other. We determine which inequality is satisfied on each subinterval.

For each pair $(\mathcal{A}', \mathcal{A}'')$ such that outcome (3) occurs, we have computed a machine number $\delta_{\ell,\mathcal{A}',\mathcal{A}''}$ in I_ℓ. We sort these numbers and remove duplicates to obtain a list

$$\delta_1 < \delta_2 < \cdots < \delta_{K_\ell}.$$

Note that $K_\ell \le \#\{(\mathcal{A}', \mathcal{A}'') : \mathcal{A}' \le \mathcal{A},\ \mathcal{A}'' \le \mathcal{A}\} \le C$ for a universal constant C. We define δ_0 and $\delta_{K_\ell+1}$ to be the left and right endpoints of I_ℓ, respectively. We let $I_\ell^k := [\delta_k, \delta_{k+1}]$ for each $0 \le k \le K_\ell$. We thus obtain a (possibly trivial) partition of I_ℓ into subintervals $I_\ell^0, \dots, I_\ell^{K_\ell}$.

For each interval I_ℓ^k and each pair $(\mathcal{A}', \mathcal{A}'')$ such that $\mathcal{A}' \le \mathcal{A}$ and $\mathcal{A}'' \le \mathcal{A}$, we guarantee either that $\eta_*^{(\widehat{Q},\mathcal{A}')} \le \eta_*^{(\widehat{Q},\mathcal{A}'')} + \Delta_\epsilon^{1/2}$ on I_ℓ^k ($\mathcal{A}'$ beats $\mathcal{A}''$ on I_ℓ^k), or that $\eta_*^{(\widehat{Q},\mathcal{A}'')} \le \eta_*^{(\widehat{Q},\mathcal{A}')} + \Delta_\epsilon^{1/2}$ on I_ℓ^k ($\mathcal{A}''$ beats $\mathcal{A}'$ on I_ℓ^k). To make such a guarantee, we look at the previous outcomes. If outcome (1) occurs, then $\mathcal{A}'$ beats $\mathcal{A}''$ on I_ℓ^k. If outcome (2) occurs, then $\mathcal{A}''$ beats $\mathcal{A}'$ on I_ℓ^k. If outcome (3) occurs, then we determine which of the split subintervals of I_ℓ contains I_ℓ^k. Once we have done that, we can make a correct guarantee by using the guarantee made in outcome (3) for the split subinterval.

For each of the intervals I_ℓ^k we perform the following computation. We initialize $\mathcal{S} = \{\mathcal{A}' \subset \mathcal{M} : \mathcal{A}' \le \mathcal{A}\}$. We initialize $\overline{\mathcal{A}}$ to be any member of $\mathcal{S}$. Then we run the following loop.

- WHILE: $\mathcal{S} \neq \{\overline{\mathcal{A}}\}$
- – Select an arbitrary $\mathcal{A}' \in \mathcal{S} \setminus \{\overline{\mathcal{A}}\}$.
  – If we guarantee that $\mathcal{A}'$ beats $\overline{\mathcal{A}}$ on I_ℓ^k, then discard $\overline{\mathcal{A}}$ from $\mathcal{S}$ and set $\overline{\mathcal{A}} = \mathcal{A}'$.
  – If we guarantee that $\overline{\mathcal{A}}$ beats $\mathcal{A}'$ on I_ℓ^k, then discard $\mathcal{A}'$ from $\mathcal{S}$. Do not modify $\overline{\mathcal{A}}$.
  – (Note that we make at most one guarantee.)

Let $\mathcal{A}_1$ denote the sole member remaining in $\mathcal{S}$ once the loop is complete. For any $\mathcal{A}' \subset \mathcal{M}$ with $\mathcal{A}' \le \mathcal{A}$, there is a sequence of "competitors" $\mathcal{A}_2, \dots, \mathcal{A}_J$ with $\mathcal{A}_J = \mathcal{A}'$, such that $\mathcal{A}_j$ beats $\mathcal{A}_{j+1}$ on I_ℓ^k for $j = 1, \dots, J-1$. This is clear because $\mathcal{A}'$ is selected in the loop at some iteration, and as long as $\mathcal{A}' \neq \mathcal{A}_1$ we can be certain that $\mathcal{A}'$ is beaten by some competitor, who in turn is beaten by another competitor, and so on until the loop terminates with the final competitor $\mathcal{A}_1$. Clearly, the number of competitors J is bounded by a universal constant C. By combining the estimates coming from each competition, we learn that $\eta_*^{(\widehat{Q},\mathcal{A}_1)} \le \eta_*^{(\widehat{Q},\mathcal{A}_J)} + J\Delta_\epsilon^{1/2}$. Therefore, $\eta_*^{(\widehat{Q},\mathcal{A}_1)} \le \eta_*^{(\widehat{Q},\mathcal{A}')} + C\,\Delta_\epsilon^{1/2}$.

Thus, for each $0 \le k \le K_\ell$, we can compute a multiindex set $\mathcal{A}_\ell^k \le \mathcal{A}$ such that

$$\eta_*^{(\widehat{Q},\mathcal{A}_\ell^k)}(\delta) \le \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta) + C\,\Delta_\epsilon^{1/2} \quad \text{for all } \delta \in I_\ell^k, \ \text{ for all } \mathcal{A}' \le \mathcal{A}. \tag{2.130}$$

We repeat the previous construction for each $\ell \in \{1, \dots, \ell_{\max}\}$.

We therefore obtain machine intervals I_ℓ^k ($0 \le k \le K_\ell$, $1 \le \ell \le \ell_{\max}$), which form a partition of $[\Delta_g, \Delta_g^{-1}]$, and multiindex sets $\mathcal{A}_\ell^k$ as in (2.130).

We define a function $\widetilde{\eta} : [\Delta_g, \Delta_g^{-1}] \to \mathbb{R}$ by

$$\widetilde{\eta}(\delta) := \eta_*^{(\widehat{Q},\mathcal{A}_\ell^k)}(\delta) = c_{\ell,\mathcal{A}_\ell^k} \cdot \delta^{\lambda_{\ell,\mathcal{A}_\ell^k}} \quad \text{if } \delta \in I_\ell^k.$$

Since $\ell_{\max} \le C$, the previous construction can be executed using work and storage at most a universal constant C'.

We will make use of the properties **(P1)**–**(P4)** of the functions $\eta_*^{(\widehat{Q},\mathcal{A}')}$ that were stated earlier in this section.

Recall that $\eta(\delta)$ is the minimum of $\eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$ over all $\mathcal{A}' \leq \mathcal{A}$ (see (2.127)). Since $\mathcal{A}_\ell^k \leq \mathcal{A}$ for all (k,ℓ), we have $\widetilde{\eta}(\delta) \geq \eta(\delta)$. Moreover, taking the minimum with respect to $\mathcal{A}'$ in (2.130), we conclude that $\widetilde{\eta}(\delta) \leq \eta(\delta) + C\,\Delta_\epsilon^{1/2}$. Thanks to **(P4)**, we have $\eta(\delta) \geq \Delta_g^C \geq C\,\Delta_\epsilon^{1/2}$. Thus, we learn that

$$\eta(\delta) \leq \widetilde{\eta}(\delta) \leq 2 \cdot \eta(\delta). \tag{2.131}$$

We next prove that the the function $\eta^{(\widehat{Q},\mathcal{A})}(\delta) = \widetilde{\eta}(\delta)$ satisfies **(A1)**–**(A4)**.

Proof of **(A1)**. Let $\delta \in [\Delta_g, \Delta_g^{-1}]$. Also, let $\eta > C \cdot \widetilde{\eta}(\delta)$, with C as in **(P1)**. Then, thanks to (2.131), we have

$$\eta > C \cdot \eta(\delta) = C \cdot \min_{\mathcal{A}' \leq \mathcal{A}} \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta).$$

Hence, $\eta > C \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$ for some $\mathcal{A}' \leq \mathcal{A}$. According to **(P1)**, we learn that $\overline{\overline{\sigma}}(\widehat{Q})$ has an $(\mathcal{A}', x_{\widehat{Q}}, \eta^{1/2}, \delta)$-basis. This completes the proof of **(A1)**.

Proof of **(A2)**. Let $\delta \in [\Delta_g, \Delta_g^{-1}]$. Also, let $\eta < \frac{c}{2} \cdot \widetilde{\eta}(\delta)$, with $c > 0$ as in **(P2)**. Then, thanks to (2.131), we have

$$\eta \leq c \cdot \eta(\delta) = c \cdot \min_{\mathcal{A}' \leq \mathcal{A}} \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta).$$

Hence, $\eta < c \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta)$ for all $\mathcal{A}' \leq \mathcal{A}$. According to **(P2)**, we learn that $\overline{\overline{\sigma}}(\widehat{Q})$ does not have an $(\mathcal{A}', x_{\widehat{Q}}, \eta^{1/2}, \delta)$-basis for any $\mathcal{A}' \leq \mathcal{A}$. This completes the proof of **(A2)**.

Proof of **(A3)**. Let $\delta_1, \delta_2 \in [\Delta_g, \Delta_g^{-1}]$, with $\frac{1}{10}\delta_1 \leq \delta_2 \leq 10\delta_1$.

According to **(P3)**, for each $\mathcal{A}' \leq \mathcal{A}$ we have

$$c \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta_1) \leq \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta_2) \leq C \cdot \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta_1).$$

Taking the minimum with respect to $\mathcal{A}' \leq \mathcal{A}$, we learn that $c \cdot \eta(\delta_1) \leq \eta(\delta_2) \leq C \cdot \eta(\delta_1)$. According to (2.131), we therefore have $\frac{1}{4}c \cdot \widetilde{\eta}(\delta_1) \leq \widetilde{\eta}(\delta_2) \leq 4C \cdot \widetilde{\eta}(\delta_1)$. This completes the proof of **(A3)**.

Proof of **(A4)**. We have

$$\widetilde{\eta}(\delta) \overset{(2.131)}{\geq} \eta(\delta) = \min_{\mathcal{A}' \leq \mathcal{A}} \eta_*^{(\widehat{Q},\mathcal{A}')}(\delta) \overset{(P4)}{\geq} \Delta_g^C.$$

This completes the proof of **(A4)**.

Thus, conditions **(A1)**, **(A2)**, **(A3)**, and **(A4)** hold for $\eta^{(\widehat{Q},\mathcal{A})}(\delta) = \widetilde{\eta}(\delta)$.

This concludes the explanation of the algorithm. □

2.15. Computing lengthscales

Each point $x \in E$ is assumed to be an $\overline{S}$-bit machine point. Recall that $\Delta_0 = 2^{-\overline{S}}$. Hence,

$$|x' - x''| \geq \Delta_0 \quad \text{for distinct } x', x'' \in E. \tag{2.132}$$

Recall that $CZ(\mathcal{A}^-)$ consists of disjoint dyadic cubes that form a partition of $Q^\circ = [0,1)^n$. According to the Main Technical Results for $\mathcal{A}^-$, we have $\delta_Q \geq c \cdot \Delta_0$ for each Q in $CZ(\mathcal{A}^-)$, for a universal constant c. Therefore, each Q in $CZ(\mathcal{A}^-)$ is an $\widetilde{S}$-bit machine cube, where $\widetilde{S} \leq C\overline{S}$ for a universal constant C.

Recall that a testing cube is a dyadic cube $\widehat{Q} \subset Q^\circ$ that can be written as a disjoint union of cubes in $CZ(\mathcal{A}^-)$. We then have $\delta_{\widehat{Q}} \geq c \cdot \Delta_0$ for a universal constant $c > 0$ (see Remark 1). We set $\lambda := 1/40$.

ALGORITHM: COMPUTE INTERESTING CUBES (FINITE-PRECISION)

We compute a tree T consisting of testing cubes. The nodes in T consist of all the cubes $Q \in CZ(\mathcal{A}^-)$ that contain points of E, all the testing cubes $\widehat{Q}$ for which $\operatorname{diam}(3\widehat{Q} \cap E) \geq \lambda \cdot \delta_{\widehat{Q}}$, and the unit cube Q°.

Here, T is a tree with respect to inclusion. We mark each internal node Q in T with pointers to its children, and we mark each node Q in T (except for the root) with a pointer to its parent.

The number of nodes in T is at most CN, and T can be computed with work at most $CN \log N$ in space CN.

Explanation. We follow the explanation in Section 1.6 of [4]. We need to check that the computation is valid in our finite-precision model of computation.

We compute representative pairs from the well-separated pairs decomposition of E using the algorithm MAKE WSPD (see Section 4.2 of [3]). The representative pairs $(x'_\nu, x''_\nu) \in E \times E \setminus \{(x,x) : x \in E\}$ $(1 \leq \nu \leq \nu_{\max})$ satisfy $|x'_\nu - x''_\nu| \geq \Delta_0$, thanks to (2.132).

Next, we loop over all ν and list all the dyadic cubes $\widetilde{Q}$ with $x'_\nu, x''_\nu \in 5\widetilde{Q}$ and $|x'_\nu - x''_\nu| \geq \frac{\lambda}{2}\delta_{\widetilde{Q}}$. We call this list $Q_1, \ldots, Q_K$. Since $5Q_k$ contains some representative pair (x'_ν, x''_ν), we have $\delta_{Q_k} \geq \frac{1}{5}|x'_\nu - x''_\nu| \geq \frac{1}{5}\Delta_0$ for each $k = 1, \ldots, K$.

Note that the "BBD Tree algorithm" in Theorem 35 of [3] is unchanged in finite-precision, so we can compute $\operatorname{diam}(3Q_k \cap E)$ for each $k = 1, \ldots, K$. We remove any cubes from our list that satisfy $\operatorname{diam}(3Q_k \cap E) < \lambda\delta_{Q_k}$, which occurs if and only if $\delta_{Q_k} > 40 \cdot \operatorname{diam}(3Q_k \cap E)$. We also compute the cube in $CZ(\mathcal{A}^-)$ that contains the center of each Q_k, using the $CZ(\mathcal{A}^-)$-ORACLE. If Q_k is strictly contained in this cube, then we remove Q_k from our list. Denote the surviving cubes by $\widetilde{Q}_1, \ldots, \widetilde{Q}_{\widetilde{K}}$.

We list all the cubes $Q \in CZ(\mathcal{A}^-)$ that contain points of E (take all the cubes Q in $CZ_{\text{main}}(\mathcal{A}^-)$ that satisfy $E \cap Q \neq \emptyset$), the cubes $\widetilde{Q}_1, \ldots, \widetilde{Q}_{\widetilde{K}}$, and the unit cube Q°. We sort this list to remove duplicates, and organize it in a tree T using the algorithm MAKE FOREST (see Section 4.1.5 of [3]). That completes the explanation of the algorithm. □

ALGORITHM: COMPUTE CRITICAL TESTING CUBES (FINITE-PRECISION)

Given $\epsilon > 0$, which is less than a small enough universal constant, we produce a collection $\widehat{\mathcal{Q}}_\epsilon$ of testing cubes with the following properties.

(a) Each point $x \in E$ belongs to some cube $\widehat{Q}_x \in \widehat{\mathcal{Q}}_\epsilon$.

(b) The cardinality of $\widehat{\mathcal{Q}}_\epsilon$ is at most $C \cdot N$.

(c) If $\widehat{Q} \in \widehat{\mathcal{Q}}_\epsilon$ strictly contains a cube in $CZ(\mathcal{A}^-)$, then $(1 + a(\mathcal{A}))\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^\kappa)$.

(d) If $\widehat{Q} \in \widehat{\mathcal{Q}}_\epsilon$ and $\delta_{\widehat{Q}} \leq c^*$, then no cube containing $S\widehat{Q}$ is tagged with $(\mathcal{A}, \epsilon^{1/\kappa})$.

(e) Each cube Q in $\mathcal{Q}_\epsilon$ satisfies $\delta_Q \geq c \cdot \Delta_0$.

The algorithm requires work at most $CN \log N$ in space CN.

Here, $c^* > 0$ and $S \geq 1$ are integer powers of 2, which depend only on m, n, p; also, $\kappa \in (0, 1)$ and $C \geq 1$ are universal constants.

Explanation. The main change to the explanation is that we use the finite-precision version of OPTIMIZE BASIS instead of the infinite-precision version. We also need to show that the roundoff errors that can arise have little effect.

Note that condition (e) will hold for each Q in $\mathcal{Q}_\epsilon$, since we promise that $\mathcal{Q}_\epsilon$ contains only testing cubes. (See Remark 1.)

We let $\Lambda \geq 1$ be a sufficiently large integer power of two, as before. We will later choose Λ to be bounded by a universal constant, but not yet. We assume that Λ is a machine number.

We construct a tree T of interesting cubes with the algorithm COMPUTE INTERESTING CUBES (finite-precision).

We next explain the construction of the collection $\widehat{\mathcal{Q}}_\epsilon$.

We proceed with Steps 0-6. The construction is almost identical to that in infinite-precision. We refer the reader to the earlier text. We will only record the necessary changes

We assume we have carried out the one-time work of the BBD Tree. Thus, given an $\widetilde{S}$-bit machine cube Q, with $\widetilde{S} \leq C\overline{S}$, we can compute $\#\left(\frac{65}{64}Q \cap E\right)$ using work at most $C \cdot \log N$.

Therefore, we can compute $\#\left(\frac{65}{64}Q \cap E\right)$ for each Q in T.

For each cube Q_1 in T we perform Steps 0-3.

Step 0 is unchanged: We find the parent Q_2 of Q_1 in T.

In Step 1 in the infinite-precision text, it says "We determine whether or not there exists a number $\delta \in [\Lambda^{10}\delta_{Q_1}, \Lambda^{-10}\delta_{Q_2}]$ with the property that $\epsilon^{1/\kappa_5} \leq \eta^{(Q_1^{up}, \mathcal{A})}(\delta) \leq \epsilon^{\kappa_5}$. If such a δ exists, we can easily find one." We can no longer make such an accurate determination because of inevitable roundoff errors. We will need to make the modifications listed below.

• *Step* 1 (*modified*). In finite-precision, we compute a piecewise-monomial representation for the function $\eta^{(Q_1^{up}, \mathcal{A})}(\delta)$ using the finite-precision version of OPTIMIZE BASIS, where Q_1^{up} is the dyadic cube with $Q_1 \subset Q_1^{up}$ and $\delta_{Q_1^{up}} = \Lambda \cdot \delta_Q$.

We produce one of two outcomes. Either we guarantee that there does not exist a $\delta \in [\Lambda^{10}\delta_{Q_1}, \Lambda^{-10}\delta_{Q_2}]$ such that

$$\epsilon^{1/\kappa_5} \le \eta^{(Q_1^{up},\mathcal{A})}(\delta) \le \epsilon^{\kappa_5}, \tag{2.133}$$

or else we compute a machine number $\delta \in [\Lambda^{10}\delta_{Q_1}, \Lambda^{-10}\delta_{Q_2}]$ satisfying

$$\frac{1}{2}\epsilon^{1/\kappa_5} \le \eta^{(Q_1^{up},\mathcal{A})}(\delta) \le 2\epsilon^{\kappa_5}. \tag{2.134}$$

The number δ is computed *exactly*.

The factors of 2 in the above estimate arise because of roundoff errors in the computation of δ. Indeed, we can bound any roundoff error by $\Delta_\epsilon\, \Delta_g^{-C}$, which is at most $100^{-1} \cdot \epsilon^{1/\kappa_5}$, since $\Delta_\epsilon\, \Delta_g^{-C} \le \Delta_\epsilon^{1/2}$ (see (2.81)) and $\Delta_\epsilon^{1/2} \le \Delta_{new} \le 100^{-1} \cdot \epsilon^{1/\kappa_5}$ (see (2.82) and (2.121)).

As in the proof of (1.159) of [4], in the second alternative we can find a dyadic cube Q with $Q_1 \subset Q \subset Q_2$, $\Lambda^{10}\delta_{Q_1} \le \delta_Q \le \Lambda^{-10}\delta_{Q_2}$, and such that

$$\left[\epsilon^{1/\kappa_6} \le \eta^{(Q_1^{up},\mathcal{A})}(\delta_Q)\right] \quad \text{and} \quad \left[\eta^{(Q_1^{up},\mathcal{A})}(\delta_Q) \le \epsilon^{\kappa_6}\right]. \tag{2.135}$$

Here, by choosing κ_6 sufficiently small, we can make the extra factors of 2 disappear.

In the second alternative, we add Q to the collection $\widehat{Q}_\epsilon$. That completes the computation in Step 1.

Note that $[\delta_{Q_1}, \delta_{Q_2}] \subset [\Delta_g, \Delta_g^{-1}]$, since each cube in T has sidelength in $[c \cdot \Delta_0, 1]$, and since $\Delta_g \le c \cdot \Delta_0$. This comment justifies the previous computation, since the function $\eta^{(Q_1^{up},\mathcal{A})}(\delta)$ is defined only for $\delta \in [\Delta_g, \Delta_g^{-1}]$.

Similarly, in Steps 2–6, we make the following changes.

• *Step* 2 (*modified*). We examine each dyadic cube Q with $Q_1 \subset Q \subset Q_2$, $\delta_Q \le \Lambda^{-10}$, and $[\delta_Q \le \Lambda^{10}\delta_{Q_1}$ or $\delta_Q \ge \Lambda^{-10}\delta_{Q_2}]$. We compute a piecewise-monomial function $\eta^{(Q^{up},\mathcal{A})}(\delta)$ using the finite-precision version of OPTIMIZE BASIS. We produce one of two outcomes. Either we guarantee that

$$\left[\epsilon^{\kappa_5^{-1}} > \eta^{(Q^{up},\mathcal{A})}(\delta_{Q^{up}})\right] \text{ or } \left[\#\left(\frac{65}{64}Q \cap E\right) \ge 2 \text{ and } \eta^{(Q,\mathcal{A})}(\delta_Q) > \epsilon^{\kappa_5}\right], \tag{2.136}$$

where Q^{up} is the unique dyadic cube with $Q \subset Q^{up}$ and $\delta_{Q^{up}} = \Lambda\delta_Q$, or else we guarantee that

$$\left[\tfrac{1}{2}\epsilon^{\kappa_5^{-1}} \le \eta^{(Q^{up},\mathcal{A})}(\delta_{Q^{up}})\right] \text{ and } \left[\#\left(\frac{65}{64}Q \cap E\right) \le 1 \text{ or } \eta^{(Q,\mathcal{A})}(\delta_Q) \le 2\epsilon^{\kappa_5}\right]. \tag{2.137}$$

The extra factors of 2 allow for small additive errors in the computation of $\eta^{(Q^{up},\mathcal{A})}(\delta_{Q^{up}})$.

We add Q to the collection $\widehat{Q}_\epsilon$ in the second alternative.

• *Step* 3 (*modified*). We examine each dyadic cube Q with $Q_1 \subset Q \subset Q_2$ and $\delta_Q \geq \Lambda^{-10}$. We apply the finite-precision version of OPTIMIZE BASIS to compute a function $\eta^{(Q,\mathcal{A})}(\delta)$. We produce one of two outcomes. Either we guarantee that

$$\Big[\#\Big(\frac{65}{64}Q \cap E\Big) \geq 2 \text{ and } \eta^{(Q,\mathcal{A})}(\delta_Q) > \epsilon^{\kappa_5}\Big], \tag{2.138}$$

or else we guarantee that

$$\Big[\#\Big(\frac{65}{64}Q \cap E\Big) \leq 1 \text{ or } \eta^{(Q,\mathcal{A})}(\delta_Q) \leq 2\epsilon^{\kappa_5}\Big]. \tag{2.139}$$

The extra factors of 2 allow for roundoff errors in the computation of $\eta^{(Q,\mathcal{A})}(\delta_Q)$.

We add Q to the collection $\widehat{Q}_\epsilon$ in the second alternative.

• *Step* 4 (*modified*). We apply the finite-precision version of OPTIMIZE BASIS to compute a function $\eta^{(Q^\circ,\mathcal{A})}(\delta)$. We produce one of two outcomes. Either we guarantee that

$$\big[\eta^{(Q^\circ,\mathcal{A})}(\delta_{Q^\circ}) > \epsilon^{\kappa_5}\big], \tag{2.140}$$

or else we guarantee that

$$\big[\eta^{(Q^\circ,\mathcal{A})}(\delta_{Q^\circ}) \leq 2\epsilon^{\kappa_5}\big]. \tag{2.141}$$

We add Q° to the collection $\widehat{Q}_\epsilon$ in the second alternative.

• *Step* 5 (*modified*). We examine all dyadic cubes $Q \subset Q^\circ$ such that $\delta_Q \geq \Lambda^{-10}$. We add Q to the collection $\widehat{\mathcal{Q}}_\epsilon$ if and only if $Q \in \mathrm{CZ}(\mathcal{A}^-)$.

• *Step* 6 (*modified*). We examine all cubes $Q \in \mathrm{CZ}(\mathcal{A}^-)$ such that $\delta_Q \leq \Lambda^{-10}$ and $Q \cap E \neq \emptyset$. We apply the finite-precision version of OPTIMIZE BASIS to compute a function $\eta^{(Q^{\mathrm{up}},\mathcal{A})}(\delta)$, where Q^{up} is the dyadic cube with $Q \subset Q^{\mathrm{up}}$ and $\delta_{Q^{\mathrm{up}}} = \Lambda\delta_Q$. We produce one of two outcomes. Either we guarantee that

$$\big[\epsilon^{\kappa_5^{-1}} > \eta^{(Q^{\mathrm{up}},\mathcal{A})}(\delta_{Q^{\mathrm{up}}})\big], \tag{2.142}$$

or else we guarantee that

$$\big[\tfrac{1}{2}\epsilon^{\kappa_5^{-1}} \leq \eta^{(Q^{\mathrm{up}},\mathcal{A})}(\delta_{Q^{\mathrm{up}}})\big]. \tag{2.143}$$

We add Q to the collection $\widehat{Q}_\epsilon$ in the second alternative.

As before, we see that $\#(\widehat{\mathcal{Q}}_\epsilon) \leq C(\Lambda) \cdot N$, hence property (b) holds.

Recall that Propositions 8 and 12 of [4] are unchanged in the finite-precision case – only their proofs required modification. Hence, the analysis that the above algorithm works proceeds as before. In place of the conditions (1.159)–(1.163) we use the conditions (2.135), (2.137), (2.139), (2.141), and (2.143).

The proof of properties (c) and (d) requires minor changes to reflect the loss of factors of 2. By choosing smaller values for $\kappa_1, \ldots, \kappa_{20}$, we arrange that the extra factors of 2 can be absorbed into relevant estimates in the proof. Thus, we can prove properties (c) and (d) for each cube in $\mathcal{Q}_\epsilon$ using the same argument as before.

The proof of property (a) requires minor changes to reflect the loss of factors of 2. We fix a point $x \in E$.

As before, we consider the increasing chain of cubes $Q_0 \subset Q_1 \subset \cdots \subset Q_{\nu_{\max}}$ in T, such that $Q_{\ell+1}$ is a parent of Q_ℓ in T, $x \in Q_0$, and $Q_0 \in \mathrm{CZ}(\mathcal{A}^-)$.

To prove (a), we will show that there exists a cube $Q \in \widehat{\mathcal{Q}}_\epsilon$ such that $x \in Q$.

As before, we consider the first extreme case (A), the second extreme case (B), and the main case (C).

In the *first extreme case*, we assume that $3Q^\circ$ is tagged with $(\mathcal{A}, \epsilon)$ and deduce that $\eta^{(Q^\circ,\mathcal{A})}(\delta_{Q^\circ}) \leq \epsilon^{\kappa_5}$. Hence, according to the above construction in Step 4, we included Q° in $\widehat{\mathcal{Q}}_\epsilon$.

In the *second extreme case*, we assume that $3Q_0$ is not tagged with $(\mathcal{A}, \epsilon)$ and we deduce that $\eta^{(Q_0^{\mathrm{up}},\mathcal{A})}(\delta_{Q_0^{\mathrm{up}}}) \geq \epsilon^{1/\kappa_5}$. Hence, according to the construction in Step 6, we included Q_0 in $\widehat{\mathcal{Q}}_\epsilon$.

In the *GI subcase* of the *main case*, from the assumptions in the GI subcase we prove (1.172) and (1.173) of [4] (see the analysis in infinite-precision). This means that (2.136) does not hold for the cube Q. Hence, according to the construction in Step 2, we included Q in $\widehat{\mathcal{Q}}_\epsilon$.

In the *GUI subcase* of the *main case*, from the assumptions in the GUI subcase we prove (1.174) and (1.175) of [4]. Hence, (2.133) holds with $\delta = \delta_Q$. Thus, we pass to the second alternative in our construction in Step 1 (for the cube $Q_\nu \in T$). Hence, we decided to include in $\widehat{\mathcal{Q}}_\epsilon$ a cube Q' with $Q_\nu \subset Q' \subset Q_{\nu+1}$.

In the *NM subcase* of the *main case*, from the assumptions in the NM subcase we prove (1.176) of [4]. Hence, (2.138) fails to hold for the cube Q. Hence, in the construction in Step 3, we included Q in $\widehat{\mathcal{Q}}_\epsilon$.

Thus, as in infinite-precision, we see that there exists $Q' \in \widehat{\mathcal{Q}}_\epsilon$ with $Q_0 \subset Q' \subset Q_{\nu_{\max}}$, and hence $x \in Q'$. This completes the proof of (a).

We choose $\Lambda \geq 1$ to a be a large enough universal constant so that the above holds. That concludes the explanation of the algorithm. □

According to our construction, each Q in $\mathcal{Q}_\epsilon$ satisfies $\delta_Q \geq c \cdot \Delta_0$. Furthermore, by hypothesis, each $x \in E$ is an $\overline{S}$-bit machine point.

Thus, we can apply the algorithm PLACING A POINT INSIDE TARGET CUBOIDS to compute a cube $Q_x \in \mathcal{Q}_\epsilon$ containing each $x \in E$. This requires work at most $CN \log N$ in space CN. Thus, the algorithm COMPUTE LENGTHSCALES is unchanged in finite-precision (see Section 1.6.2 of [4]).

Proposition 13 of [4] still holds in the finite-precision setting.

2.16. Passing from lengthscales to CZ decompositions

We explain how to define a decomposition $\mathrm{CZ}(\mathcal{A})$ of Q° into machine cubes, and how to define a $\mathrm{CZ}(\mathcal{A})$-ORACLE.

For each $x \in E$, we compute the machine numbers

$$\Delta_{\mathcal{A}}(x) := \delta_{Q_x}.$$

We say that a testing cube $Q \subset Q^\circ$ is $OK(\mathcal{A})$ if either $Q \in CZ(\mathcal{A}^-)$ or $\Delta_{\mathcal{A}}(x) \geq K\delta_Q$ for all $x \in E \cap 3Q$, where $K := 2^{30}/a(\mathcal{A})$ (here, the constant 10^9 in Section 1.7 of [4] is replaced by 2^{30}).

We define a Calderón–Zygmund decomposition $CZ(\mathcal{A})$ of the unit cube Q° to consist of the maximal dyadic subcubes $Q \subset Q^\circ$ that are $OK(\mathcal{A})$.

Clearly, $CZ(\mathcal{A}^-)$ refines the decomposition $CZ(\mathcal{A})$, namely, each cube in $CZ(\mathcal{A})$ is a disjoint union of the cubes in $CZ(\mathcal{A}^-)$. Since $\delta_Q \geq \frac{1}{32} \cdot \Delta_0$ for each $Q \in CZ(\mathcal{A}^-)$ (by the finite-precision version of the Main Technical Results for $\mathcal{A}^-$), we have

$$\delta_Q \geq \frac{1}{32} \cdot \Delta_0 \quad \text{for each } Q \in CZ(\mathcal{A}). \tag{2.144}$$

This implies an additional property of $CZ(\mathcal{A})$ that is required in the finite-precision version of Main Technical Results for $\mathcal{A}$.

We construct a $CZ(\mathcal{A})$-ORACLE using the GLORIFIED CZ-ORACLE, where we take $\Delta(x) := \Delta_{\mathcal{A}}(x)/K = \Delta_{\mathcal{A}}(x) \cdot a(\mathcal{A}) \cdot 2^{-30}$. Note that $a(\mathcal{A}) = a_{new} = 2^{-\widetilde{S}}$, where $\widetilde{S} \leq C\overline{S}$ for a universal constant C (see (2.97)). Note also that $\Delta_{\mathcal{A}}(x) = \delta_{Q_x}$ is an $\widetilde{S}$-bit machine number (recall that Q_x is a testing cube, and use Remark 1). Thus, $\Delta(x)$ is an $\widetilde{S}$-bit machine number for each $x \in E$, where $\widetilde{S} \leq C'\overline{S}$ for a universal constant C'. Thus, the extra hypotheses required for the finite-precision version of the GLORIFIED CZ-ORACLE are valid (see Section 2.10).

We refer the reader to Section 1.7 of [4] for a proof of the remaining properties **(CZ1)**–**(CZ5)** of the decomposition $CZ(\mathcal{A})$. See Propositions 14, 15, and 16 of [4], and equations (1.179) and (1.180) of [4].

We have thus proven all the properties of the decomposition $CZ(\mathcal{A})$ stated in the Main Technical Results for $\mathcal{A}$.

2.17. Completing the induction

In executing the algorithm PRODUCE ALL SUPPORTING DATA in finite-precision, we need to produce extra stuff, since we added stuff to the definition of *modified supporting data* (see **Modification 1** in Section 2.13.2). For each $Q \in CZ_{main}(\mathcal{A})$, we need to list all the points $x \in E \cap \frac{65}{64}Q$. However, it's easy to do that. The procedure is as follows: we loop over all points $x \in E$. For each x, we use the $CZ(\mathcal{A})$-ORACLE to find all the $Q \in CZ_{main}(\mathcal{A})$ such that $x \in \frac{65}{64}Q$, and we then add x to a list associated to each relevant Q. Any given x is associated to at most C cubes Q, and we can find each cube in the list $CZ_{main}(\mathcal{A})$ by binary search that requires work at most $C \log N$. Therefore, this procedure requires work at most $CN \log N$ in space CN. Thus, the work and storage used by the finite-precision version of the algorithm PRODUCE ALL SUPPORTING DATA are bounded as required.

In place of (1.182) and (1.183) of [4], we have to prove the estimates

$$\|(f,P)\|_{(1+a(\mathcal{A}))\widehat{Q}} \leq CM_{\widehat{Q}}(f,P) \quad \text{and} \quad M_{\widehat{Q}}(f,P) \leq C\|(f,P)\|_{\frac{65}{64}\widehat{Q}} + C\,\Delta_{new}|P|.$$

We prove these estimates using the finite-precision unconditional and conditional inequalities, just as in the infinite-precision case.

We separately treat the *simple* and *non-simple* cubes $\widehat{Q} \in CZ(\mathcal{A})$ as in Sections 1.8.1 and 1.8.2 of [4]. We make a few small changes to the analysis. which are documented below.

• In Section 1.8.1 of [4]: We defined lists $\Xi(\widehat{Q}, \mathcal{A})$ and $\Omega(\widehat{Q}, \mathcal{A})$ of linear functionals, and a linear map $T_{(\widehat{Q},\mathcal{A})}$ for each of the *non-simple cubes* $\widehat{Q} \in CZ(\mathcal{A})$. The definitions are unchanged. See the versions of the algorithms COMPUTE NEW ASSISTS, COMPUTE NEW ASSISTED FUNCTIONALS, and COMPUTE NEW EXTENSION OPERATOR in Section 2.13.2. The linear functionals and linear maps here are all computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

• In Section 1.8.1 of [4]: We need to control an extra sum when evaluating the upper bound on the work and storage. Namely, we have to control the sum

$$\sum_{\widehat{Q}\in CZ_{\text{main}}(\mathcal{A})} \Big\{ \#\Big(\frac{65}{64}\widehat{Q} \cap E\Big)\Big\}.$$

This extra term arises from the work of applying the finite-precision version of COMPUTE NEW ASSISTED FUNCTIONALS (see Section 2.13.2). This sum is bounded by CN, thanks to the bounded overlap of the cubes $\frac{65}{64}\widehat{Q}$, for $\widehat{Q} \in CZ(\mathcal{A})$. Hence, the work and storage needed to compute all the functionals defined in Section 1.8.1 of [4] are bounded as required.

• In Section 1.8.2 of [4]: We defined lists $\Xi(\widehat{Q}, \mathcal{A})$ and $\Omega(\widehat{Q}, \mathcal{A})$, and a linear map $T_{(\widehat{Q},\mathcal{A})}$ for each of the *simple cubes* $\widehat{Q} \in CZ(\mathcal{A})$. The definitions are unchanged. See the relevant text. The linear functionals and linear maps here are all computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

• In Section 1.8.2 of [4]: The finite-precision version of (1.185) (from the Main Technical Results for $\mathcal{A}^-$) states that

$$C^{-1}\|(f,R)\|_{(1+a)Q} \le M_{(Q,\mathcal{A}^-)}(f,P)| \le C\left[\|(f,P)\|_{\frac{65}{64}Q} + \Delta_{\text{junk}}|P|\right]. \tag{2.145}$$

• In Section 1.8.2 of [4]: The statement and proof of Prop. 17 are unchanged.

• In Section 1.8.2 of [4]: The finite-precision version of Lemma 15 states that

$$C^{-1}\|(f,P)\|_{(1+a_{\text{new}})\widehat{Q}} \le M_{(\widehat{Q},\mathcal{A})}(f,P) \le C\left[\|(f,P)\|_{\frac{65}{64}\widehat{Q}} + \Delta_{\text{new}}|P|\right].$$

We prove this estimate as follows. From (2.145) we have

$$\big[M_{(\widehat{Q},\mathcal{A})}(f,P)\big]^p \le C \sum_{\substack{Q\in CZ_{\text{main}}(\mathcal{A}^-)\\ Q\subset(1+t_G)\widehat{Q}}} \left[\|(f,P)\|^p_{\frac{65}{64}Q} + \Delta^p_{\text{junk}}|P|^p\right].$$

The number of Q arising in the above sum is at most C. Hence,

$$M_{(\widehat{Q},\mathcal{A})}(f,P) \le C\left[\|(f,P)\|_{\frac{65}{64}\widehat{Q}} + \Delta_{\text{new}}|P|\right],$$

as in the proof in the infinite-precision setting. Here, we use that $\Delta_{\text{junk}} \le \Delta_{\text{new}}$; see (2.82).

• In the **Closing Remarks**: We fix ϵ to be a small enough universal constant. The parameters $\Delta_g = \Delta_g(\mathcal{A}^-)$, $\Delta_\epsilon = \Delta_\epsilon(\mathcal{A}^-)$, and Δ_{new} are assumed to satisfy (2.82), (2.98) and (2.121).[4] We also impose the assumptions $\Delta_{\text{junk}}(\mathcal{A}) \geq \Delta_{\text{new}}$, $\Delta_g(\mathcal{A}) \leq \Delta_g^C$, and $\Delta_\epsilon(\mathcal{A}) \geq \Delta_g^{-C}\Delta_\epsilon$, for a large enough universal constant C. Thus, we obtain the Main Technical Results for $\mathcal{A}$ from the above bullet points.

• If $\mathcal{A} = \emptyset$ (the maximal multiindex set) then the induction is complete. We do not fix a choice of the parameters $\Delta_\epsilon(\mathcal{A})$, $\Delta_g(\mathcal{A})$, $\Delta_{\text{junk}}(\mathcal{A})$ (for $\mathcal{A} \subset \mathcal{M}$) just yet. These parameters are determined later in the proofs of our Main Theorems.

2.18. Main theorems

2.18.1. Homogeneous Sobolev spaces. In this section we prove Theorem 1 using the Main Technical Results for $\mathcal{A} = \emptyset$.

We assume we are given parameters $\Delta_{\min} = 2^{-K_{\max}\overline{S}}$, $\Delta_\epsilon^\circ := 2^{-K_1\overline{S}}$, $\Delta_g^\circ := 2^{-K_2\overline{S}}$, and $\Delta_{\text{junk}}^\circ := 2^{-K_3\overline{S}}$, for integers $K_1, K_2, K_3, K_{\max} \geq 1$ as in Theorem 1.

The proof is identical to the argument in Section 2.1 of [4], except for minor changes, which we describe below.

We start from the sentence "By translating and rescaling, we may assume $\cdots$."

We let the parameters $\Delta_\epsilon = \Delta_\epsilon(\emptyset)$, $\Delta_g = \Delta_g(\emptyset)$, and $\Delta_{\text{junk}} = \Delta_{\text{junk}}(\emptyset)$ be as in the Main Technical Results for $\mathcal{A} = \emptyset$.

According to the Main Technical Results for $\mathcal{A} = \emptyset$, we are given the following objects.

There is a dyadic decomposition CZ of the unit cube Q°. The CZ-ORACLE operates as before, except that the query point $\underline{x} \in Q^\circ$ is required to be an S-bit machine point. We can list all the cubes $Q \in CZ$ such that $\underline{x} \in \frac{65}{64}Q$, using work at most $C \log N$.

For each $Q \in CZ$ with $\frac{65}{64}Q \cap E \neq \emptyset$, we are given a collection $\Omega(Q) \subset \left[\mathbb{X}(E \cap \frac{65}{64}Q)\right]^*$ of assist functionals, a collection $\Xi(Q) \subset \left[\mathbb{X}(E \cap \frac{65}{64}Q) \oplus \mathcal{P}\right]^*$ of assisted functionals, and a linear map $T_Q : \mathbb{X}(E \cap \frac{65}{64}Q) \oplus \mathcal{P} \to \mathbb{X}$.

We recall some of the main properties of these objects in the points below.

• **Modification 1.** For each $Q \in CZ$ with $\frac{65}{64}Q \cap E \neq \emptyset$, the linear functionals $\omega \in \Omega(Q)$ are given with parameters $(\Delta_g, \Delta_\epsilon)$; also, the linear functionals $\xi \in \Xi(Q)$ are given in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of the assists $\Omega(Q)$.

Given $Q \in CZ$ with $\frac{65}{64}Q \cap E \neq \emptyset$, given an S-bit machine point $\underline{x} \in Q^\circ$, and given $\alpha \in \mathcal{M}$, we compute the linear functional $(f, P) \mapsto \partial^\alpha(T_Q(f, P))(\underline{x})$ in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in terms of the assists $\Omega(Q)$.

• **Modification 2.** We replace (2.1) of [4] with the corresponding estimate from the finite-precision version of the Main Technical Results for $\mathcal{A} = \emptyset$, namely:

$$\sum_{\xi \in \Xi(Q)} |\xi(f, P)|^p \leq C \left[\|(f, P)\|^p_{\frac{65}{64}Q} + \Delta_{\text{junk}}^p |P|^p \right]. \tag{2.146}$$

[4] Recall that we have fixed t_G and ϵ to be universal constants. Hence, the conditions (2.98) and (2.121) state that Δ_{new} is less than a small enough universal constant. These are among the conditions (2.73) and (2.74) imposed before.

• The linear maps T_Q satisfy (2.2) and (2.3) of [4] just as before.

• From the conditions in the Main Technical Results we learn that $\delta_Q > c_*$ for every $Q \in CZ$, for the universal constant $c_* = c_*(\emptyset)$. Using the CZ-ORACLE, we can list all the cubes in CZ using work at most $C \log N$. The algorithm is as before.

• **Modification 3.** As before, we let a denote the universal constant $a(\emptyset)$. According to the finite-precision version of the Main Technical Results, we know that a is an integer power of 2. Thus, a is a machine number. We define a family of cutoff functions $\widetilde{\theta}_Q$ (for $Q \in CZ$) as before. For a statement of the relevant properties of $\widetilde{\theta}_Q$, we refer the reader to the text following (2.4) of [4]. The finite-precision version of the algorithm COMPUTE AUXILIARY FUNCTIONS requires slight modification to allow for roundoff errors. Given $Q \in CZ$ and given an S-bit machine point $\underline{x} \in Q^\circ$, we compute the numbers $\partial^\alpha(\widetilde{\theta}_Q)(\underline{x})$ for all $\alpha \in \mathcal{M}$. We guarantee that the numbers $\partial^\alpha(\widetilde{\theta}_Q)(\underline{x})$ have magnitude at most Δ_g^{-C} and are computed to precision $\Delta_g^{-C}\Delta_\epsilon$ for a universal constant C. For the explanation, we define a spline function $\widetilde{\theta}$ (depending on a) with $\widetilde{\theta} \geq 1/2$ on $Q^\circ = [0,1)^n$, $\widetilde{\theta} \equiv 0$ outside $(1+a)Q^\circ$, $0 \leq \widetilde{\theta}_Q \leq 1$ on $\mathbb{R}^n$, and $\left|\partial^\beta \widetilde{\theta}_Q(x)\right| \leq C$ (for $\beta \in \mathcal{M}$, $x \in \mathbb{R}^n$). We also assume that the derivatives of $\widetilde{\theta}$ at a general S-bit machine point in $\mathbb{R}^n$ can be computed to precision Δ_ϵ. This is possible because the machine precision of our computer is $\Delta_{\min} \ll \Delta_\epsilon$. We define $\widetilde{\theta}_Q$ to be an appropriately shifted and rescaled version of $\widetilde{\theta}$ that is supported on the cube $(1+a)Q$. Since $\delta_Q \geq c^*$ for all $Q \in CZ$, we learn that $\left|\partial^\beta \widetilde{\theta}_Q(x)\right| \leq C' \leq \Delta_g^{-C'}$ for a large enough universal constant C'. We can compute $\partial^\alpha \widetilde{\theta}_Q(\underline{x})$ (for $\alpha \in \mathcal{M}$) with precision $\Delta_g^{-C}\Delta_\epsilon$ by rescaling the α-derivative of $\widetilde{\theta}$ at a suitable machine point in $\mathbb{R}^n$ (determined by $\underline{x}$).

• **Modification 4.** We modify COMPUTE POU2 to take into account roundoff errors. Given $Q \in CZ$ and given an S-bit machine point $\underline{x} \in Q^\circ$, we compute the numbers $\partial^\alpha(\theta_Q)(\underline{x})$ for each $\alpha \in \mathcal{M}$. The numbers $\partial^\alpha(\theta_Q)(\underline{x})$ are bounded in magnitude by Δ_g^{-C} and are computed to precision $\Delta_g^{-C}\Delta_\epsilon$ for a universal constant C. Here, θ_Q is defined in terms of $\widetilde{\theta}_Q$ as in the infinite-precision text. The explanation is obvious. We choose the function $\eta(t)$ to be a spline function whose derivatives can be computed to precision Δ_ϵ, and we compute the derivatives of θ_Q using the Leibniz rule. Of course, we still have properties (1)–(4) of the partition of unity (θ_Q).

• The definitions of Ξ°, Ω°, and T°, are unchanged. We define Ξ° to be the union of the lists $\Xi(Q)$, and we define Ω° to be the union of the lists $\Omega(Q)$. We define $T^\circ(f,P)$ as in (2.5) of [4], namely:

$$T^\circ(f,P) = \sum_{\substack{Q \in CZ \\ \frac{65}{64}Q \cap E \neq \emptyset}} \theta_Q \cdot T_Q(f,P) + \sum_{\substack{Q \in CZ \\ \frac{65}{64}Q \cap E = \emptyset}} \theta_Q \cdot P.$$

• **Modification 5.** The second bullet point in Proposition 18 of [4] is changed to account for roundoff errors. Given an S-bit machine point $\underline{x} \in Q^\circ$ and given

$\alpha \in \mathcal{M}$, we compute the linear functional $(f, P) \mapsto \partial^\alpha(T^\circ(f,P))(\underline{x})$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of the assists Ω°. The explanation is an obvious consequence of the Leibniz rule, since the functionals $(f, P) \mapsto \partial^\beta(T_Q(f,P))(\underline{x})$ can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, and the numbers $\partial^\beta(\theta_Q)(\underline{x})$ can be computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.

• **Modification 6.** The fourth bullet point of Proposition 18 of [4] is changed to instead consist of the estimate

$$\sum_{\xi\in\Xi^\circ} |\xi(f,P)|^p \le C \cdot \left[\|(f,P)\|^p_{\frac{65}{64}Q^\circ} + \Delta_{\text{junk}}^p |P|^p \right]. \tag{2.147}$$

Next, we explain how to modify the proof of Proposition 18.

• **Modification 7.** We replace (2.10) of [4] with

$$\sum_{\substack{Q\in CZ \\ \frac{65}{64}Q\cap E\neq\emptyset}} \sum_{\xi\in\Xi(Q)} |\xi(f,P)|^p \le C \cdot \sum_{\substack{Q\in CZ \\ \frac{65}{64}Q\cap E\neq\emptyset}} \left[\|(f,P)\|^p_{\frac{65}{64}Q} + \Delta_{\text{junk}}^p |P|^p \right],$$

which follows from (2.146).

Now, the cardinality of CZ is at most a universal constant and $\|(f,P)\|_{\frac{65}{64}Q} \le C\|(f,P)\|_{\frac{65}{64}Q^\circ}$, just as before. Hence, we have

$$\sum_{\substack{Q\in CZ \\ \frac{65}{64}Q\cap E\neq\emptyset}} \sum_{\xi\in\Xi(Q)} |\xi(f,P)|^p \le C \cdot \left[\|(f,P)\|^p_{\frac{65}{64}Q^\circ} + \Delta_{\text{junk}}^p |P|^p \right].$$

But this is just the estimate in the modified fourth bullet point (see **Modification 6**). The remainder of the proof of Proposition 18 of [4] is unchanged. This completes the proof of the modified version of Proposition 18 of [4].

• **Modification 8.** Now we introduce a linear map $\mathfrak{R} : \mathbb{X}(E) \mapsto \mathcal{P}$ using the finite-precision version of OPTIMIZE VIA MATRIX with $\Delta = \Delta_{\text{junk}}$. We compute the map $\mathfrak{R}$ in short form with parameters $(\Delta_g, \Delta_\epsilon)$ in the following sense: for each $\alpha \in \mathcal{M}$, we compute the linear functional $f \mapsto \partial^\alpha(\mathfrak{R}(f))(0)$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ (without assists). We guarantee that

$$\sum_{\xi\in\Xi^\circ} |\xi(f,\mathfrak{R}(f))|^p \le C \cdot \Big[\sum_{\xi\in\Xi^\circ} |\xi(f,R)|^p + \Delta_{\text{junk}}^p |R|^p \Big] \quad \text{for any } R \in \mathcal{P}. \tag{2.148}$$

(This estimate is the finite-precision analogue of (2.14) of [4].)

We can answer slightly more general queries: given an S-bit machine point $\underline{x} \in Q^\circ$, and given $\alpha \in \mathcal{M}$, we compute the linear functional $f \mapsto \partial^\alpha(\mathfrak{R}(f))(\underline{x})$. This follows because of Taylor's formula, which allows us to express the functional $f \mapsto \partial^\alpha(\mathfrak{R}(f))(\underline{x})$ as a weighted combination

$$\sum_{|\beta|\le m-1-|\alpha|} \frac{1}{\beta!} \cdot (\underline{x})^\beta \, \partial^{\alpha+\beta}(\mathfrak{R}(f))(0)$$

of the linear functionals $f \mapsto \partial^\gamma(\mathfrak{R}(f))(0)$ $(\gamma \in \mathcal{M})$. The coefficients in this combination can be computed to precision $(\Delta_g, \Delta_\epsilon)$, and so the claim follows.

• **Modification 9.** The list Ξ consists of all the functionals $\xi : f \mapsto \xi^\circ(f, \mathfrak{R}(f))$ with $\xi^\circ \in \Xi^\circ$. We compute each $\xi \in \Xi$ with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ by composing a linear functional $\xi^\circ \in \Xi^\circ$ with the linear map $f \mapsto \mathfrak{R}(f)$.

• **Modification 10.** The cutoff function θ° is defined as before. The same properties (1)–(4) hold. For the construction, we choose θ° to be an appropriate spline function. The computation of θ° is modified to take into account roundoff errors. Given an S-bit machine point $\underline{x} \in Q^\circ$, we compute the numbers $\partial^\alpha(\theta^\circ)(\underline{x})$ (all $\alpha \in \mathcal{M}$) to within precision $\Delta_g^{-C}\Delta_\epsilon$; these numbers have magnitude at most Δ_g^{-C}. This computation requires work at most C.

• **Modification 11.** Just as before, we define $T\colon \mathbb{X}(E) \to \mathbb{X}$ by the formula $Tf = \theta^\circ \cdot T^\circ(f, \mathfrak{R}(f)) + (1 - \theta^\circ) \cdot \mathfrak{R}(f)$. We need to modify the query algorithm for T to take into account roundoff error. Given an S-bit machine point $\underline{x} \in Q^\circ$, and given $\alpha \in \mathcal{M}$, we compute the linear functional $f \mapsto \partial^\alpha(T(f))(\underline{x})$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of the assists Ω. The explanation is an obvious consequence of the Leibniz rule, since the linear maps $\mathfrak{R}$, T°, and the cutoff function θ° have been computed with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, as described in the previous bullet points.

• For the same reason as before, we have

$$\|Tf\|_{\mathbb{X}}^p \le C \cdot \sum_{\xi \in \Xi} |\xi(f)|^p .$$

(As in the proof of (2.17) of [4]).

• **Modification 12.** The estimate (2.18) of [4] no longer holds. Instead, we have

$$\begin{aligned}\sum_{\xi \in \Xi} |\xi(f)|^p = \sum_{\xi^\circ \in \Xi^\circ} |\xi^\circ(f, \mathfrak{R}(f))|^p &\le C \inf_{R \in \mathcal{P}} \Big\{ \sum_{\xi^\circ \in \Xi^\circ} |\xi^\circ(f, R)|^p + \Delta_{\text{junk}}^p |R|^p \Big\} \\ &\le C \inf_{R \in \mathcal{P}} \Big\{ \|(f, R)\|_{\frac{65}{64}Q^\circ}^p + \Delta_{\text{junk}}^p |R|^p \Big\} .\end{aligned}$$

(As in the proof of (2.20) of [4].)

For an arbitrary $F \in \mathbb{X}$ with $F = f$ on E, set $R = J_x F$, and estimate $\|F - R\|_{L^p(\frac{65}{64}Q^\circ)} \le C\|F\|_{\mathbb{X}}$ using the Sobolev inequality. Also, by the Sobolev inequality, $|J_x F| \le \|F\|_{\mathbb{X}} + \|F\|_{L^p(Q^\circ)}$. So the last infimum above is dominated by $C \cdot \big[\|F\|_{\mathbb{X}}^p + \Delta_{\text{junk}}^p \|F\|_{L^p(Q^\circ)}^p\big]$. Hence,

$$\sum_{\xi \in \Xi} |\xi(f)|^p \le C \cdot \inf \Big\{ \|F\|_{\mathbb{X}}^p + \Delta_{\text{junk}}^p \|F\|_{L^p(Q^\circ)}^p : F \in \mathbb{X},\ F = f \text{ on } E \Big\} .$$

• Just as before, we prove that

$$c \cdot \|f\|_{\mathbb{X}(E)}^p \le \sum_{\xi \in \Xi} |\xi(f)|^p .$$

(As in the proof of (2.20) of [4].)

Recall that we have set $\Delta_g = \Delta_g(\emptyset)$, $\Delta_\epsilon = \Delta_\epsilon(\emptyset)$, and $\Delta_{\text{junk}} = \Delta_{\text{junk}}(\emptyset)$ in the above bullet points.

From (2.73), we may impose the assumption $\Delta_{\text{junk}} \leq \Delta_{\text{junk}}^{\circ}$. Thus, from the last three bullet points we learn that

$$c\|f\|_{\mathbb{X}(E)} \leq \Big(\sum_{\xi \in \Xi} |\xi(f)|^p \Big)^{1/p} \leq C \inf \left\{ \|F\|_{\mathbb{X}} + \Delta_{\text{junk}}^{\circ} \|F\|_{L^p(Q^\circ)} : F \in \mathbb{X},\ F = f \text{ on } E \right\}$$

and

$$\|Tf\|_{\mathbb{X}} \leq C \cdot \inf \left\{ \|F\|_{\mathbb{X}} + \Delta_{\text{junk}}^{\circ} \cdot \|F\|_{L^p(Q^\circ)} : F \in \mathbb{X},\ F = f \text{ on } E \right\},$$

as desired (see Theorem 1).

All of the functionals $f \mapsto \omega(f)$, $f \mapsto \xi(f)$, and $f \mapsto \partial^\alpha(Tf)(\underline{x})$ in the above bullet points, which arise in the statement of Theorem 1, are specified with parameters $(\Delta_g^{C_0}, \Delta_g^{-C_0}\Delta_\epsilon)$ for a universal constant C_0. According to (2.73) and (2.74), we may assume that $\Delta_g^{\circ} \leq (\Delta_g)^{C_0}$ and $\Delta_g^{-C_0}\Delta_\epsilon \leq \Delta_\epsilon^{1/2} \leq \Delta_\epsilon^{\circ}$. Thus, we can compute all of the functionals relevant to Theorem 1 with parameters $(\Delta_g^{\circ}, \Delta_\epsilon^{\circ})$.

This completes the proof of Theorem 1.

2.18.2. Inhomogeneous Sobolev spaces. Once we pass from Homogeneous $L^{m,p}(\mathbb{R}^n)$ to Inhomogeneous $W^{m,p}(\mathbb{R}^n)$, the error terms $\Delta_{\text{junk}}^{\circ}\|F\|_{L^p(Q^\circ)}$ in Theorem 1 will become irrelevant.

Our main result for inhomogeneous Sobolev spaces is Theorem 2 written below. We follow the argument in Section 2.2 of [4], with the following changes.

• **Modification 1.** We let $T^\circ, \Xi^\circ, \Omega^\circ$ be defined as in the previous section. We will use the finite-precision version of Proposition 18 of [4], which guarantees the following:

- We list the functionals in Ω°. Each $\omega^\circ \in \Omega^\circ$ is specified in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$.
- We list the functionals in Ξ°. Each $\xi^\circ \in \Xi^\circ$ is specified in short form in terms of the assists Ω° with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$. The functionals in Ξ° satisfy the modified estimate (2.147).
- Given an S-bit machine point $\underline{x} \in Q^\circ$ and given $\alpha \in \mathcal{M}$, we compute the linear functional $(f, P) \mapsto \partial^\alpha(T^\circ(f, P))(\underline{x})$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of the assists Ω°, using work at most $C \log N$.

• **Modification 2.** We introduce a cutoff function θ°. Let $\underline{x} \in Q^\circ$ be a given point with S-bit machine numbers as coordinates. We compute the numbers $\partial^\alpha(\theta^\circ)(\underline{x})$ (all $\alpha \in \mathcal{M}$) up to an additive error of absolute value at most $\Delta_g^{-C}\Delta_\epsilon$; these numbers have absolute value at most Δ_g^{-C}. This requires work at most C.

• **Modification 3.** In the proof of (2.23) of [4], we use the modified estimate from the fourth bullet point in the finite-precision version of Proposition 18. (See (2.147).) This gives

$$\sum_{\xi \in \Xi^\circ} |\xi(f, 0)|^p \leq C \inf \left\{ \|F\|^p_{L^{m,p}(\frac{65}{64}Q^\circ)} + \|F\|^p_{L^p(\frac{65}{64}Q^\circ)} :\ F \in \mathbb{X},\ \ F = f \text{ on } E \right\}.$$

The junk term in (2.147) disappears because we set $P = 0$.

• **Modification 4.** The rest of the content of the section is unchanged. In particular, the collections Ξ and Ω consisting of linear functionals on $\mathbb{X}(E)$, and the linear map $T\colon \mathbb{X}(E) \mapsto \mathbb{X}$ are defined as before. The functionals in Ω are computed in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$, and the functionals in Ξ are computed in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of the assists Ω. Given an S-bit machine point $\underline{x} \in Q^\circ$ and given $\alpha \in \mathcal{M}$, we can compute the linear functional $f \mapsto \partial^\alpha(Tf)(\underline{x})$ in short form with parameters $(\Delta_g^C, \Delta_g^{-C}\Delta_\epsilon)$ in terms of the assists Ω, using work at most $C \log N$.

All the functionals in the above bullet points are computed with parameters $(\Delta_g^{C_0}, \Delta_g^{-C_0}\Delta_\epsilon)$ for a universal constant C_0. According to (2.73) and (2.74), we may assume that $\Delta_g^\circ \le (\Delta_g)^{C_0}$ and $\Delta_g^{-C_0}\Delta_\epsilon \le \Delta_\epsilon^{1/2} \le \Delta_\epsilon^\circ$, for parameters Δ_g° and Δ_ϵ° as in the statement of Theorem 1. Thus, we can compute our functionals with parameters $(\Delta_g^\circ, \Delta_\epsilon^\circ)$ for suitable Δ_g° and Δ_ϵ° (see below).

We have proven the following theorem, which is our main extension theorem for inhomogeneous Sobolev spaces in a finite-precision model of computation.

Theorem 2. *There exists $C = C(m, n, p) \ge 1$ such that the following holds.*

Let $\overline{S} \ge 1$ be an integer.

Assume $E \subset \frac{1}{32}Q^\circ$ satisfies $\#(E) = N \ge 2$, where $Q^\circ = [0,1)^n$. Assume that the points of E have $\overline{S}$-bit machine numbers as coordinates.

Assume that $K_1, K_2, K_{\max} \in \mathbb{N}$ satisfy $K_{\max} \ge C \cdot K_1 \ge C^2 \cdot K_2 \ge C^3$.

Let $\Delta_{\min}^\circ = 2^{-K_{\max}\overline{S}}$, $\Delta_\epsilon^\circ := 2^{-K_1\overline{S}}$, and $\Delta_g^\circ = 2^{-K_2\overline{S}}$.

We assume that our computer can perform arithmetic operations on S-bit machine numbers with precision $\Delta_{\min}^\circ$, where $S = K_{\max} \cdot \overline{S}$.

Then we compute lists Ω and Ξ, consisting of linear functionals on $W^{m,p}(E) = \{f : E \to \mathbb{R}\}$, with the following properties.

- *The sum of $\mathrm{depth}(\omega)$ over all $\omega \in \Omega$ is bounded by CN. The number of functionals in Ξ is at most CN.*
- *Each functional ξ in Ξ has Ω-assisted bounded depth.*
- *The functionals $\omega \in \Omega$ and $\xi \in \Xi$ are computed in short form with parameters $(\Delta_g^\circ, \Delta_\epsilon^\circ)$.*
- *For all $f \in W^{m,p}(E)$ we have*

$$C^{-1}\,\|f\|_{W^{m,p}(E)} \le \Big[\sum_{\xi\in\Xi} |\xi(f)|^p\Big]^{1/p} \le C\,\|f\|_{W^{m,p}(E)}.$$

Moreover, there exists a linear map $T\colon W^{m,p}(E) \to W^{m,p}(\mathbb{R}^n)$ with the following properties.

- *T has Ω-assisted bounded depth.*
- *$Tf = f$ on E and*

$$\|Tf\|_{W^{m,p}(\mathbb{R}^n)} \le C \cdot \|f\|_{W^{m,p}(E)}$$

for all $f \in \mathbb{X}(E)$.

- *We produce a query algorithm that operates as follows.*

  Given an S*-bit machine point* $\underline{x} \in Q^\circ$ *and given* $\alpha \in \mathcal{M}$*, we compute a short form description of the* Ω*-assisted bounded depth linear functional* $W^{m,p}(E) \ni f \mapsto \partial^\alpha(Tf)(\underline{x})$*. We compute this functional in short form with parameters* $(\Delta_g^\circ, \Delta_\epsilon^\circ)$*. This requires work at most* $C \log N$*.*

The computations above require one-time work at most $CN \log N$ *in space* CN*.*

References

[1] de Berg, M., Cheong, O., van Kreveld, M. and Overmars, M.: *Computational geometry: algorithms and applications*. Springer-Verlag, 2008.

[2] Fefferman, C. and Klartag, B.: Fitting a C^m-smooth function to data II. *Rev. Mat. Iberoam.* **25** (2009), no. 1, 49–273.

[3] Fefferman, C., Israel, A. and Luli, G. K.: Fitting a Sobolev function to data I. *Rev. Mat. Iberoam.* **32** (2016), no. 1, 275–376.

[4] Fefferman, C., Israel, A. and Luli, G.K.: Fitting a Sobolev function to data II. *Rev. Mat. Iberoam.* **32** (2016), no. 2, 649–750.

[5] von Neumann, J.: First draft of a report on the EDVAC. Contract No. W-670-ORD-492, Moore School of Electrical Engineering, Univ. of Penn., Philadelphia, 1945. Reprinted in *IEEE Ann. Hist. Comput.* **15** (1993), no. 4, 27–75.

Received November 19, 2014.

Charles Fefferman: Department of Mathematics, Princeton University, Fine Hall, Washington Road, Princeton, NJ 08544, USA.
E-mail: cf@math.princeton.edu

Arie Israel: Department of Mathematics, University of Texas at Austin, RLM Hall, 2515 Speedway, Austin, TX 78712, USA.
E-mail: arie@math.utexas.edu

Garving K. Luli: Department of Mathematics, University of California at Davis, One Shields Avenue, Davis, CA 95616, USA.
E-mail: kluli@math.ucdavis.edu

Charles Fefferman's research is supported by NSF grant DMS-1265524 and AFOSR grant FA9550-12-1-0425. Arie Israel's research is supported by an NSF postdoctoral fellowship, DMS-1103978. Garving Luli's research is supported by NSF grant DMS-1355968.