Rev. Mat. Iberoamericana **25** (2009), no. 1, 49–273

Fitting a C^m-Smooth Function to Data II

Charles Fefferman and Bo'az Klartag

Abstract

We exhibit efficient algorithms to perform the following task: Given a function f defined on a finite subset $E \subset \mathbb{R}^n$, compute a C^m function F on $\mathbb{R}^n$, with a controlled C^m norm, that approximates f on the subset E.

Contents

Chapter I - Blobs and ALPs

Chapter II - The Basic Families of Convex Sets

2000 Mathematics Subject Classification: 65D05, 65D17.
Keywords: Algorithm, interpolation, approximation, C^m-smoothness.

Chapter III - Lengthscales and Calderón-Zygmund decompositions

Chapter IV - Main Algorithm

Chapter V - Proofs

Appendix - Computation in Finite Precision

1. Introduction

This manuscript is the second part in a series of papers tackling the problem of interpolation of finite data, in any dimension and for any degree of smoothness. Here, and in [20], we give detailed proofs of the results announced in [19]. Let us briefly remind the reader of the problems and results presented in [19].

We fix positive integers m and n. Suppose we are given a finite set $E \subset \mathbb{R}^n$, and a function $f : E \to \mathbb{R}$. We are interested in constructing a function $F : \mathbb{R}^n \to \mathbb{R}$, that extends the given function f, and whose $C^m(\mathbb{R}^n)$ norm is of the smallest possible order of magnitude.

As in [19], here the "construction of a function" is interpreted from the viewpoint of theoretical computer science. That is, we give algorithms for computing smooth extensions of functions, and we try to minimize the time and storage required by an (idealized) computer when executing these algorithms.

Let us define the problem more precisely. Suppose that $E \subset \mathbb{R}^n$ is a finite set, and let $f : E \to \mathbb{R}$, $\sigma : E \to [0, \infty)$ be functions. We denote by $\| f \|_{C^m(E,\sigma)}$ the infimum over all $M > 0$, for which there exists a function $F \in C^m(\mathbb{R}^n)$ such that

$$(1) \quad \| F \|_{C^m(\mathbb{R}^n)} \leq M \text{ and } |F(x) - f(x)| \leq M\sigma(x) \text{ for each } x \in E.$$

We pay particular attention to the case $\sigma \equiv 0$, and hence we set

$$\| f \|_{C^m(E)} := \| f \|_{C^m(E,0)} .$$

Two numbers $X, Y \geq 0$ determined by E, f, σ, m, n are said to have "the same order of magnitude" if $cX \leq Y \leq CX$, with constants c and C depending only on m and n. To "compute the order of magnitude of X" is to compute some Y such that X and Y have the same order of magnitude. The main result proved in [19] can be summarized as follows:

Theorem 1. *The algorithm that was presented in* [19] *receives as input a set $E \subset \mathbb{R}^n$ of cardinality N, and two functions $f : E \to \mathbb{R}$, $\sigma : E \to [0, \infty)$. The algorithm computes the order of magnitude of $\| f \|_{C^m(E,\sigma)}$, using work that is at most $CN \log N$ and storage that is at most CN, where C is a constant depending only on m and n.*

The algorithm mentioned in Theorem 1 runs on an (idealized) von Neumann computer [34] that is able to add, subtract, multiply and divide exact real numbers, and also to detect their sign. We assume in addition that a real number can be stored at a single memory address.

In this follow-up paper, we deal with the problem of actually computing a near-optimal function F that satisfies (1) with M having the order of magnitude of $\| f \|_{C^m(E,\sigma)}$.

As was described in [19], to "compute a function F" means the following: First, we enter the data E, f, σ into a computer. The computer runs for a while, performing L_0 machine operations. It then signals that it is ready to accept further input. Whenever we enter a point $x \in \mathbb{R}^n$, the computer responds by producing an $m^{\underline{th}}$ degree polynomial P_x on $\mathbb{R}^n$, using L_1 machine operations to perform the computation. We say that our algorithm "computes the function F" if, for each $x \in \mathbb{R}^n$, the polynomial P_x produced by that algorithm is precisely the $m^{\underline{th}}$ order Taylor polynomial of F at x.

We call L_0 the "one-time work" and L_1 the "work to answer a query". Our main result here is the following theorem, announced in [19].

Theorem 2. *The algorithm we present below computes a function $F \in C^m(\mathbb{R}^n)$ that satisfies* (1)*, with M having the same order of magnitude as $\| f \|_{C^m(E,\sigma)}$. The one-time work of our algorithm is at most $CN \log N$, the storage is at most CN, and the work to answer a query is at most $C \log N$. Here, C depends only on m and n.*

In addition to the algorithm of Theorem 2, we provide algorithms for related and generalized problems. First, we claim that the function F we compute depends linearly on f. Furthermore, for any $x \in \mathbb{R}^n$ the polynomial P_x actually depends (linearly) only on at most C parameters among $(f(x))_{x \in E}$, where $C > 0$ is a constant depending only on m and n. We may modify our algorithm, to respond to a query by returning the coefficients of this short linear dependence (see the exact formulation in Section 35).

Second, rather than specifying the function value at any point $x \in E$, we might want to provide input of different types; for example, we might also want to specify the gradient of the function at some points. Permissible types of input are discussed in Section 36. Third, we would also like to gather some information regarding all possible smooth extensions with a bounded $C^m(\mathbb{R}^n)$ norm. Simple variants of the algorithm from Theorem 2 yield solutions to these problems (and others). These will be described in Section 34 and Section 35.

We would like to remark here that our model of computation for Theorem 2 is slightly different from the one used in [19]; in addition to comparisons and arithmetic operations on real numbers, we also require the operations of logarithm, exponent, and of rounding a real number to the closest integer. In the Appendix we analyze the performance of the algorithm of Theorem 2, on an (idealized) digital computer, that is unable to work with exact real numbers. That is, we assume that a real number is represented in a digital computer, to a certain accuracy, using S bits. We prove that the output of the algorithm from Theorem 2, is exact to a degree of accuracy of S bits. A precise statement and analysis are given in the Appendix.

Theorem 2 is essentially proved by rewriting the proof from [16] in an algorithmic way. Let us explain here the basic notions that are relevant to the proof, and at the same time review the structure of this manuscript. We denote by $\mathcal{P}^+$ the space of all polynomials of degree at most m on $\mathbb{R}^n$, and let $\mathcal{P} \subset \mathcal{P}^+$ be the space of all polynomials of degree at most $m-1$. For $x \in \mathbb{R}^n$, $\delta > 0$ we define

$$(2) \qquad B^+(x,\delta) = \{P \in \mathcal{P}^+ : |\partial^\beta P(x)| \le \delta^{m-|\beta|} \text{ for } |\beta| \le m\}.$$

We also set

$$B(x,\delta) = \mathcal{P} \cap B^+(x,\delta) = \{P \in \mathcal{P} : |\partial^\beta P(x)| \le \delta^{m-|\beta|} \text{ for } |\beta| \le m-1\}.$$

Given $M > 0$ and a subset σ in a linear space V we abbreviate

$$M \cdot \sigma = \{Mv : v \in \sigma\}.$$

For a function $F \in C^m(\mathbb{R}^n)$ and $x \in \mathbb{R}^n$ we denote by $J_x^+(F)$ the m-jet of F at x, that is the $m^{\underline{\text{th}}}$ order Taylor polynomial of F at x. We write $J_x(F)$ for the $(m-1)$-jet of F at x.

By Taylor's theorem, if $F \in C^m(\mathbb{R}^n)$ with $\| F \|_{C^m(\mathbb{R}^n)} \le M$, then for any $x, y \in \mathbb{R}^n$,

$$(3) \qquad J_x(F) - J_y(F) \in CM \cdot B(x, |x-y|),$$

where $C > 0$ is a constant depending only on m and n.

Conversely, suppose $S \subset \mathbb{R}^n$ is a finite set, and for each $x \in S$ suppose we are given a polynomial $P_x \in M \cdot B(x, 1)$ such that

(4) $\quad P_x - P_y \in M \cdot B(x, |x - y|)$ for all $x, y \in S$.

According to the classical Whitney Theorem (see [35] or [32, Section VI]), under condition (4), there exists a C^m function $F : \mathbb{R}^n \to \mathbb{R}$ with $\| F \|_{C^m(\mathbb{R}^n)} < CM$ such that

(5) $\quad J_x(F) = P_x$ for all $x \in S$.

Thus, the "balls" $MB(x, \delta)$ capture the exact essence of having a bounded C^m norm. This family of balls will play a central rôle in this paper. We say that $\mathcal{B}(x, \delta) = (MB(x, \delta))_{M>0}$ is a "blob" in $\mathcal{P}$, that is, an increasing family of convex sets in $\mathcal{P}$.

We will need to represent various "blobs" in the computer, to a certain degree of approximation. To that end, we present in Chapter I a detailed discussion of the relevant algorithmic issues. Readers who are familiar with computer programming, might choose at first reading, to begin at Section 2 and then, perhaps, skip to Section 6 or Section 7.

We will make use of some standard algorithms from theoretical computer science. In Section 9 we survey the Callahan-Kosaraju decomposition that was previously used in [19]. Using the Callahan-Kosaraju decomposition, we compute and discuss in Sections 10,...,13 the basic blobs that will accompany us throughout the paper. We think of these blobs, that are denoted by

$$\Gamma(x_0, \ell) = (\Gamma(x_0, \ell, M))_{M>0}, \qquad \text{for } x_0 \in E \text{ and } \ell > 0,$$

as representing candidate Taylor polynomials of our desired C^m function at x_0. To help understand the meaning of these basic blobs, it might be useful to consider the following set: Fix $x_0 \in E, \ell \geq 0$, $M > 0$. Consider all polynomials $P_0 \in \mathcal{P}$ for which

(6) $\forall x_1, \dots, x_\ell \in E$, $\exists P_1, \dots, P_\ell \in \mathcal{P}$ such that for any $i, j = 0, \dots, \ell$,

$$|P_i(x_i) - f(x_i)| \leq M\sigma(x_i),\ P_i \in M \cdot B(x_i, 1) \text{ and } P_i - P_j \in M \cdot B(x_i, |x_i - x_j|).$$

We would like to emphasize that (6) is similar *only in spirit* to the actual set $\Gamma(x_0, \ell, M)$, and that the actual definition of $\Gamma(x_0, \ell, M)$ will be different. However, both (6) and $\Gamma(x_0, \ell, M)$ share two important characteristics. First, when $M > C \| f \|_{C^m(E,\sigma)}$, the Taylor polynomials of all admissible extensions belong to our set, as follows from (3). Second, and somewhat more exciting, is that these blobs stabilize very quickly; for some constant ℓ_* depending only on m and n, we have that $(\Gamma(x_0, \ell_*, M))_{M>0}$ is a good approximation of $(\Gamma(x_0, \ell, M))_{M>0}$ for all $\ell > \ell_*$ (and any $x_0 \in E$).

Having constructed the basic blobs, we continue along the lines of [16]. In Section 18, we attach, to each $x_0 \in E$ a family of small, positive numbers we call "lengthscales". Very roughly, these numbers represent sizes of neighborhoods of x_0 in which we know how to solve various partial extension problems. These length scales are used in Sections 19,...,24 to create certain nested Calderón-Zygmund decompositions on $\mathbb{R}^n$. Our desired extending function will be constructed recursively, from a very fine scale to a mesoscopic one. The nested Calderón-Zygmund decompositions, and the corresponding partitions of unity, are used to "glue" together different patches of the extending function.

We move to Section 23, where our second algorithmic ingredient is exposed: the so-called "BBD tree", due to Arya, Mount, Netanyahu, Silverman and Wu [1]. The results of that paper are summarized in Section 23, and later applied in Sections 25, 26, 27 to aid several computations related to the above Calderón-Zygmund decompositions.

Our *Main Algorithm* recursively constructs extension functions defined on certain cubes in $\mathbb{R}^n$. The *Main Algorithm* is presented in Section 29, along with the *Main Lemma*. The *Main Lemma* states, more or less, that the *Main Algorithm* works. The proof of the *Main Lemma* is inductive, and is dealt with in Sections 30,...,33. The proof is similar to the proof in [16]. The Appendix contains a discussion of various issues related to the implementation of our algorithm in an (idealized) digital computer in which real numbers are represented only with finite precision. The mathematical issues involved here are minor; readers unconcerned with the rigorous treatment of roundoff errors may wish to omit the Appendix.

This paper is part of a literature on the problem of extending a given function $f : E \to \mathbb{R}$, defined on an arbitrary subset $E \subset \mathbb{R}^n$, to a function $F \in C^m(\mathbb{R}^n)$. The question goes back to Whitney [35, 36, 37], with contributions by Glaeser [21], Brudnyi-Shvartsman [5,...,10 and 29, 30, 31], Zobin [38, 39], Bierstone-Milman-Pawłucki [2, 3], Fefferman [12,...,18] and A. and Y. Brudnyi [4].

Acknowledgments. We are grateful to A. Tsao for making us aware of the practical problem of fitting a smooth surface to N given points. We would like to thank A. Razborov for helpful discussions pertaining to the model of computation, and A. Naor for pointing us to the relevant computer science literature. We also benefitted from a short letter by J. Ericksson, posted in a computational geometry internet forum, regarding models of computation. We are grateful to Gerree Pecht for accurately and artistically LaTeXing our long manuscript.

Chapter I - Blobs and ALPs

2. Blobs and ALPs: Definitions

In the next several sections we introduce the data structures that are used to describe families of convex sets, and we explain some basic algorithms to manipulate those data structures.

Let V be a finite-dimensional vector space. A "blob" in V is a family $\mathcal{K} = (K_M)_{M>0}$ of (possibly empty) convex subsets $K_M \subseteq V$, parametrized by $M \in (0, \infty)$, such that $M < M'$ implies $K_M \subseteq K_{M'}$. The "onset" of a blob $\mathcal{K} = (K_M)_{M>0}$ is defined as the infimum of all the $M > 0$ for which $K_M \neq \emptyset$. (If all K_M are empty, then onset $\mathcal{K} = +\infty$.)

Let $\mathcal{K} = (K_M)_{M>0}$ be a blob in V, let $v \in V$ be a vector, and let $C \geq 1$ be a constant. Then we call v a "C-original vector" for $\mathcal{K}$ if we have $v \in K_M$ for all $M > C\cdot$ onset $\mathcal{K}$. (By definition, we cannot have $v \in K_M$ for any $M <$ onset $\mathcal{K}$.)

Suppose $\mathcal{K} = (K_M)_{M>0}$ and $\mathcal{K}' = (K'_M)_{M>0}$ are blobs in V, and let $C \geq 1$ be a number. We say that $\mathcal{K}$ and $\mathcal{K}'$ are "C-equivalent" if they satisfy $K_M \subseteq K'_{CM}$ and $K'_M \subseteq K_{CM}$ for all $M \in (0, \infty)$.

Note that, if $\mathcal{K}$ and $\mathcal{K}'$ are C_1-equivalent, and if $\mathcal{K}'$ and $\mathcal{K}''$ are C_2-equivalent, then $\mathcal{K}$ and $\mathcal{K}''$ are $(C_1 \cdot C_2)$-equivalent. Note also that, if $\mathcal{K}$ and $\mathcal{K}'$ are C-equivalent, then $(1/C) \cdot$ onset $\mathcal{K} \leq$ onset $\mathcal{K}' \leq C \cdot$ onset $\mathcal{K}$.

In addition, suppose $\mathcal{K}$ and $\mathcal{K}'$ are C_1-equivalent, and suppose v is a C_2-original vector for $\mathcal{K}$. Then v is a C_3-original vector for $\mathcal{K}'$, where C_3 is determined by C_1 and C_2. (We can take $C_3 = C_1^2 \cdot C_2$, as the reader may easily verify.)

We describe a few elementary operations on blobs. First, suppose $V = V_1 \oplus V_2$ is a direct sum of vector spaces, and let $\mathcal{K}^i = (K^i_M)_{M>0}$ be a blob in V_i for $i = 1, 2$. Then we write $\mathcal{K}^1 \times \mathcal{K}^2$ for the blob $(K^1_M \times K^2_M)_{M>0}$ in V. If $K^1_M = V_1$ for all $M \in (0, \infty)$, then we write $V_1 \times \mathcal{K}^2$ for $\mathcal{K}^1 \times \mathcal{K}^2$; and similarly for $\mathcal{K}^1 \times V_2$.

Next, suppose $T : V_1 \longrightarrow V_2$ is a linear map of finite-dimensional vector spaces, and let $\mathcal{K} = (K_M)_{M>0}$ be a blob in V_1. Then we write $T\mathcal{K}$ to denote the blob $(TK_M)_{M>0}$ in V_2. Note that $T'(T\mathcal{K}) = (T'T)\mathcal{K}$ if $T : V_1 \longrightarrow V_2$ and $T' : V_2 \longrightarrow V_3$.

Now suppose that $\mathcal{K}^i = (K^i_M)_{M>0}$ are blobs in a vector space V, for $i = 1, 2, \dots, T$. Then we define their intersection $\mathcal{K}^1 \cap \cdots \cap \mathcal{K}^T$ to be the blob $(K^1_M \cap \cdots \cap K^T_M)_{M>0}$.

Finally, suppose $\mathcal{K}^1 = (K^1_M)_{M>0}$ and $\mathcal{K}^2 = (K^2_M)_{M>0}$ are blobs in V. Then we define their Minkowski sum $\mathcal{K}^1 + \mathcal{K}^2$ to be the blob $(K^1_M + K^2_M)_{M>0}$ in V, where $K^1_M + K^2_M = \{v^1 + v^2 : v^1 \in K^1_M, v^2 \in K^2_M\}$.

This concludes our list of elementary operations on blobs. Note that the above operations behave well with respect to C-equivalence. More precisely:

- If $\mathcal{K}^1$ and $\tilde{\mathcal{K}}^1$ are C-equivalent blobs in V_1, and if $\mathcal{K}^2$ and $\tilde{\mathcal{K}}^2$ are C-equivalent blobs in V_2, then $\mathcal{K}^1 \times \mathcal{K}^2$ and $\tilde{\mathcal{K}}^1 \times \tilde{\mathcal{K}}^2$ are C-equivalent blobs in $V_1 \oplus V_2$;

- If $\mathcal{K}$ and $\tilde{\mathcal{K}}$ are C-equivalent blobs in V_1, and if $T : V_1 \rightarrow V_2$ is linear, then $T\mathcal{K}$ and $T\tilde{\mathcal{K}}$ are C-equivalent blobs in V_2.

- If $\mathcal{K}^i$ and $\tilde{\mathcal{K}}^i$ are C-equivalent blobs in V for each $i = 1, \dots, T$, then $\mathcal{K}^1 \cap \cdots \cap \mathcal{K}^T$ and $\tilde{\mathcal{K}}^1 \cap \cdots \cap \tilde{\mathcal{K}}^T$ are again C-equivalent.

- Finally, if $\mathcal{K}^i$ and $\tilde{\mathcal{K}}^i$ are C-equivalent blobs in V, for $i = 1, 2$, then $\mathcal{K}^1 + \mathcal{K}^2$ and $\tilde{\mathcal{K}}^1 + \tilde{\mathcal{K}}^2$ are again C-equivalent.

Among all blobs in a finite-dimensional vector space V, we focus attention on those given by "Approximate Linear Algebra Problems", or "ALPs". To define these, let $\lambda_1, \dots, \lambda_L$ be (real) linear functionals on V, let $b_1, \dots, b_L$ be real numbers, let $\sigma_1, \dots, \sigma_L$ be non-negative real numbers, and let $M_* \in [0, +\infty]$. We call

(1) $\mathcal{A} = [(\lambda_1, \dots, \lambda_L), (b_1, \dots b_L), (\sigma_1, \dots, \sigma_L), M_*]$ an "ALP" in V.

With $\mathcal{A}$ given by (1), we define a blob

(2) $\mathcal{K}(\mathcal{A}) = (K_M(\mathcal{A}))_{M>0}$ in V, by setting

(3) $K_M(\mathcal{A}) = \{v \in V : |\lambda_\ell(v) - b_\ell| \leq M\sigma_\ell \text{ for } \ell = 1, \dots, L\}$ when $M \geq M_*$; and

(4) $K_M(\mathcal{A}) = \emptyset$ for $M < M_*$.

Our definition (3) motivates the use of the phrase "approximate linear algebra problem." We allow $L = 0$ in (1), in which case (3) says simply that $K_M(\mathcal{A}) = V$ for $M \geq M_*$.

We call $\mathcal{K}(\mathcal{A})$ "the blob arising from the ALP $\mathcal{A}$". Unlike an arbitrary blob, an ALP is specified by finitely many (real) parameters, and may therefore be manipulated by algorithms.

We call L and M_* in (1), respectively, the "length" and "threshold" of the ALP $\mathcal{A}$; and we call $\lambda_1, \dots, \lambda_L$, $b_1, \dots, b_L$ and $\sigma_1, \dots, \sigma_L$, respectively, the "functionals", "targets", and "tolerances" of $\mathcal{A}$. Note that the onset of the blob $\mathcal{K}(\mathcal{A})$ is greater than or equal to the threshold of $\mathcal{A}$, thanks to (4). The onset may be strictly greater than the threshold, since the set $K_M(\mathcal{A})$ described by (3) may be empty for some $M > M_*$.

We say that two ALPs $\mathcal{A}, \mathcal{A}'$ are C-equivalent, provided the blobs $\mathcal{K}(\mathcal{A})$, $\mathcal{K}(\mathcal{A}')$ arising from $\mathcal{A}, \mathcal{A}'$ are C-equivalent.

In the next several sections, we exhibit algorithms to perform the following tasks:

- Given an ALP $\mathcal{A}$ in a vector space V, compute the order of magnitude of onset $\mathcal{K}(\mathcal{A})$.
- Given an ALP $\mathcal{A}$ in a vector space V, compute a C-original vector for $\mathcal{K}(\mathcal{A})$.
- Let $\mathcal{A}$ be an ALP in a vector space V. Compute an ALP $\mathcal{A}'$ of length at most dim$V$, such that $\mathcal{K}(\mathcal{A}')$ and $\mathcal{K}(\mathcal{A})$ are C-equivalent.
- Let $\mathcal{A}^1$, $\mathcal{A}^2$ be ALPs in vector spaces V_1, V_2, respectively. Compute an ALP $\mathcal{A}$ in $V_1 \oplus V_2$ such that $\mathcal{K}(\mathcal{A}) = \mathcal{K}(\mathcal{A}^1) \times \mathcal{K}(\mathcal{A}^2)$.
- Let $\mathcal{A}$ be an ALP in a vector space V_1, and let $T : V_1 \to V_2$ be a linear map. Compute an ALP $\mathcal{A}'$ in V_2 such that $\mathcal{K}(\mathcal{A}')$ is C-equivalent to $T(\mathcal{K}(\mathcal{A}))$.
- Let $\mathcal{A}^1, \dots, \mathcal{A}^T$ be ALPs in a vector space V. Compute an ALP $\mathcal{A}'$ such that $\mathcal{K}(\mathcal{A}') = \mathcal{K}(\mathcal{A}^1) \cap \dots \cap \mathcal{K}(\mathcal{A}^T)$.
- Let $\mathcal{A}^1, \mathcal{A}^2$ be ALPs in a vector space V. Compute an ALP $\mathcal{A}$ such that $\mathcal{K}(\mathcal{A})$ is C-equivalent to $\mathcal{K}(\mathcal{A}^1) + \mathcal{K}(\mathcal{A}^2)$.

Here, C denotes a constant depending only on the dimensions of the relevant vector spaces. To "compute the order of magnitude" of onset $\mathcal{K}(\mathcal{A})$ is to compute a number X such that $cX \leq$ onset $\mathcal{K}(\mathcal{A}) \leq CX$ for positive constants c, C depending only on dim$V$.

In the next two sections, we describe some elementary linear algebra on ALPs, including the reduction of an ALP to "echelon form". In the following sections, we apply our result on echelon form, to carry out the tasks set down in the preceding paragraphs. We also study what happens when our ALPs depend on parameters.

To close this section, we discuss "homogeneous ALPs". An ALP $\mathcal{A}$ as in (1) is called "homogeneous" if we have $b_1 = \dots = b_L = 0$ and $M_* = 0$. Thus, for all $M \in (0, \infty)$, (3) gives

(5) $K_M(\mathcal{A}) = M\sigma(\mathcal{A})$, with

(6) $\sigma(\mathcal{A}) = \{v \in V : |\lambda_\ell(v)| \leq \sigma_\ell \text{ for } \ell = 1, \dots, L\}$.

In other words, a homogeneous ALP really describes a convex, centrally symmetric polyhedron $\sigma(\mathcal{A})$ in V, given by finitely many linear inequalities. (Recall that a convex set K in a vector space V is called centrally symmetric if $x \in K$ implies $-x \in K$. Often, we say "symmetric" instead of "centrally symmetric".)

Moreover, two blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\tilde{\mathcal{A}})$ given by homogeneous ALPs $\mathcal{A}, \tilde{\mathcal{A}}$ are C-equivalent if and only if the polyhedra $\sigma(\mathcal{A})$ and $\sigma(\tilde{\mathcal{A}})$ defined by (6) satisfy

$$(7) \quad \sigma(\mathcal{A}) \subseteq C\sigma(\tilde{\mathcal{A}}) \text{ and } \sigma(\tilde{\mathcal{A}}) \subseteq C\sigma(\mathcal{A}).$$

If (7) holds, then we say that "$\sigma(\mathcal{A})$ and $\sigma(\tilde{\mathcal{A}})$ are C-equivalent".

By specializing to the case of homogeneous ALPs, we see that the tasks we set ourselves above include, for instance, the following:

> Given two convex symmetric polyhedra $\sigma(\mathcal{A}^1)$ and $\sigma(\mathcal{A}^2)$ in the form (6), compute a homogeneous ALP $\mathcal{A}$ for which $\sigma(\mathcal{A})$ is C-equivalent to the Minkowski sum $\sigma(\mathcal{A}^1) + \sigma(\mathcal{A}^2)$.

Details are left to the reader, but we provide an elementary remark that helps with the verifications:

Suppose $\mathcal{A}$ and $\mathcal{A}'$ are ALPs, and suppose that $\mathcal{K}(\mathcal{A})$ is C-equivalent to $\mathcal{K}(\mathcal{A}')$. Then $\mathcal{A}$ is homogeneous if and only if $\mathcal{A}'$ is homogeneous.

3. Elementary Row Operations

In this section, we show how to perform elementary row operations on ALPs, analogous to the elementary processes of linear algebra. This will be used in the next section to place an ALP into "echelon form". When we implement linear algebra computations in a finite precision digital computer, certain accuracy issues arise. These are discussed in the Appendix.

Our row operations are of three types. To describe the first row operation, let

$$(1) \qquad \mathcal{A} = [(\lambda_1, \dots, \lambda_L),\ (b_1, \dots, b_L),\ (\sigma_1, \dots, \sigma_L)\ ,\ M_*]$$

be an ALP in a vector space V, and let $\pi : \{1, \dots, L\} \to \{1, \dots, L\}$ be a permutation. Then

$$\mathcal{A}^\pi = [(\lambda_{\pi 1}, \dots, \lambda_{\pi L})\ ,\ (b_{\pi 1}, \dots, b_{\pi L}),\quad (\sigma_{\pi 1}, \dots, \sigma_{\pi L})\ , M_*]$$

is again an ALP in V, and, evidently, $\mathcal{K}(\mathcal{A}) = \mathcal{K}(\mathcal{A}^\pi)$. We say that $\mathcal{A}^\pi$ arises from $\mathcal{A}$ by "permuting rows". (In the next section, we will regard each λ_ℓ as a row vector.)

Our second type of row operation arises for an ALP (1) in case there is some $\bar{L} < L$ such that $\lambda_{\bar{L}+1} = \lambda_{\bar{L}+2} = \cdots = \lambda_L = 0$. In that case, for $\bar{L} < \ell \leq L$, the estimate $|\lambda_\ell(v) - b_\ell| \leq M\sigma_\ell$, appearing in the definition of $\mathcal{K}(\mathcal{A})$, reduces to $|b_\ell| \leq M\sigma_\ell$, which is equivalent to $M \geq M^*_\ell$, for an $M^*_\ell \in [0,+\infty]$ determined trivially by b_ℓ and σ_ℓ. Consequently, we have $\mathcal{K}(\mathcal{A}) = \mathcal{K}(\bar{\mathcal{A}})$, where

$$\bar{\mathcal{A}} \;=\; [(\lambda_1,\ldots,\lambda_{\bar{L}})\,,(b_1,\ldots,b_{\bar{L}})\,,(\sigma_1,\ldots,\sigma_{\bar{L}})\,,\max(M_*,M^*_{\bar{L}+1},\ldots,M^*_{L})]\,.$$

We say that $\bar{\mathcal{A}}$ arises from $\mathcal{A}$ by "stripping away zeros".

Our third row operation on an ALP (1) arises by adding a multiple of one of the functionals $\lambda_1,\ldots,\lambda_L$ to each of the other λ's. More precisely, let $\mathcal{A}$ be the ALP given by (1), and let $1 \le \ell_0 \le L$. Suppose we are given real coefficients $\beta_1,\ldots,\beta_L$, with $\beta_{\ell_0} = 0$. We define a new ALP $\hat{\mathcal{A}}$ in V by setting

(2) $\hat{\mathcal{A}} = [(\hat{\lambda}_1,\ldots,\hat{\lambda}_L)\,,(\hat{b}_1,\ldots,\hat{b}_L),\,(\sigma_1,\ldots,\sigma_L)\,,\,M_*]$, where

(3) $\hat{\lambda}_\ell = \lambda_\ell + \beta_\ell\lambda_{\ell_0}$ and $\hat{b}_\ell = b_\ell + \beta_\ell b_{\ell_0}$ for $\ell = 1,\ldots,L$.

The blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\hat{\mathcal{A}})$ are then related by the following simple result.

Proposition. *Assume that* $|\beta_\ell|\cdot\sigma_{\ell_0} \le \sigma_\ell$ *for* $\ell = 1,2,\ldots,L$. *Then the blobs* $\mathcal{K}(\mathcal{A})$ *and* $\mathcal{K}(\hat{\mathcal{A}})$ *are 2-equivalent.*

Proof. Fix $M \ge M_*$, and let $v \in K_M(\mathcal{A})$. Then, for $\ell = 1,\ldots,L$, we have $|\lambda_\ell(v) - b_\ell| \le M\sigma_\ell$, and consequently

$$|\hat{\lambda}_\ell(v)-\hat{b}_\ell| \;=\; \big|[\lambda_\ell(v)-b_\ell] + \beta_\ell[\lambda_{\ell_0}(v)-b_{\ell_0}]\big| \;\le\; M\sigma_\ell + |\beta_\ell|\cdot M\sigma_{\ell_0} \;\le\; 2M\sigma_\ell\,,$$

since $|\beta_\ell|\sigma_{\ell_0} \le \sigma_\ell$.

This shows that

(4) $K_M(\mathcal{A}) \subseteq K_{2M}(\hat{\mathcal{A}})$,

for all $M \ge M_*$. On the other hand, (4) is obvious for $M < M_*$, since $K_M(\mathcal{A})$ is empty in that case. Thus, (4) holds for all $M > 0$. Moreover, since $\beta_{\ell_0} = 0$, (3) implies $\lambda_\ell = \hat{\lambda}_\ell - \beta_\ell\hat{\lambda}_{\ell_0}$ and $b_\ell = \hat{b}_\ell - \beta_\ell\hat{b}_{\ell_0}$ for $\ell = 1,\ldots,L$. Hence, we may repeat the proof of (4), with the rôles of $\mathcal{A}$ and $\hat{\mathcal{A}}$ interchanged, to conclude that

(5) $K_M(\hat{\mathcal{A}}) \subseteq K_{2M}(\mathcal{A})$ for all $M > 0$.

Inclusions (4), (5) tell us that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\hat{\mathcal{A}})$ are 2-equivalent. The proof of the proposition is complete. ■

When $\mathcal{A}$ and $\hat{\mathcal{A}}$ are related as in (1), (2), (3) with $\beta_{\ell_0} = 0$, then we say that $\hat{\mathcal{A}}$ arises from $\mathcal{A}$ by "row addition". If also $|\beta_\ell|\sigma_{\ell_0} \le \sigma_\ell$ for each $\ell = 1,\ldots,L$, so that the above Proposition applies, then we say that $\hat{\mathcal{A}}$ arises from $\mathcal{A}$ by "stable row addition". Note that the tolerances $\sigma_1,\ldots,\sigma_L$ and the threshold M_* remain unchanged when we pass from $\mathcal{A}$ to $\hat{\mathcal{A}}$ by row addition.

4. Echelon Form

In this section, we use the elementary row operations from the preceding section to place a given ALP $\mathcal{A}$ into "echelon form", somewhat like the standard echelon form in linear algebra. In the next section, we use our echelon form to exhibit algorithms to carry out the tasks we set ourselves in Section 2.

We take our vector space V to be $\mathbb{R}^D$, for some positive integer D. Let

(1) $\mathcal{A} = [(\lambda_1, \dots, \lambda_L), (b_1, \dots, b_L), (\sigma_1, \dots, \sigma_L), M_*]$ be an ALP in V.

Each functional λ_ℓ may be identified with a row vector $\lambda = (\lambda_{\ell 1}, \dots, \lambda_{\ell D}) \in \mathbb{R}^D$. Thus, the ALP $\mathcal{A}$ may be rewritten in the form

(2) $$\mathcal{A} = \left[(\lambda_{\ell j})_{\substack{1 \le \ell \le L \\ 1 \le j \le D}}, (b_\ell)_{1 \le \ell \le L}, (\sigma_\ell)_{1 \le \ell \le L}, M_*\right].$$

For $0 \le I \le L$, we say that an ALP $\mathcal{A}$ as in (2) is in "echelon form through row I", with "pivots" $p_1, \dots, p_I$, if the following conditions are satisfied.

$(\text{EF0})_I$ The p_i are integers, and $1 \le p_1 < p_2 < \cdots < p_I \le D$.

$(\text{EF1})_I$ $\lambda_{ip_i} \ne 0$ for $i = 1, \dots, I$.

$(\text{EF2})_I$ $\lambda_{ij} = 0$ for $1 \le j < p_i$, $i = 1, \dots, I$.

$(\text{EF3})_I$ $\lambda_{ij} = 0$ for $1 \le j \le p_I$, $i > I$.

See Figure 1 for a matrix in echelon form through row I.

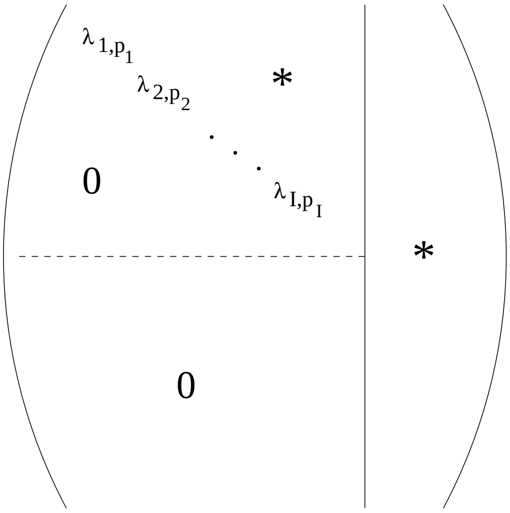

Figure 1

We adopt the convention that every ALP is in echelon form through row zero. On the other hand, an ALP (2) can never be in echelon form through row I with $\mathrm{I} > \mathsf{D}$, as one sees at once from $(\mathrm{EF0})_{\mathrm{I}}$. An ALP $\mathcal{A}$ as in (2), which is in echelon form through row $\mathsf{L}=$ length $(\mathcal{A})$, is said to be in "echelon form". Note that an ALP in $\mathbb{R}^{\mathsf{D}}$ in echelon form has length at most D.

To place a given ALP into echelon form by row operations, we repeatedly apply the following result.

Lemma 1. *Let $\mathcal{A}$ be an ALP as in* (2)*, and suppose $\mathcal{A}$ is in echelon form through row* I*. Then one of the following alternatives holds.*

- Alternative 1: $\lambda_{\ell j} = 0$ *for all* $\ell > \mathrm{I}$, $1 \leq j \leq \mathsf{D}$.
- Alternative 2: *There exists an ALP $\bar{\mathcal{A}}$ in echelon form through row* $\mathrm{I}+1$*, such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\bar{\mathcal{A}})$ are 2-equivalent, and* $\mathrm{length}(\bar{\mathcal{A}}) = \mathrm{length}(\mathcal{A})$*. Moreover, we can compute $\bar{\mathcal{A}}$ from $\mathcal{A}$ by an algorithm that uses at most* $\mathsf{CD}(\mathsf{L}+1)$ *computer operations, where* C *is a universal constant.*

Proof. Let $p_1, \ldots, p_{\mathrm{I}}$ be the pivots for $\mathcal{A}$. Suppose Alternative 1 doesn't hold. We take $p_{\mathrm{I}+1}$ to be the least j for which there exists $\ell > \mathrm{I}$ with $\lambda_{\ell j} \neq 0$. Thus,

(3) $\lambda_{\ell, p_{\mathrm{I}+1}} \neq 0$ for some $\ell > \mathrm{I}$, and

(4) $\lambda_{\ell, j} = 0$ for $j < p_{\mathrm{I}+1}, \ell > \mathrm{I}$.

Also, if $\mathrm{I} \neq 0$, then we have

(5) $p_{\mathrm{I}} < p_{\mathrm{I}+1} \leq \mathsf{D}$, as we see by comparing (3) with $(\mathrm{EF3})_{\mathrm{I}}$.

Among all $\ell > \mathrm{I}$ with $\lambda_{\ell, p_{\mathrm{I}+1}} \neq 0$, we pick $\bar{\ell}$ to minimize $\sigma_\ell / |\lambda_{\ell, p_{\mathrm{I}+1}}|$.

Once we have picked $\bar{\ell}$, we can act on $\mathcal{A}$ by permuting rows, to reduce matters to the case in which $\bar{\ell} = \mathrm{I}+1$. Thus,

(6) $\lambda_{\mathrm{I}+1\,, p_{\mathrm{I}+1}} \neq 0$, and

(7) $|\lambda_{\ell, p_{\mathrm{I}+1}} / \lambda_{\mathrm{I}+1\,, p_{\mathrm{I}+1}}| \cdot \sigma_{\mathrm{I}+1} \leq \sigma_\ell$ for all $\ell > \mathrm{I}$.

We now perform "addition of rows" on the ALP $\mathcal{A}$, as in the previous section, taking $\ell_0 = \mathrm{I}+1$, and using coefficients

(8) $\beta_\ell = -\lambda_{\ell, p_{\mathrm{I}+1}} / \lambda_{\mathrm{I}+1\,, p_{\mathrm{I}+1}}$ for all $\ell > \mathrm{I}+1$,

(9) $\beta_\ell = 0$ for $\ell \leq \mathrm{I}+1$.

Note that $\beta_{\ell_0} = \beta_{\mathrm{I}+1} = 0$, as required for addition of rows.

Note also that $|\beta_\ell| \cdot \sigma_{\ell_0} \leq \sigma_\ell$ for all ℓ, as we see from (7), (8), (9).

Hence, the Proposition from the preceding section applies. Thus, from $\mathcal{A}$, we obtain by "stable addition of rows" an ALP,

(10) $\bar{\mathcal{A}} = \big[(\bar{\lambda}_{\ell j})_{\substack{1\le\ell\le L\\ 1\le j\le D}}, \big(\bar{b}_\ell\big)_{1\le\ell\le L}, (\sigma_\ell)_{1\le\ell\le L}, M_*\big]$, such that

(11) The blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\bar{\mathcal{A}})$ are 2-equivalent, and

(12) $\bar{\lambda}_{\ell j} = \lambda_{\ell j} + \beta_\ell \lambda_{\ell_0 j}$ for all ℓ, j.

From (8), (9), (12), we see that

(13) $\bar{\lambda}_{\ell j} = \lambda_{\ell j}$ for $\ell \le I+1$, $1 \le j \le D$; and

(14) $\bar{\lambda}_{\ell j} = \lambda_{\ell j} - \big(\lambda_{\ell, p_{I+1}} / \lambda_{I+1, p_{I+1}}\big) \cdot \lambda_{I+1, j}$ for $\ell > I+1$, $1 \le j \le D$.

In particular, (4) and (13), (14) give

(15) $\bar{\lambda}_{\ell j} = 0$ for $j < p_{I+1}$, $\ell \ge I+1$.

Another application of (14) gives $\bar{\lambda}_{\ell, p_{I+1}} = 0$ for $\ell > I+1$.
Together with (15), this yields

(16) $\bar{\lambda}_{\ell j} = 0$ for $j \le p_{I+1}$, $\ell > I+1$.

It is now easy to check, using (6), (13), (15) and (16), that

(17) $\bar{\mathcal{A}}$ is in echelon form through row $I+1$.

In view of (11), (17), and the obvious remark length $(\bar{\mathcal{A}})$ = length $(\mathcal{A})$, we find ourselves in Alternative 2. Moreover, the above argument produced $\bar{\mathcal{A}}$ from $\mathcal{A}$ by an algorithm that uses at most $CD(L+1)$ operations, as the reader may easily check. Here, C denotes a universal constant.

The proof of Lemma 1 is complete. ■

Repeatedly applying Lemma 1, we can easily derive the main result of this section.

Lemma 2. *Let $\mathcal{A}$ be an ALP in $\mathbb{R}^D$, as in (2). Then there exists an* ALP $\mathcal{A}^\#$ *in echelon form in $\mathbb{R}^D$, such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^\#)$ are 2^D-equivalent, and such that* length$(\mathcal{A}^\#) \le \min\{$length$(\mathcal{A}), D\}$*. Moreover, we can compute $\mathcal{A}^\#$ from $\mathcal{A}$ in at most $CD^2(L+1)$ operations, where C is a universal constant.*

Proof. Starting at $\mathcal{A}^0 = \mathcal{A}$, which is in echelon form through row zero, we repeatedly apply Lemma 1, until we find ourselves in Alternative 1 in the statement of that lemma. Thus, we obtain a sequence of ALPs $\mathcal{A} = \mathcal{A}^0, \mathcal{A}^1, \mathcal{A}^2, \ldots$, with $\mathcal{A}^I$ in echelon form through row I, and such that the blobs $\mathcal{K}(\mathcal{A}^{\mathrm{I}})$ and $\mathcal{K}(\mathcal{A}^{\mathrm{I}+1})$ are 2-equivalent. An ALP in $\mathbb{R}^{\mathsf{D}}$ can never be in echelon form through row $\mathrm{I} > \mathsf{D}$, and therefore our sequence terminates at some $\mathcal{A}^{\mathsf{J}}$ with $\mathsf{J} \leq \mathsf{D}$. Thus, $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^{\mathsf{J}})$ are 2^{D}-equivalent, $\mathcal{A}^{\mathsf{J}}$ is in echelon form through row J, and $\mathcal{A}^{\mathsf{J}}$ satisfies Alternative 1, i.e.,

$$\mathcal{A}^{\mathsf{J}} = \left[(\bar{\lambda}_{\ell j})_{\substack{1 \leq \ell \leq \mathsf{L} \\ 1 \leq j \leq \mathsf{D}}}, (\bar{b}_\ell)_{1 \leq \ell \leq \mathsf{L}}, (\sigma_\ell)_{1 \leq \ell \leq \mathsf{L}}, \mathsf{M}_*\right],$$

with $\bar{\lambda}_{\ell j} = 0$ for $\mathsf{J} < \ell \leq \mathsf{L}$, $1 \leq j \leq \mathsf{D}$.

Stripping away zeros from $\mathcal{A}^{\mathsf{J}}$, we obtain an ALP $\mathcal{A}^{\#}$ in echelon form, with $\mathcal{K}(\mathcal{A}^{\mathsf{J}}) = \mathcal{K}(\mathcal{A}^{\#})$. Thus, $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^{\#})$ are 2^{D}-equivalent. Moreover, the above argument produces $\mathcal{A}^{\#}$ from $\mathcal{A}$ by an algorithm that uses at most $\mathsf{C}\mathsf{D}^2(\mathsf{L}+1)$ operations, since we apply Lemma 1 at most D times. Here C denotes a universal constant.

It remains only to check that length $(\mathcal{A}^{\#}) \leq \min\{\text{length}(\mathcal{A}), \mathsf{D}\}$. Recall from Lemma 1 that length $(\mathcal{A}^{\mathrm{I}+1}) =$ length $(\mathcal{A}^{\mathrm{I}})$ for each I. This yields length $(\mathcal{A}) =$ length $(\mathcal{A}^0) =$ length $(\mathcal{A}^{\mathsf{J}})$. Since $\mathcal{A}^{\#}$ arises from $\mathcal{A}^{\mathsf{J}}$ by stripping away zeros, we have length $(\mathsf{A}^{\#}) \leq$ length $(\mathcal{A}^{\mathsf{J}}) =$ length $(\mathcal{A})$. Also, since $\mathcal{A}^{\#}$ is an ALP in $\mathbb{R}^{\mathsf{D}}$ in echelon form, we have length $(\mathcal{A}^{\#}) \leq \mathsf{D}$.

Thus,

$$\text{length}(\mathcal{A}^{\#}) \leq \min\{\,\text{length}(\mathcal{A}), \mathsf{D}\},$$

completing the proof of Lemma 2. ■

5. Applications of Echelon Form

In this section, we apply our results on echelon form, to carry out the tasks we set ourselves in Section 2.

Algorithm ALP1: Given an ALP $\mathcal{A}$ of length L in $\mathbb{R}^{\mathsf{D}}$, we exhibit an ALP $\mathcal{A}^{\#}$ in $\mathbb{R}^{\mathsf{D}}$, in echelon form, such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^{\#})$ are 2^{D}-equivalent. The ALP $\mathcal{A}^{\#}$ has length at most $\min(\mathsf{L}, \mathsf{D})$.

Explanation: This is the main result in Section 4. The work and storage used by this algorithm are at most $\mathsf{C}\mathsf{D}^2(\mathsf{L}+1)$, where C is a universal constant.

Algorithm ALP2: Given an ALP $\mathcal{A}$ of length L in $\mathbb{R}^{\mathsf{D}}$, we compute a number $\mathsf{X} \geq 0$ such that $2^{-\mathsf{D}}\mathsf{X} \leq$ onset $\mathcal{K}(\mathcal{A}) \leq 2^{\mathsf{D}}\mathsf{X}$.

Explanation: Using Algorithm ALP1, we compute an ALP

(1) $\mathcal{A}^{\#} = \left[(\bar{\lambda}_{\ell j})_{\substack{1 \leq \ell \leq \bar{\mathsf{L}} \\ 1 \leq j \leq \mathsf{D}}}, (\bar{b}_\ell)_{1 \leq \ell \leq \bar{\mathsf{L}}}, (\bar{\sigma}_\ell)_{1 \leq \ell \leq \bar{\mathsf{L}}}, \bar{\mathsf{M}}_*\right]$ in echelon form, with $\bar{\mathsf{L}} \leq \mathsf{D}$, and such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^{\#})$ are 2^{D}-equivalent.

We then return $\mathsf{X} = \bar{\mathsf{M}}_*$. This algorithm uses work and storage at most $\mathsf{C}\mathsf{D}^2(\mathsf{L}+1)$ for a universal constant C.

We check that $2^{-D}X \leq$ onset $\mathcal{K}(\mathcal{A}) \leq 2^D X$. In fact, since $\mathcal{K}(\mathcal{A}^\#)$ and $\mathcal{K}(\mathcal{A})$ are 2^D-equivalent, we have $2^{-D}\cdot$ onset $\mathcal{K}(\mathcal{A}^\#) \leq$ onset $\mathcal{K}(\mathcal{A}) \leq 2^D\cdot$ onset $\mathcal{K}(\mathcal{A}^\#)$. Hence, it is enough to check that onset $\mathcal{K}(\mathcal{A}^\#) = \bar{M}_*$. This amounts to saying that

(2) $K_M(\mathcal{A}^\#) \neq \emptyset$ for $M \geq \bar{M}_*$,

as we see from the definitions of "onset" and "threshold". Recall that, for $M \geq \bar{M}_*$, we have

(3) $$K_M(\mathcal{A}^\#) = \left\{(v_1,\dots,v_D) \in \mathbb{R}^D : \left|\sum_{j=1}^{D} \bar{\lambda}_{\ell j} v_j - \bar{b}_\ell\right| \leq M\bar{\sigma}_\ell \text{ for } \ell = 1,\dots,\bar{L}\right\}.$$

The ALP $\mathcal{A}^\#$ is in echelon form. Let $1 \leq p_1 < \cdots < p_{\bar{L}} \leq D$ be the pivots of $\mathcal{A}^\#$. Then by the definition of "echelon form", we have

(4) $\bar{\lambda}_{\ell p_\ell} \neq 0$ for $\ell = 1,\dots,\bar{L}$; and $\bar{\lambda}_{\ell j} = 0$ for $1 \leq j < p_\ell$, $\ell = 1,\dots,\bar{L}$.

We will define a vector $v = (v_1, ..., v_D) \in \mathbb{R}^D$ as follows. The entries $v_D, v_{D-1}, \dots, v_1$ are determined successively by the rule:

(4a) $v_i = 0$ if i is not one of the p_ℓ; and

(4b) $$v_{p_\ell} = \bar{\lambda}_{\ell p_\ell}^{-1} \cdot \left[\bar{b}_\ell - \sum_{j=p_\ell+1}^{D} \bar{\lambda}_{\ell j} v_j\right] \quad \text{for } \ell = \bar{L}, \bar{L}-1, \dots, 1.$$

(When $p_\ell = D$ the sum in (4b) is zero.) From (4) and (4b), we see that the vector $v = (v_1,\dots,v_D) \in \mathbb{R}^D$ satisfies $\sum_{j=1}^{D} \bar{\lambda}_{\ell j} v_j = \bar{b}_\ell$ for $\ell = 1,\dots,\bar{L}$. Consequently,

(5) $v \in K_M(\mathcal{A}^\#)$ for $M \geq \bar{M}_*$, as we see from (3).

This completes the proof of (2), which shows that $2^{-D}X \leq$ onset $\mathcal{K}(\mathcal{A}) \leq 2^D X$, as claimed.

Note that (5) shows that v is a 1-original vector for $\mathcal{K}(\mathcal{A}^\#)$.

Algorithm ALP3: Given an ALP $\mathcal{A}$ of length L in $\mathbb{R}^D$, we exhibit a 2^{2D}-original vector v for $\mathcal{K}(\mathcal{A})$.

Explanation: Using Algorithm ALP1, we exhibit an ALP $\mathcal{A}^\#$ in echelon form, such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^\#)$ are 2^D-equivalent. We then determine a 1-original vector v for $\mathcal{K}(\mathcal{A}^\#)$, as in our explanation of Algorithm ALP2. Since $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^\#)$ are 2^D-equivalent, it follows that v is a 2^{2D}-original vector for $\mathcal{K}(\mathcal{A})$.

The work and storage to compute $\mathcal{A}^\#$ are at most $CD^2(L+1)$, and the length of $\mathcal{A}^\#$ is $\bar{L} \leq \min(D, L) \leq D$.

The work and storage needed to compute ν by (4a) and (4b) are at most $\mathsf{C}\mathsf{D}(\bar{\mathsf{L}}+1) \leq \mathsf{C}'\mathsf{D}^2$. Hence, altogether, the work and storage used by Algorithm ALP3 are at most $\mathsf{C}\mathsf{D}^2(\mathsf{L}+1)$, for a universal constant C.

Algorithm ALP4: Given ALPs $\mathcal{A}^1, \mathcal{A}^2$ in vector spaces $\mathsf{V}^1, \mathsf{V}^2$, respectively, we exhibit an ALP $\mathcal{A}$ in $\mathsf{V}^1 \oplus \mathsf{V}^2$, such that $\mathcal{K}(\mathcal{A}) = \mathcal{K}(\mathcal{A}^1) \times \mathcal{K}(\mathcal{A}^2)$. We have length $(\mathcal{A}) =$ length $(\mathcal{A}^1) +$ length $(\mathcal{A}^2)$.

(Lack of) Explanation: We leave the trivial algorithm for the reader. The work and storage used are at most

$$\mathsf{C} \cdot (\dim \mathsf{V}^1 + \dim \mathsf{V}^2) \cdot (1 + \text{length } \mathcal{A}^1 + \text{ length } \mathcal{A}^2),$$

for a universal constant C.

Algorithm ALP5: Given an ALP $\mathcal{A}$ in a vector space V, and given a linear map $\mathsf{T} : \mathsf{V} \to \mathsf{V}'$, we compute an ALP $\mathcal{A}'$ in V', of length at most $\dim \mathsf{V}'$, such that the blobs $\mathsf{T}(\mathcal{K}(\mathcal{A}))$ and $\mathcal{K}(\mathcal{A}')$ are 2^{D}-equivalent, where $\mathsf{D} = \dim \mathsf{V}$.

Explanation: We consider three basic special cases, and then pass to the general case. The special cases are as follows.

Case 1: $\mathsf{T} : \mathsf{V} \to \mathsf{V}'$ is an isomorphism.

Case 2: $\mathsf{T} : \mathbb{R}^{\mathsf{D}} \to \mathbb{R}^{\mathsf{D}'}$ is the injection $(\nu_1, \dots, \nu_{\mathsf{D}}) \mapsto (\nu_1, \dots, \nu_{\mathsf{D}}, 0, \dots, 0)$.

Case 3: $\mathsf{T} : \mathbb{R}^{\mathsf{D}} \to \mathbb{R}^{\mathsf{D}'}$ is the projection $(\nu_1, \dots, \nu_{\mathsf{D}}) \mapsto (\nu_{\mathsf{D}-\mathsf{D}'+1}, \dots, \nu_{\mathsf{D}})$.

In Case 1, it is obvious how to produce an ALP $\mathcal{A}'$, of the same length as $\mathcal{A}$, such that $\mathsf{T}\mathcal{K}(\mathcal{A}) = \mathcal{K}(\mathcal{A}')$.

The work and storage used to produce $\mathcal{A}'$ in Case 1 are at most

$$\mathsf{C}(\dim \mathsf{V}')^3 + \mathsf{C}(\dim \mathsf{V}')^2 \text{ length } (\mathcal{A}),$$

since we must compute T^{-1} and then compose each functional λ_ℓ (appearing in $\mathcal{A}$) with T^{-1}. Here, C is a universal constant.

In Case 2, it is obvious how to produce an ALP $\mathcal{A}'$, of length equal to $\mathsf{L} = (\mathsf{D}' - \mathsf{D}) +$ length $(\mathcal{A})$, such that $\mathsf{T}\mathcal{K}(\mathcal{A}) = \mathcal{K}(\mathcal{A}')$.

The work and storage used to produce $\mathcal{A}'$ in Case 2 are at most $\mathsf{C}\mathsf{D}'(\mathsf{L}+1)$, for a universal constant C.

In Case 3, we proceed as follows. Using Algorithm ALP1, we produce an ALP $\mathcal{A}^{\#}$ in echelon form in $\mathbb{R}^{\mathsf{D}}$, such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^{\#})$ are 2^{D}-equivalent, and such that length $(\mathcal{A}^{\#}) \leq \min(\mathsf{D},$ length $(\mathcal{A}))$. We will exhibit an ALP $\mathcal{A}'$, of length at most $\mathsf{L}' = \min\{\mathsf{D}',$ length $(\mathcal{A})\}$, such that $\mathsf{T}\mathcal{K}(\mathcal{A}^{\#}) = \mathcal{K}(\mathcal{A}')$. Since $\mathsf{T}\mathcal{K}(\mathcal{A}^{\#})$ and $\mathsf{T}\mathcal{K}(\mathcal{A})$ are 2^{D}-equivalent, it follows that $\mathcal{K}(\mathcal{A}')$ will be 2^{D}-equivalent to $\mathsf{T}\mathcal{K}(\mathcal{A})$. To compute $\mathcal{A}'$, let

$$(7) \quad \mathcal{A}^{\#} = \left[(\bar{\lambda}_{\ell j})_{\substack{1\leq \ell \leq \bar{\mathsf{L}} \\ 1 \leq j \leq \mathsf{D}}}, (\bar{\mathsf{b}}_\ell)_{1\leq\ell\leq\bar{\mathsf{L}}}, (\bar{\sigma}_\ell)_{1\leq\ell\leq\bar{\mathsf{L}}}, \bar{\mathsf{M}}_*\right],$$

and let $1 \le p_1 < \cdots < p_{\bar{L}} \le D$ be the pivots for $\mathcal{A}^{\#}$. As before, we have

(8) $\bar{\lambda}_{\ell p_\ell} \neq 0$ for $\ell = 1, \ldots, \bar{L}$; and $\bar{\lambda}_{\ell j} = 0$ for $1 \le j < p_\ell$, $\ell = 1, \ldots, \bar{L}$.

Recall that $\mathcal{K}(\mathcal{A}^{\#}) = (K_M(\mathcal{A}^{\#}))_{M>0}$, with

(9) $K_M(\mathcal{A}^{\#}) = \Big\{(v_1, \ldots, v_D) \in \mathbb{R}^D : \Big|\sum_{j=1}^{D} \bar{\lambda}_{\ell j} v_j - \bar{b}_\ell\Big| \le M\bar{\sigma}_\ell \text{ for } \ell = 1, \ldots, \bar{L}\Big\}$

when $M \ge \bar{M}_*$; and

(10) $K_M(\mathcal{A}^{\#}) = \emptyset$ when $M < \bar{M}_*$.

We define a blob $\mathcal{K}' = (K'_M)_{M>0}$, by setting

(11) $K'_M = \Big\{(v_{D-D'+1}, \ldots, v_D) \in \mathbb{R}^{D'} : \Big|\sum_{j=p_\ell}^{D} \bar{\lambda}_{\ell j} v_j - \bar{b}_\ell\Big| \le M\bar{\sigma}_\ell \text{ for } p_\ell > D - D'\Big\}$

when $M \ge \bar{M}_*$; and

(12) $K'_M = \emptyset$ when $M < \bar{M}_*$.

Then $\mathcal{K}' = \mathcal{K}(\mathcal{A}')$ for an obvious ALP $\mathcal{A}'$ in echelon form in $\mathbb{R}^{D'}$. In particular, length $(\mathcal{A}') \le D'$. We check that

(13) $K'_M = T\, K_M(\mathcal{A}^{\#})$ for all $M > 0$.

For $M < \bar{M}_*$, (13) is obvious from (10) and (12). Suppose $M \ge \bar{M}_*$.

From (9), (11) and the definition of T, we obtain $T\, K_M(\mathcal{A}^{\#}) \subseteq K'_M$.

On the other hand, let $v' = (v_{D-D'+1}, \ldots, v_D)$ belong to K'_M. We define $v_{D-D'}, \ldots, v_1$ successively, by the rule:

(13a) $v_i = 0$ if $i \le D - D'$ is not among the p_ℓ ($\ell = 1, \ldots, \bar{L}$); and

(13b) $v_{p_\ell} = \bar{\lambda}_{\ell p_\ell}^{-1}[\bar{b}_\ell - \sum_{j>p_\ell} \bar{\lambda}_{\ell j} v_j]$ if $p_\ell \le D - D'$.

This yields $v = (v_1, \ldots, v_D) \in \mathbb{R}^D$ satisfying $\sum_{j \ge p_\ell} \bar{\lambda}_{\ell j} v_j - \bar{b}_\ell = 0$ for $p_\ell \le D - D'$, thanks to (13b); and $|\sum_{j \ge p_\ell} \bar{\lambda}_{\ell j} v_j - \bar{b}_\ell| \le M\bar{\sigma}_\ell$ for $p_\ell > D - D'$, since $v' \in K'_M$. Consequently, $|\sum_{j=1}^{D} \bar{\lambda}_{\ell j} v_j - \bar{b}_\ell| \le M\bar{\sigma}_\ell$ for $\ell = 1, \ldots, \bar{L}$, thanks to (8). That is, $v \in K_M(\mathcal{A}^{\#})$. Since also $Tv = v'$, we conclude that $v' \in T\, K_M(\mathcal{A}^{\#})$. This shows that $K'_M \subseteq T\, K_M(\mathcal{A}^{\#})$, completing the proof of (13).

Thus, in Case 3, we have computed an ALP $\mathcal{A}'$ of length at most min{D', length $(\mathcal{A})$}, such that the blobs $T\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}')$ are 2^D-equivalent.

The work and storage used to produce $\mathcal{A}'$ are at most $CD^2 \cdot (1 + \text{length}(\mathcal{A}))$, for a universal constant C. This concludes our discussion of Cases 1, 2, 3 above.

We now discuss Algorithm ALP5 in the general case. Suppose we are given a linear map $\mathsf{T} : \mathsf{V} \to \mathsf{V}'$, and an ALP $\mathcal{A}$ in V. We factor T as the composition of the projection $\pi : \mathsf{V} \to \mathsf{V}/\mathsf{ker}(\mathsf{T})$, the isomorphism $[\mathsf{T}]$: $\mathsf{V}/\mathsf{ker}(\mathsf{T}) \to \mathsf{Im}(\mathsf{T})$, and the injection $\iota :$ $\mathsf{Im}\ (\mathsf{T}) \to \mathsf{V}'$. Here, of course, $\mathsf{ker}(\mathsf{T})$ stands for the kernel of T and $\mathsf{Im}(\mathsf{T})$ stands for the image of T. By picking bases in the relevant vector spaces, and by using the known Cases 1, 2, 3 above, we can exhibit ALPs $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'$ such that:

- $\mathcal{K}(\mathcal{A}_1)$ is 2^{D}-equivalent to $\pi\mathcal{K}(\mathcal{A})$ (where $\mathsf{D} = \dim \mathsf{V}$), and length $(\mathcal{A}_1) \leq$ rank T;
- $\mathcal{K}(\mathcal{A}_2) = [\mathsf{T}]\,\mathcal{K}(\mathcal{A}_1)$ and length $(\mathcal{A}_2) =$ length $(\mathcal{A}_1)$; and
- $\mathcal{K}(\mathcal{A}') = \iota\,\mathcal{K}(\mathcal{A}_2)$ and length $(\mathcal{A}') =$ length $(\mathcal{A}_2) + (\dim \mathsf{V}' -$ rank $\mathsf{T})$.

Thus, $\mathcal{K}(\mathcal{A}')$ is 2^{D}-equivalent to $\mathsf{T}\mathcal{K}(\mathcal{A})$, and length $(\mathcal{A}')$ $\leq \dim \mathsf{V}'$. This completes the implementation of Algorithm ALP5.

It is straightforward to verify that the work and storage needed for Algorithm ALP5 are at most

$$\mathsf{C} \cdot (\dim \mathsf{V} + \dim \mathsf{V}')^3 + \mathsf{C} \cdot (\dim \mathsf{V})^2 \cdot \text{length}\,(\mathcal{A})\,,$$

for a universal constant C.

Algorithm ALP6: Given ALPs $\mathcal{A}^1, \mathcal{A}^2, \dots, \mathcal{A}^{\mathsf{T}}$ in a vector space V, we compute an ALP $\mathcal{A}$ in V, such that

$$\begin{aligned} \mathcal{K}(\mathcal{A}) &= \mathcal{K}(\mathcal{A}^1) \cap \mathcal{K}(\mathcal{A}^2) \cap \cdots \cap \mathcal{K}(\mathcal{A}^{\mathsf{T}})\,, \text{ and} \\ \text{length}\ (\mathcal{A}) &= \text{length}\ (\mathcal{A}^1) + \cdots + \text{length}\ (\mathcal{A}^{\mathsf{T}})\,. \end{aligned}$$

Explanation: This algorithm is trivial. To form $\mathcal{A}'$ from $\mathcal{A}^1, \dots, \mathcal{A}^{\mathsf{T}}$, we concatenate the lists of $\lambda_\ell, b_\ell, \sigma_\ell$ from the individual $\mathcal{A}^i$; and we take the maximum of the thresholds over all the $\mathcal{A}^i$.

The total work and storage used by Algorithm ALP6 are at most

$$\mathsf{C} \cdot (\dim \mathsf{V}) \cdot \left(1 + \sum_{i=1}^{\mathsf{T}} \text{length}\ (\mathcal{A}^i)\right) \text{ for a universal constant } \mathsf{C}\,.$$

Remark. For large T, the length of the ALP $\mathcal{A}$ produced by the above algorithm will be large. Hence, it is prudent to apply Algorithm ALP1, immediately after applying Algorithm ALP6 with a large T.

Algorithm ALP7: Given ALPs $\mathcal{A}^1, \mathcal{A}^2$ in a vector space V, we compute an ALP $\mathcal{A}$ of length at most $\dim \mathsf{V}$, such that the blobs $\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\mathcal{A}^1) + \mathcal{K}(\mathcal{A}^2)$ are 2^{D}-equivalent.

Explanation: Let $T : V \oplus V \to V$ be given by $(v_1, v_2) \mapsto v_1 + v_2$. Then

$$\mathcal{K}(\mathcal{A}^1) + \mathcal{K}(\mathcal{A}^2) = T[\mathcal{K}(\mathcal{A}^1) \times \mathcal{K}(\mathcal{A}^2)] .$$

Hence, we may carry out Algorithm ALP7, by applying our previous Algorithms ALP4 and ALP5.

The total work and storage needed by Algorithm ALP7 are at most

$$C \cdot (\dim V)^3 + C \cdot (\dim V)^2 \cdot \big(\text{length}\,(\mathcal{A}^1) + \text{length}\,(\mathcal{A}^2)\big),$$

for a universal constant C.

Next, we recall from Section 2 the notion of a "homogeneous ALP". By applying Algorithms ALP1, ALP6 and ALP7 in the case of "homogeneous ALPs", we obtain the following algorithms to manipulate convex symmetric polyhedra.

Algorithm ALP8: Let V be a finite-dimensional vector space. For $i = 1, \dots, T$, let

$$\sigma^i = \{v \in V : |\lambda^i_\ell(v)| \le \sigma^i_\ell \text{ for } \ell = 1, \dots, L^i\},$$

where the λ^i_ℓ are (real) linear functionals on V, and the σ^i_ℓ belong to $[0, \infty)$. Given the λ^i_ℓ and σ^i_ℓ, we compute functionals $\bar\lambda_1, \dots, \bar\lambda_L$, and non-negative numbers $\bar\sigma_1, \dots, \bar\sigma_L$, such that $L \le \dim V$, and $\bar\sigma = \{v \in V : |\bar\lambda_\ell(v)| \le \bar\sigma_\ell$ for $\ell = 1, \dots, L\}$ satisfies

$$2^{-(\dim V)}\,\bar\sigma \subseteq \sigma^1 \cap \cdots \cap \sigma^T \subseteq 2^{+(\dim V)}\,\bar\sigma .$$

Algorithm ALP9: Let V be a finite-dimensional vector space. For $i = 1, 2$, let

$$\sigma^i = \{v \in V : |\lambda^i_\ell(v)| \le \sigma^i_\ell \text{ for } \ell = 1, \dots, L^i\}$$

where the λ^i_ℓ are (real) linear functionals on V, and the σ^i_ℓ belong to $[0, \infty)$. Given the λ^i_ℓ and σ^i_ℓ, we compute functionals $\bar\lambda_1, \dots, \bar\lambda_L$, and non-negative numbers $\bar\sigma_1, \dots, \bar\sigma_L$, such that $L \le \dim V$ and

$$\bar\sigma = \{v \in V : |\bar\lambda_\ell(v)| \le \bar\sigma_\ell \text{ for } \ell = 1, \dots, L\}$$

satisfies

$$2^{-(\dim V)}\,\bar\sigma \subseteq \sigma^1 + \sigma^2 \subseteq 2^{+(\dim V)}\,\bar\sigma .$$

The work and storage needed for Algorithm ALP8 are at most $C(\dim V)^2 \cdot (1 + L^1 + \cdots + L^T)$; for Algorithm ALP9 we need work and storage at most $C(\dim V)^3 + C(\dim V)^2 \cdot (1 + L^1 + L^2)$. Here C denotes a universal constant.

In summary, we have carried out all the tasks we set ourselves in Section 2. As long as we keep the length of our ALPs from growing, we retain good control of the work and storage used by our algorithms. We can prevent the length of our ALPs from growing, by applying Algorithm ALP1 as needed. When we intersect blobs arising from T ALPs (with T large), the work and storage are proportional to T, but the intersection is computed up to C-equivalence, with C independent of T. We can apply our algorithms to compute (up to C-equivalence) the intersection and Minkowski sum of convex, symmetric polyhedra.

6. Linear Dependence on Parameters

We want to discuss the linear dependence of some of the ALP algorithms in the previous section on the targets $(b_\ell)_{1\le\ell\le L}$ of the input ALPs. To do so conveniently, we introduce another data structure called a "PALP".

Let $\bar{N}$ be a large integer, to be fixed much later. ($\bar{N}$ will have the order of magnitude of the size of our input set E.) A (real) linear functional on $\mathbb{R}^{\bar{N}}$ is said to have "depth k" if it has the form

(1) $\mathbb{R}^{\bar{N}} \ni (\xi_1, \dots, \xi_{\bar{N}}) \mapsto \mu_1\xi_1 + \cdots + \mu_{\bar{N}}\xi_{\bar{N}}$, with $\mu_i \neq 0$ for at most k distinct values of i.

Note that a linear combination of p functionals of depth k has depth pk. We represent the functional (1) by keeping only the nonzero μ_i, along with the (increasing) sequence of i's for which $\mu_i \neq 0$. We make here the assumption, to be justified in all applications (see Section 35), that

(2) An integer index i in the range $[1, \bar{N}]$ can be stored in at most C memory words.

Thus, a depth k functional on $\mathbb{R}^{\bar{N}}$ can be held with storage $C(k+1)$; and two depth k functionals can be added with work and storage $C(k+1)$. Here and below, C denotes a universal constant.

We will work with vector spaces V, V', of dimension D and D', respectively. We write c_D, C_D, C'_D, etc. to denote constants depending only on D. Similarly, $c_{D,D'}$, $C_{D,D'}$, etc. denote constants depending only on D and D'. These constants need not be the same from one occurrence to the next.

A "parametrized ALP" or "PALP" in V is an object of the form

(3) $\underline{\mathcal{A}} = [(\underline{\lambda}_1, \dots, \underline{\lambda}_L), (\underline{b}_1, \dots, \underline{b}_L), (\underline{\sigma}_1, \dots, \underline{\sigma}_L)]$, where:

- Each $\underline{\lambda}_\ell$ is a linear functional on V;
- Each $\underline{b}_\ell$ is a linear functional on $\mathbb{R}^{\bar{N}}$; and
- Each $\underline{\sigma}_\ell$ belongs to $[0, \infty)$.

We say that a PALP (3) has "depth k" if each of the functionals $\underline{b}_\ell$ on $\mathbb{R}^{\bar{N}}$ has depth k.

As for ALPs, we call $\underline{\lambda}_1, \dots, \underline{\lambda}_L$ the "functionals" of the PALP (3), even though $\underline{b}_1, \dots, \underline{b}_L$ are now linear functionals as well. Similarly, we call $\underline{b}_1, \dots, \underline{b}_L$ and $\underline{\sigma}_1, \dots, \underline{\sigma}_L$, respectively, the "targets" and "tolerances" of the PALP (3). Also, we call L the "length" of the PALP (3). Observe that, unlike an ALP, a PALP has no threshold.

When $V = \mathbb{R}^D$, then (as for ALPs), we may regard each $\underline{\lambda}_\ell$ in (3) as a row vector, and thus rewrite our PALP in the form

(4) $\underline{\mathcal{A}} = \left[(\underline{\lambda}_{\ell j})_{\substack{1 \le \ell \le L \\ 1 \le j \le D}}, (\underline{b}_\ell)_{1 \le \ell \le L}, (\underline{\sigma}_\ell)_{1 \le \ell \le L}\right]$, where the $\underline{\lambda}_{\ell j}, \underline{\sigma}_\ell$ are real numbers, and each $\underline{b}_\ell$ is a linear functional on $\mathbb{R}^{\bar{N}}$.

We may view $\underline{\mathcal{A}}$ as an object of either form (3) or (4). Let $\underline{\mathcal{A}}$ be a PALP as in (3), and let

$$\mathcal{A} = [(\lambda_1, \dots, \lambda_L), (b_1, \dots, b_L), (\sigma_1, \dots \sigma_L), M_*]$$

be an ALP of the same length as $\underline{\mathcal{A}}$.

For a given $\xi \in \mathbb{R}^{\bar{N}}$, we say that $\underline{\mathcal{A}}$ and $\mathcal{A}$ "agree at ξ," if we have:

- $\lambda_\ell = \underline{\lambda}_\ell$ for each $\ell = 1, \dots, L$;
- $b_\ell = \underline{b}_\ell(\xi)$ for each $\ell = 1, \dots, L$; and
- $\sigma_\ell = \underline{\sigma}_\ell$ for each $\ell = 1, \dots, L$.

There is no condition here on the threshold M_*, since the PALP $\underline{\mathcal{A}}$ has no threshold.

We can make elementary row operations on PALPs, just as on ALPs. In fact, if $\underline{\mathcal{A}}$ is as in (3), and if $\pi : \{1, \dots, L\} \to \{1, \dots, L\}$ is a permutation, then by "permuting rows", we obtain the PALP

$$\underline{\mathcal{A}}^\pi = \left[(\underline{\lambda}_{\pi 1}, \dots, \underline{\lambda}_{\pi L}), (\underline{b}_{\pi 1}, \dots, \underline{b}_{\pi L}), (\underline{\sigma}_{\pi 1}, \dots, \underline{\sigma}_{\pi L})\right].$$

If $\underline{\mathcal{A}}$ is as in (3), and if $\underline{\lambda}_\ell = 0$ for all ℓ in the range $\bar{L} < \ell \le L$, then by "stripping away zeros", we obtain the PALP

$$\left[(\underline{\lambda}_1, \dots, \underline{\lambda}_{\bar{L}}), (\underline{b}_1, \dots, \underline{b}_{\bar{L}}), (\underline{\sigma}_1, \dots, \underline{\sigma}_{\bar{L}})\right].$$

Finally, if $\underline{\mathcal{A}}$ is as in (3), and if $\beta_1, \dots, \beta_L \in \mathbb{R}$ with $\beta_{\ell_0} = 0$, then by "addition of rows", we obtain the PALP

$$\left[(\underline{\lambda}_1 + \beta_1 \cdot \underline{\lambda}_{\ell_0}, \dots, \underline{\lambda}_L + \beta_L \cdot \underline{\lambda}_{\ell_0}), (\underline{b}_1 + \beta_1 \cdot \underline{b}_{\ell_0}, \dots, \underline{b}_L + \beta_L \cdot \underline{b}_{\ell_0}), (\underline{\sigma}_1, \dots, \underline{\sigma}_L)\right].$$

Suppose $\underline{\mathcal{A}}$ has depth k. Then by "permuting rows" or "stripping away zeros", we again obtain a PALP of depth k. However, by "addition of rows", we obtain from $\underline{\mathcal{A}}$ a PALP of depth $2k$.

Our row operations on PALPs may be implemented on a computer in an obvious way. A PALP in $\mathbb{R}^{\mathsf{D}}$ of length L and depth k takes up storage $\mathsf{CD}(\mathsf{k}+1)(\mathsf{L}+1)$, and a row operation on such a PALP takes work and storage $\mathsf{CD}(\mathsf{k}+1)(\mathsf{L}+1)$.

The usefulness of row operations on PALPs lies in the following observation.

(5) **Remark.** Suppose $\underline{\mathcal{A}}$ is a PALP, $\mathcal{A}$ is an ALP, and $\xi \in \mathbb{R}^{\bar{\mathsf{N}}}$. Assume that $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ. Let $\underline{\mathcal{A}}'$ and $\mathcal{A}'$ arise from $\underline{\mathcal{A}}$ and $\mathcal{A}$, respectively, either by permuting rows with respect to the same permutation, by stripping away the same zero rows, or by addition of rows with respect to the same parameters. Then again $\underline{\mathcal{A}}'$ and $\mathcal{A}'$ agree at ξ.

Thanks to the above remark, we can carry out the following algorithms on PALPs.

Algorithm PALP1: Given a PALP $\underline{\mathcal{A}}$ of depth k and length L in $\mathbb{R}^{\mathsf{D}}$, we produce a PALP $\underline{\mathcal{A}}^{\#}$ of depth $2^{\mathsf{D}}\mathsf{k}$ and length $\leq \min(\mathsf{L}, \mathsf{D})$, with the following property:

Let $\mathcal{A}$ be an ALP in $\mathbb{R}^{\mathsf{D}}$, let $\xi \in \mathbb{R}^{\bar{\mathsf{N}}}$, and let $\mathcal{A}^{\#}$ be the ALP produced from $\mathcal{A}$ by Algorithm ALP1. If $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ, then $\underline{\mathcal{A}}^{\#}$ and $\mathcal{A}^{\#}$ also agree at ξ.

Explanation: Starting with $\underline{\mathcal{A}}$, we perform exactly the same elementary row operations as in Algorithm ALP1 (only now they act on PALPs instead of ALPs). We obtain a PALP of depth $2^{\mathsf{D}}\mathsf{k}$, since we perform "addition of rows" at most D times. The desired properties of $\underline{\mathcal{A}}^{\#}$ follow easily from Remark (5). The work and storage in a row operation on a PALP of depth $2^{\mathsf{D}}\mathsf{k}$ is at most $\mathsf{C}2^{\mathsf{D}}(\mathsf{k}+1)$ larger than the corresponding work and storage for an ALP. Hence, the work and storage of Algorithm PALP1 are at most $\mathsf{C}_{\mathsf{D}}(\mathsf{k}+1)(\mathsf{L}+1)$.

There is no analogue of Algorithm ALP2 for PALPs, since for that algorithm the threshold plays an essential rôle.

Algorithm PALP3: Given a PALP $\underline{\mathcal{A}}$ of length L and depth k in $\mathbb{R}^{\mathsf{D}}$, we produce functionals $\nu_1(\xi), \dots, \nu_{\mathsf{D}}(\xi)$ of depth $\mathsf{C}_{\mathsf{D}}\mathsf{k}$ on $\mathbb{R}^{\bar{\mathsf{N}}}$, with the following property:

Let $\mathcal{A}$ be an ALP in $\mathbb{R}^{\mathsf{D}}$, let $\xi \in \mathbb{R}^{\bar{\mathsf{N}}}$, and let $(\nu_1, \dots, \nu_{\mathsf{D}}) \in \mathbb{R}^{\mathsf{D}}$ be the vector produced from $\mathcal{A}$ by Algorithm ALP3. If $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ, then $(\nu_1(\xi), \dots \nu_{\mathsf{D}}(\xi)) = (\nu_1, \dots, \nu_{\mathsf{D}})$.

Explanation: First we apply Algorithm PALP1, and then we follow the same recursive procedure as in Algorithm ALP3 to determine $\nu_{\mathsf{D}}, \nu_{\mathsf{D}-1}, \dots, \nu_1$ (only now the b_ℓ and ν_ℓ are to be regarded as linear functionals on $\mathbb{R}^{\bar{\mathsf{N}}}$). It is straightforward to check that the resulting functionals $\nu_1(\xi), \dots, \nu_D(\xi)$ are of depth $C_D k$ and have the desired property, thanks to Remark (5). As in the discussion of Algorithm PALP1, the work and storage needed for Algorithm PALP3 are at most $C_D(k+1)$ times what is required for Algorithm ALP3, since the b_ℓ and ν_ℓ are now functionals of depth $C_D k$ instead of numbers. Hence, the work and storage required for Algorithm PALP3 are at most $C_D(k+1)(L+1)$.

Let $\underline{\mathcal{A}}$ be a PALP on $\mathbb{R}^D$, $C \geq 1$ and let $(\nu_1(\xi), \dots, \nu_D(\xi))$ be a vector of linear functionals on $\mathbb{R}^{\bar{N}}$. Suppose that for any ALP $\mathcal{A}$ on $\mathbb{R}^D$ and $\xi \in \mathbb{R}^{\bar{N}}$ such that $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ, we have that $(\nu_1(\xi), \dots, \nu_D(\xi))$ is a C-original vector for $\mathcal{A}$. Then we say that $(\nu_1(\xi), \dots, \nu_D(\xi))$ is a "C-original parametrized vector for the PALP $\underline{\mathcal{A}}$".

Note that Algorithm PALP3 computes a C_D-original parametrized vector for the PALP $\underline{\mathcal{A}}$.

Algorithm PALP4: Given PALPs $\underline{\mathcal{A}}^1$, $\underline{\mathcal{A}}^2$ of depth k in vector spaces V^1, V^2, respectively, we produce a PALP $\underline{\mathcal{A}}$ of depth k in $V^1 \oplus V^2$, with the following property: Let $\mathcal{A}^1, \mathcal{A}^2$ be ALPs in V^1, V^2 respectively; let $\xi \in \mathbb{R}^{\bar{N}}$; and let $\mathcal{A}$ be the ALP in $V^1 \oplus V^2$ produced by Algorithm ALP4. If $\underline{\mathcal{A}}^i$ and $\mathcal{A}^i$ agree at ξ for $i = 1, 2$, then also $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ.

(Lack of) Explanation: We perform the same trivial manipulation as for Algorithm ALP4. The work and storage needed for this algorithm are at most

$$C(k+1) \cdot (\dim V^1 + \dim V^2) \cdot (1 + \text{length}\, \underline{\mathcal{A}}^1 + \text{length}\, \underline{\mathcal{A}}^2)$$

Algorithm PALP5: Let V, V' be vector spaces of dimension D, D' respectively. Given a PALP $\underline{\mathcal{A}}$ of length L and depth k in V, and given a linear map $T : V \to V'$, we produce a PALP $\underline{\mathcal{A}}'$ of length $\leq D'$ and depth $C_{D,D'}k$ in V', with the following property:

Let $\mathcal{A}$ be an ALP in V, let $\xi \in \mathbb{R}^{\bar{N}}$, and let $\mathcal{A}'$ be the ALP produced from $\mathcal{A}$ and T by Algorithm ALP5. If $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ, then also $\underline{\mathcal{A}}'$ and $\mathcal{A}'$ agree at ξ.

Explanation: We follow the same procedure as for Algorithm ALP5, but with the targets b_ℓ of every relevant ALP being regarded now as linear functionals on $\mathbb{R}^{\bar{N}}$ of depth $C_{D,D'}k$ rather than real numbers. The work and storage needed are at most $C_{D,D'}(k+1)(L+1)$. We omit the details.

Algorithm PALP6: Given PALPs $\underline{\mathcal{A}}^1, \dots, \underline{\mathcal{A}}^T$ of depth k in a vector space V, we compute a PALP $\underline{\mathcal{A}}$ of depth k in V, with the following property:

Let $\mathcal{A}^1, \dots, \mathcal{A}^T$ be ALPs in V; let $\xi \in \mathbb{R}^{\bar{N}}$; and let $\mathcal{A}$ be the ALP in V produced from $\mathcal{A}^1, \dots, \mathcal{A}^T$ by Algorithm ALP6. If $\underline{\mathcal{A}}^i$ and $\mathcal{A}^i$ agree at ξ for each $i = 1, \dots, T$, then also $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ.

(Lack of) Explanation: Trivial. The work and storage used are at most

$$C(k+1)(\dim V) \cdot \left(1 + \sum_{i=1}^{T} \text{ length } (\underline{\mathcal{A}}_i)\right).$$

Algorithm PALP7: Given PALPs $\underline{\mathcal{A}}^1, \underline{\mathcal{A}}^2$ of depth k in a vector space V of dimension D, we produce a PALP $\underline{\mathcal{A}}$ of depth $C_D k$ in V, with the following property:

Let $\mathcal{A}^1, \mathcal{A}^2$ be ALPs in V; let $\xi \in \mathbb{R}^{\bar{N}}$; and let $\mathcal{A}$ be the ALP in V produced from $\mathcal{A}^1, \mathcal{A}^2$ by Algorithm ALP7. If $\underline{\mathcal{A}}^i$ and $\mathcal{A}^i$ agree at ξ for $i = 1, 2$, then also $\underline{\mathcal{A}}$ and $\mathcal{A}$ agree at ξ.

Explanation: We follow the same procedure as for Algorithm ALP7, but with the targets of the relevant ALPs being regarded now as linear functionals of depth $C_D k$ on $\mathbb{R}^{\bar{N}}$, instead of real numbers. The work and storage used by this algorithm are at most $C_D(k+1) \cdot$ (length $(\underline{\mathcal{A}}^1) +$ length $(\underline{\mathcal{A}}^2) + 1)$.

7. A Lemma on Rational Functions

The following elementary result on rational functions will be used in the next section.

Lemma 1. *Let $R(t) = p(t)/q(t)$ on $(0, \infty)$, where p and q are non-zero, real polynomials of degree at most d. Then there exists a partition of $(0, \infty)$ into finitely many intervals $I_1, \dots, I_{\mu_{\max}}$ (the "hard" intervals), and $J_1, \dots, J_{\nu_{\max}}$ (the "easy" intervals), with the following properties.*

(a) *For each "easy" interval J_ν, there exists a monomial $a_\nu t^{\bar{m}_\nu}$ with $0 \neq a_\nu \in \mathbb{R}$, $\bar{m}_\nu \in \mathbb{Z}$, $|\bar{m}_\nu| \le d$, such that*

$$|R(t) - a_\nu t^{\bar{m}_\nu}| \le \frac{1}{2} |a_\nu| t^{\bar{m}_\nu} \text{ for all } t \in J_\nu .$$

(b) *Each "hard" interval I_μ has the form (y_μ, y_μ^+), with $y_\mu^+/y_\mu \le C$ and $y_\mu \in J_{\nu(\mu)}$ for some $\nu(\mu)$. Here, C depends only on d.*

(c) *The "hard" intervals are open; the "easy" intervals are relatively closed in $(0, \infty)$. Some of the J_ν may consist of a single point. Additionally, $\mu_{\max} < C$ and $\nu_{\max} < C$ for a constant C depending only on d.*

(d) *Given $p(\cdot)$ and $q(\cdot)$, we can compute the I_μ, J_ν, $a_\nu, \bar{m}_\nu$, and $\nu(\mu)$, with work and storage bounded by a constant depending only on d.*

Proof. We write c, C, C', etc., to denote constants depending only on d. For each pair of distinct non-zero monomials bt^k and $b't^{k'}$, both appearing in $p(t)$ or both appearing in $q(t)$, we introduce the interval

$$I(k, k') = \left\{t \in (0, \infty) : (5\, d)^{-1} |\, bt^k| < |b'\, t^{k'}| < (5\, d)\, |\, b\, t^k|\right\}.$$

Then $I(k, k')$ has the form $(t_{\mathsf{low}}, t_{\mathsf{high}})$, with $t_{\mathsf{high}}/t_{\mathsf{low}} \leq C$. Hence, the union U of all the above intervals is a finite union of open intervals, and we have

$$(1) \qquad \int_U dt/t \leq C.$$

We take the "hard" intervals I_μ to be the component intervals of U, and we take the "easy" intervals J_ν to be the component intervals of $(0, \infty) \smallsetminus U$. Note that each I_μ has the form (y_μ, y_μ^+) with $y_\mu \in J_{\nu(\mu)}$ for some $\nu(\mu)$. From (1) we obtain the bound $y_\mu^+/y_\mu \leq C$.

Properties (b), (c) are obvious for the above intervals I_μ, J_ν. It remains to check properties (a) and (d).

Fix one of the "easy" intervals J_ν. Thus, J_ν is one of the component intervals of $(0, \infty) \smallsetminus U$.

Let bt^k and $b't^{k'}$ be two distinct non-zero monomials, both appearing in $p(t)$. Then either

(i) $|b\, t^k| \geq (5\, d) \cdot |b' t^{k'}|$ for all $t \in J_\nu$;

(ii) $|b'\, t^{k'}| \geq (5\, d) \cdot |bt^k|$ for all $t \in J_\nu$; or

(iii) $(5\, d)^{-1}|bt^k| < |b't^{k'}| < (5\, d)|bt^k|$ for some $t \in J_\nu$.

In case (i), we say that bt^k "dominates" $b't^{k'}$; in case (ii) we say that $b't^{k'}$ "dominates" bt^k. Case (iii) cannot occur, since otherwise J_ν would contain some point in an $I(k, k') \subseteq U$. Consequently, "domination" is a linear order relation between non-zero monomials appearing in $p(t)$. Since there are only finitely many such monomials, it follows that some non-zero monomial $a_{\bar{k}} t^{\bar{k}}$ appearing in $p(t)$ dominates all the others. Thus,

$$p(t) = a_d t^d + \cdots + a_0, \quad \text{with } |a_k t^k| \leq (5\, d)^{-1} |a_{\bar{k}} t^{\bar{k}}| \text{ for all } k \neq \bar{k},\ t \in J_\nu.$$

This implies that

$$(2) \qquad |p(t) - a_{\bar{k}} t^{\bar{k}}| \leq \tfrac{1}{5} |a_{\bar{k}} t^{\bar{k}}| \text{ for all } t \in J_\nu,$$

since there are at most d non-zero monomials other than $a_{\bar{k}} t^{\bar{k}}$ appearing in $p(t)$.

A similar argument for $q(t)$ shows that

$$(3) \qquad |q(t) - b_{\bar{\ell}} t^{\bar{\ell}}| \leq \tfrac{1}{5} |\, b_{\bar{\ell}} t^{\bar{\ell}}| \text{ for all } t \in J_\nu,$$

where $b_{\bar{\ell}} t^{\bar{\ell}}$ is the dominating monomial in $q(t)$. In (2) and (3), we have $a_{\bar{k}} \neq 0$, $b_{\bar{\ell}} \neq 0$, and also $0 \leq \bar{k} \leq d$, $0 \leq \bar{\ell} \leq d$ since $p(t)$ and $q(t)$ have degree at most d.

The desired conclusion (a) for J_ν is now obvious from (2) and (3). Also, conclusion (d) is now obvious. The proof of the lemma is complete. ■

8. Non-linear Parameters

In this section, we ask for the largest $\delta > 0$ for which there exists $v \in V$ such that

$$(0) \qquad |\lambda_\ell(v) - b_\ell| \leq \sigma_\ell \delta^{-m_\ell} \text{ for } \ell = 1, \dots, L.$$

Here, as usual, V is a finite-dimensional vector space, each λ_ℓ is a functional on V, each b_ℓ is a real number, each σ_ℓ belongs to $[0, \infty)$. Each m_ℓ is a non-negative integer. In principle, we could decide this question using Tarski's decision procedure for real-closed fields [33]. However, we will content ourselves with solving an easier version of the problem, and will need no tools beyond the Lemma in the preceding section.

Throughout this section, we write c, C, C', etc. to denote constants depending only on L, $\dim V$, and $\max_{1 \leq \ell \leq L} m_\ell$. Our result on (0) is as follows.

Lemma 1. *Let V be a finite-dimensional vector space, let $\lambda_1, \dots, \lambda_L$ be linear functionals on V, let $b_1, \dots, b_L$ be real numbers, let $\sigma_1, \dots, \sigma_L$ be non-negative real numbers, and suppose that $m_1, \dots, m_L$ are non-negative integers.*

Then, with work and storage at most C, we can compute a number $\delta_{OK} \in [0, \infty]$, satisfying the following properties.

(a) *If $0 < \delta < \delta_{OK}$, then there exists $v \in V$, such that*

$$|\lambda_\ell(v) - b_\ell| \leq C \sigma_\ell \delta^{-m_\ell} \text{ for } \ell = 1, \dots, L.$$

(b) *Let $\delta > 0$. Suppose there exists $v \in V$, such that*

$$|\lambda_\ell(v) - b_\ell| \leq c \sigma_\ell \delta^{-m_\ell} \text{ for } \ell = 1, \dots, L.$$

Then $0 < \delta < \delta_{OK}$.

Proof. Let $\Lambda_0 = \{\ell : \sigma_\ell = 0\}$, $\Lambda_1 = \{\ell : \sigma_\ell \neq 0\}$, and

$$\begin{aligned} H &= \{v \in V : |\lambda_\ell(v) - b_\ell| \leq \sigma_\ell \delta^{-m_\ell} \text{ for } \ell \in \Lambda_0\} \\ &= \{v \in V : \lambda_\ell(v) = b_\ell \text{ for } \ell \in \Lambda_0\}. \end{aligned}$$

Then H is a (possibly empty) affine subspace of V. If H is empty, then we can detect the fact that H is empty by elementary linear algebra; we then have conclusions (a) and (b) with $\delta_{OK} = 0$. Hence, we may assume from now on that H is non-empty. We define

$$(1) \quad Q(v, \delta) = \sum_{\ell \in \Lambda_1} \left(\frac{\lambda_\ell(v) - b_\ell}{\sigma_\ell \delta^{-m_\ell}} \right)^2 \text{ for } v \in H,\ \delta > 0, \text{ and}$$

$$(2) \quad R(\delta) = \min\{Q(v, \delta) : v \in H\}.$$

By elementary linear algebra, the minimum in (2) is attained, and $R(\delta)$ is a rational function of δ,

$$(3) \qquad R(\delta) = \frac{\alpha_d \delta^d + \cdots + \alpha_0}{\beta_d \delta^d + \cdots + \beta_0}.$$

Here, d is an integer constant determined by L, dim V, and max m_ℓ.

We can compute the coefficients $\alpha_0, \ldots, \alpha_d$ and $\beta_0, \ldots, \beta_d$ with work and storage at most C, again by elementary linear algebra . The coefficients $\beta_0, \ldots, \beta_d$ are not all zero.

It may happen that $\alpha_0, \ldots, \alpha_d$ are all zero, i.e., $R(\delta) = 0$ for all δ. In that case, there exists a vector $v \in H$ with $Q(v, \delta) = 0$ (thanks to (2)), hence $\lambda_\ell(v) - b_\ell = 0$ for $\ell \in \Lambda_1$ (thanks to (1)), and for $\ell \in \Lambda_0$ (since $v \in H$). Thus,

$$\lambda_\ell(v) - b_\ell = 0 \quad \text{for } \ell = 1, \ldots, L,$$

and consequently the inequalities (0) admit a solution $v \in V$ for any $\delta > 0$. Therefore, after checking that $\alpha_0 = \alpha_1 = \cdots = \alpha_d = 0$, we may just set $\delta_{OK} = +\infty$, and conclusions (a) and (b) will hold. Hence, we may assume from now on that $\alpha_0, \ldots, \alpha_d$ are not all zero. With $\bar{d} = \max(m_1, \ldots, m_L)$, we have

$$(4) \qquad Q(v, \delta) \le Q(v, \delta') \le \left(\tfrac{\delta'}{\delta}\right)^{2\bar{d}} Q(v, \delta) \text{ for } v \in V \text{ and } 0 < \delta < \delta',$$

as we see easily from the definition (1). (Recall that $m_1, \ldots, m_L \ge 0$.) Consequently,

$$(5) \qquad 0 \le R(\delta) \le R(\delta') \le \left(\tfrac{\delta'}{\delta}\right)^{2\bar{d}} R(\delta) \text{ for } 0 < \delta < \delta',$$

by definition (2). In particular, (5) shows that $R(\delta)$ cannot vanish for any $\delta > 0$, since we are assuming that $R(\delta)$ doesn't vanish identically.

We now apply the lemma from the preceding section to the rational function $R(\delta)$. Let $I_\mu, J_\nu, a_\nu, \bar{m}_\nu, \nu(\mu), y_\mu, y_\mu^+$ be as in that lemma.

For each "easy" interval J_ν, we have

$$(6) \qquad |R(\delta) - a_\nu \delta^{\bar{m}_\nu}| \le \tfrac{1}{2}|a_\nu \delta^{\bar{m}_\nu}| \text{ for all } \delta \in J_\nu, \text{ with } a_\nu \ne 0 \text{ and } |\bar{m}_\nu| \le d.$$

Since $R(\delta) \ge 0$ by definition, and since $R(\delta)$ never vanishes, it follows that $a_\nu > 0$. Also, each "hard" interval I_μ has the form (y_μ, y_μ^+) with $y_\mu \in J_{\nu(\mu)}$ and $1 \le y_\mu^+/y_\mu \le C$. Applying (5) with $\delta = y_\mu$, we learn that

$$(7) \qquad R(y_\mu) \le R(\delta) \le CR(y_\mu) \text{ for all } \delta \in I_\mu.$$

Let $\hat{I}_1, \dots, \hat{I}_{s_{\max}}$ be an enumeration of the I_μ and J_ν; these intervals form a partition of $(0, \infty)$. For $s = 1, \dots, s_{\max}$, we define a monomial $\gamma_s \delta^{\mu_s}$, as follows:

- If $\hat{I}_s = J_\nu$, then we set $\gamma_s = a_\nu$ and $\mu_s = \bar{m}_\nu$.
- If $\hat{I}_s = I_\mu$, then we set $\gamma_s = R(y_\mu)$ and $\mu_s = 0$.

Thanks to (6) and (7), we have

$$(8) \qquad c\gamma_s \delta^{\mu_s} \leq R(\delta) \leq C\gamma_s \delta^{\mu_s} \text{ for } \delta \in \hat{I}_s,\ s = 1, \dots, s_{\max}.$$

For each $s = 1, \dots, s_{\max}$, we can trivially apply (8) to produce one of the following three outcomes, relevant to $\hat{I}_s$:

$(\mathcal{O}_1)_s$: We guarantee that $R(\delta) \leq C$ for all $\delta \in \hat{I}_s$.

$(\mathcal{O}_2)_s$: We guarantee that $R(\delta) \geq c$ for all $\delta \in \hat{I}_s$.

$(\mathcal{O}_3)_s$: We produce $\delta_s \in \hat{I}_s$ satisfying $c' < R(\delta_s) < C'$.

Thanks to (5), it then follows that we can produce one of the following three outcomes, relevant to $(0, \infty)$:

$(\mathcal{O}_1)$: We guarantee that $R(\delta) \leq C$ for all $\delta \in (0, \infty)$; and we define $\delta_{\mathrm{OK}} := \infty$.

$(\mathcal{O}_2)$: We guarantee that $R(\delta) \geq c$ for all $\delta \in (0, \infty)$; and we define $\delta_{\mathrm{OK}} := 0$.

$(\mathcal{O}_3)$: We produce $\delta_{\mathrm{OK}} \in (0, \infty)$ satisfying $c' < R(\delta_{\mathrm{OK}}) < C'$.

The work and storage used to arrive at an outcome $(\mathcal{O}_1)$, $(\mathcal{O}_2)$ or $(\mathcal{O}_3)$, and to compute $\delta_{\mathrm{OK}} \in [0, \infty]$, are at most C.

It remains to check that δ_{OK} satisfies properties (a) and (b) in the statement of Lemma 1.

We begin with (a). Suppose $0 < \delta < \delta_{\mathrm{OK}}$. Then $(\mathcal{O}_2)$ cannot hold, and we have

$$(9) \qquad R(\delta) \leq C''.$$

Indeed (9) holds trivially in case $(\mathcal{O}_1)$, and it follows from (5) in case $(\mathcal{O}_3)$. From (9) and (2), we conclude that $Q(\nu, \delta) \leq C''$ for some $\nu \in H$. Thanks to (1), this ν satisfies $|\lambda_\ell(\nu) - b_\ell| \leq C''' \sigma_\ell \delta^{-m_\ell}$ for $\ell \in \Lambda_1$. Moreover, $\lambda_\ell(\nu) - b_\ell = 0$ for $\ell \in \Lambda_0$, since $\nu \in H$. Thus, $|\lambda_\ell(\nu) - b_\ell| \leq C''' \sigma_\ell \delta^{-m_\ell}$ for all $\ell = 1, \dots, L$, completing the proof of (a).

We turn to (b). Suppose $\delta \in (0, \infty)$ satisfies $\delta \geq \delta_{\mathrm{OK}}$. Then $(\mathcal{O}_1)$ cannot hold, and we have

$$(10) \qquad R(\delta) \geq c''.$$

Indeed, (10) holds trivially in case $(\mathcal{O}_2)$, and it follows from (5) in case $(\mathcal{O}_3)$. From (10) and (2), we conclude that $Q(\nu,\delta) \geq c''$ for all $\nu \in H$. Recalling (1) and the definition of H, we conclude that for all $\nu \in V$ we cannot have

$$|\lambda_\ell(\nu) - b_\ell| \leq c'''\sigma_\ell\delta^{-m_\ell} \quad \text{for} \quad \ell = 1, \dots, L.$$

This proves (b), completing the proof of Lemma 1. ■

We were helped greatly by the fact that the constants in Lemma 1 are allowed to depend on L, a luxury we were denied in our earlier sections on blobs and ALPs. A major point in our proof of Lemma 1 is that, thanks to (5), the "hard" intervals I_μ are not so hard after all.

Chapter II - The Basic Families of Convex sets

9. The Callahan-Kosaraju Decomposition

In this section, we recall the results of Callahan-Kosaraju [11], together with some obvious consequences of their work, spelled out in our earlier paper [19].

Let $E \subset \mathbb{R}^n$ with $\#(E) = N$, and let $\varkappa \in (0,1)$. We write c, C, C', etc. to denote constants depending only on n and $\varkappa$. A "$\varkappa$-well-separated pairs decomposition", or "WSPD" is a finite sequence of Cartesian products,

(0) $E'_1 \times E''_1, \dots, E'_L \times E''_L,$

each contained in $E \times E$, and having the following properties:

(1) Each pair $(x', x'') \in E \times E$ with $x' \neq x''$ belongs to precisely one of the sets $E'_\ell \times E''_\ell$ $(\ell = 1, \dots, L)$. Moreover, $E'_\ell \cap E''_\ell = \emptyset$ for $\ell = 1, ..., L$.

(2) For each $\ell = 1, \dots, L$, we have $\mathrm{diam}\,(E'_\ell), \mathrm{diam}\,(E''_\ell) \leq \varkappa \cdot \mathrm{dist}\,(E'_\ell, E''_\ell)$.

Here, of course,

$$\mathrm{diam}\,(A) = \max_{x,y \in A} |x - y| \quad \text{and} \quad \mathrm{dist}(A,B) = \min_{x \in A, y \in B} |x - y|$$

for finite sets $A, B \subset \mathbb{R}^n$. It is convenient to introduce also

$$\mathrm{diam}_\infty(A) = n^{1/2} \cdot \max_{\substack{(x_1,\dots,x_n) \in A \\ (y_1,\dots,y_n) \in A}} \max_{1 \leq i \leq n} |x_i - y_i|.$$

Note that $n^{-1/2}\mathrm{diam}_\infty(A) \leq \mathrm{diam}(A) \leq \mathrm{diam}_\infty(A)$.

Callahan and Kosaraju show in [11] that there exists a WSPD (0), with $L \leq CN$; in fact, they construct one by an algorithm that uses storage at most CN, and work at most $CN \log N$.

Moreover, the WSPD whose construction is described in [19] has additional structure, that allows us to perform efficiently certain computational tasks. The sets E'_ℓ, E''_ℓ in (0) are defined in terms of two auxiliary objects $\mathcal{T}$ and $\mathcal{L}$, which we now describe.

- $\mathcal{T}$ is a collection of subsets of E. We write A or B to denote elements of $\mathcal{T}$. The sets $A \in \mathcal{T}$ form a tree under inclusion.
- $\mathcal{L}$ is a collection of pairs (Λ_1, Λ_2), with Λ_1 and Λ_2 subsets of $\mathcal{T}$.

Each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ gives rise to a Cartesian product

(3) $(\cup\Lambda_1) \times (\cup\Lambda_2) \subseteq E \times E$, where

$$\cup\Lambda = \{x \in E : x \in A \text{ for some } A \in \Lambda\} \quad \text{for } \Lambda \subseteq \mathcal{T}\,.$$

The WSPD (0) constructed in [19] consists of all the Cartesian products (3), for $(\Lambda_1, \Lambda_2) \in \mathcal{L}$.

In addition to $\mathcal{T}$ and $\mathcal{L}$, the algorithms in [19] allow us to compute and store the following auxiliary data:

- For each $A \in \mathcal{T}$, a point x_A belonging to A.
- For each $A \in \mathcal{T}$, the quantity $\mathsf{diam}_\infty(A)$.
- For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, two points $x'_{\Lambda_1}, x''_{\Lambda_2}$, with $x'_{\Lambda_1} \in \cup\Lambda_1$ and $x''_{\Lambda_2} \in \cup\Lambda_2$.
- For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, the quantities $\mathsf{diam}_\infty(\cup\Lambda_1)$ and $\mathsf{diam}_\infty(\cup\Lambda_2)$.

We can describe any given $A \in \mathcal{T}$ or $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ in a "compressed form" that uses storage at most C. In fact, the set E may be ordered in such a way that each $A \in \mathcal{T}$ and each $\cup\Lambda_i$ [$i = 1$ or $2, (\Lambda_1, \Lambda_2) \in \mathcal{L}$] is an interval. Moreover, we can efficiently recover $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ from the intervals $\cup\Lambda_1$ and $\cup\Lambda_2$. Hence, it is enough to store the endpoints of the relevant intervals. Whenever we store or specify $A \in \mathcal{T}$ or $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, we always use the "compressed form".

Using the above (and additional) properties of $\mathcal{T}$ and $\mathcal{L}$, we can perform the following computations of lists.

(a) Given an $A \in \mathcal{T}$, we compute a list of all the elements of A.

This takes work W_A (to be discussed below), and storage at most CN.

(b) Given a $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, we compute a list of all the elements $A \in \Lambda_1$, and a list of all the elements $B \in \Lambda_2$.

This takes work $W(\Lambda_1, \Lambda_2)$ (to be discussed below), and storage at most CN.

(c) Given an $A \in \mathcal{T}$, we compute a list of all the $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ for which $\Lambda_1 \ni A$.

This takes work W'_A (to be discussed below), and storage at most CN.

(d) Given an $x \in E$, we compute a list of all the $A \in \mathcal{T}$ for which $A \ni x$.

This takes work W_x (to be discussed below), and storage at most CN.

Regarding the work of the above computations, we have

$$(4) \qquad \sum_{A \in \mathcal{T}} W_A + \sum_{(\Lambda_1,\Lambda_2) \in \mathcal{L}} W(\Lambda_1, \Lambda_2) + \sum_{A \in \mathcal{T}} W'_A + \sum_{x \in E} W_x \leq CN \log N.$$

The Callahan-Kosaraju decomposition constructed in [19] is shown there to satisfy

$$(5) \quad \#\{\text{nodes in } \mathcal{T}\} \leq CN\,,\ \#\{(\Lambda_1, \Lambda_2) \in \mathcal{L}\} \leq CN,$$

$$(6) \quad \sum_{A \in \mathcal{T}} \#(A) \leq CN \log N\,,\ \text{and} \sum_{(\Lambda_1,\Lambda_2) \in \mathcal{L}} \left[\#(\Lambda_1) + \#(\Lambda_2)\right] \leq CN \log N.$$

In the next section, we use the above "Callahan-Kosaraju decomposition" to construct a family of blobs and ALPs that plays a basic rôle in our work.

10. The Basic Blobs and ALPs: Definitions and Computations

In this section, we recall from [19] an important family of blobs and ALPs. We work in $\mathcal{P}$, the vector space of (real) $(m-1)^{\text{rst}}$ degree polynomials on $\mathbb{R}^n$. Let $D = \dim \mathcal{P}$.

For $x \in \mathbb{R}^n$ and $r \geq 0$, we recall from Section 1 the useful blob

$$(0) \qquad \mathcal{B}(x,r) = (M \cdot B(x,r))_{M \geq 0} \text{ in } \mathcal{P}, \text{ where}$$

$$(1) \qquad M \cdot B(x,r) = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M\, r^{m-|\alpha|} \text{ for } |\alpha| \leq m-1\}.$$

This blob arises from an obvious ALP of length D in $\mathcal{P}$. Recall that we are given a finite subset $E \subset \mathbb{R}^n$ and functions $\sigma : E \to [0,\infty)$ and $f : E \to \mathbb{R}$. We use the Callahan-Kosaraju decomposition for E, with $\varkappa = 1/2$. We retain the notation of the previous section, except that in this section C denotes a constant depending only on m and n. Also, for $\ell \geq 0$, we write c_ℓ, C_ℓ, C'_ℓ, etc., to denote constants depending only on ℓ, m and n.

We now construct from $\mathsf{E}, \mathsf{f}, \sigma$ a family of blobs $\Gamma(x,\ell) = (\Gamma(x,\ell,M))_{M>0}$ in $\mathcal{P}$, parametrized by $x \in E$ and $\ell \geq 0$. This is exactly the same family of blobs that was constructed in [19]. For the convenience of the reader, we provide here a detailed exposition of the construction. We proceed by induction on ℓ.

For $\ell = 0$, we define

$$\text{(2)} \qquad \Gamma(x,0,M) = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M \ \text{ for } |\alpha| \leq m-1, \\ \text{and } |P(x) - f(x)| \leq M\sigma(x)\}$$

for all $x \in E$.

For the inductive step, fix $\ell \geq 0$, and suppose we have defined the blobs $\Gamma(x,\ell)$ for all $x \in E$. We will define the blob $\Gamma(x,\ell+1)$ for all $x \in E$.

To do so, we use the Callahan-Kosaraju decomposition, and proceed in five steps, as follows.

Step 1: For each $A \in \mathcal{T}$, we form the blob

$$\text{(3)} \qquad \Gamma(A,\ell) = \bigcap_{x \in A} \left[\Gamma(x,\ell) + \mathcal{B}(x, \mathrm{diam}_\infty(A))\right].$$

Step 2: For each $(\Lambda_1,\Lambda_2) \in \mathcal{L}$, and for $i = 1,2$, we form the blob

$$\text{(4)} \qquad \Gamma_i(\Lambda_i,\ell) = \bigcap_{A \in \Lambda_i} \left[\Gamma(A,\ell) + \mathcal{B}(x_A, \mathrm{diam}_\infty(\cup\Lambda_i))\right].$$

Step 3: For each $(\Lambda_1,\Lambda_2) \in \mathcal{L}$, we form the blob

$$\text{(5)} \qquad \bar{\Gamma}(\Lambda_1,\Lambda_2,\ell) = \Gamma_1(\Lambda_1,\ell) \cap \left[\Gamma_2(\Lambda_2,\ell) + \mathcal{B}(x_{\Lambda_1}, |x_{\Lambda_1} - x_{\Lambda_2}|)\right].$$

Step 4: For each $A \in \mathcal{T}$, we form the blob

$$\text{(6)} \qquad \Gamma'(A,\ell+1) = \bigcap_{\substack{(\Lambda_1,\Lambda_2)\in\mathcal{L} \\ \Lambda_1 \ni A}} \bar{\Gamma}(\Lambda_1,\Lambda_2,\ell).$$

Step 5: For each $x \in E$, we define the blob

$$\text{(7)} \qquad \Gamma(x,\ell+1) = \Gamma(x,\ell) \cap \bigcap_{\substack{A\in\mathcal{T} \\ A \ni x}} \Gamma'(A,\ell+1).$$

This completes the inductive definition of the blobs $\Gamma(x,\ell)$.

By following the above induction on ℓ, we can compute ALPs $\mathcal{A}(x, \ell)$ for each $x \in E$ and $\ell \geq 0$, such that $\Gamma(x, \ell)$ is C_ℓ-equivalent to $\mathcal{K}(\mathcal{A}(x, \ell))$, the blob arising from $\mathcal{A}(x, \ell)$. In fact, for $\ell = 0$, the blob $\Gamma(x, 0)$ is already given by an obvious ALP of length $D + 1$.

For the inductive step, fix $\ell \geq 0$, and suppose we have already computed $\mathcal{A}(x, \ell)$ for all $x \in E$. Then, by using Algorithms ALP1, ALP6, ALP7 from Section 5, we may follow our five steps to produce ALPs as follows.

Step 1′: For each $A \in \mathcal{T}$, we compute an ALP $\mathcal{A}(A, \ell)$ of length $\leq D$, such that $\mathcal{K}(\mathcal{A}(A, \ell))$ is C_ℓ-equivalent to

$$\bigcap_{x \in A} \left[\mathcal{K}(\mathcal{A}(x, \ell)) + \mathcal{B}(x, \mathsf{diam}_\infty(A))\right].$$

Step 2′: For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, and for $i = 1, 2$, we compute an ALP $\mathcal{A}_i(\Lambda_i, \ell)$ of length $\leq D$, such that $\mathcal{K}(\mathcal{A}_i(\Lambda_i, \ell))$ is C_ℓ-equivalent to

$$\bigcap_{A \in \Lambda_i} \left[\mathcal{K}(\mathcal{A}(A, \ell)) + \mathcal{B}(x_A, \mathsf{diam}_\infty(\cup \Lambda_i))\right].$$

Step 3′: For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, we compute an ALP $\bar{\mathcal{A}}(\Lambda_1, \Lambda_2, \ell)$ of length $\leq D$, such that $\mathcal{K}(\bar{\mathcal{A}}(\Lambda_1, \Lambda_2, \ell))$ is C_ℓ-equivalent to

$$\mathcal{K}(\mathcal{A}_1(\Lambda_1, \ell)) \cap \left[\mathcal{K}(\mathcal{A}_2(\Lambda_2, \ell)) + \mathcal{B}(x_{\Lambda_1}, |x_{\Lambda_1} - x_{\Lambda_2}|)\right].$$

Step 4′: For each $A \in \mathcal{T}$, we compute an ALP $\mathcal{A}'(A, \ell + 1)$ of length $\leq D$, such that $\mathcal{K}(\mathcal{A}'(A, \ell + 1))$ is C_ℓ-equivalent to

$$\bigcap_{\substack{(\Lambda_1, \Lambda_2) \in \mathcal{L} \\ \Lambda_1 \ni A}} \mathcal{K}(\bar{\mathcal{A}}(\Lambda_1, \Lambda_2, \ell)).$$

Step 5′: For each $x \in E$, we compute an ALP $\mathcal{A}(x, \ell + 1)$ of length $\leq D$, such that $\mathcal{K}(\mathcal{A}(x, \ell + 1))$ is C_ℓ-equivalent to

$$\mathcal{K}(\mathcal{A}(x, \ell)) \cap \bigcap_{\substack{A \in \mathcal{T} \\ A \ni x}} \mathcal{K}(\mathcal{A}'(A, \ell + 1)).$$

This completes our description of the computation of the ALPs $\mathcal{A}(x, \ell)$. We will need the $\mathcal{A}(x, \ell)$ only for $\ell = 0, 1, \ldots, \ell_*$, with ℓ_* depending only on m and n.

Note that all the ALPs computed above have length $\leq D$, except for $\mathcal{A}(x, 0)$, which has length $D + 1$.

Comparing Steps $1', \ldots, 5'$ with Steps $1, \ldots, 5$, we see that

(8) $\mathcal{K}(\mathcal{A}(x, \ell + 1))$ is C'_ℓ-equivalent to $\Gamma(x, \ell + 1)$, for all $x \in E$;

provided

(9) $\mathcal{K}(\mathcal{A}(x,\ell))$ is C_ℓ-equivalent to $\Gamma(x,\ell)$, for all $x \in E$.

Since also $\mathcal{K}(\mathcal{A}(x,0)) = \Gamma(x,0)$ for all $x \in E$, an obvious induction on ℓ shows that (9) holds for all $\ell \geq 0$. Thus, we can compute the $\Gamma(x,\ell)$ up to C_ℓ-equivalence. In [19], we showed that the computation of the ALPs $\mathcal{A}(x,\ell)$ for all $x \in E$ and $\ell = 0,\dots,\ell_*$ requires work at most $CN \log N$ and storage at most CN. This is a straightforward application of our results on the basic ALP algorithms, together with the estimate (4) from the preceding section. (In [19] we computed with "ellipsoidal blobs" rather than ALPs. This has no effect on our estimates for the work or storage used by our algorithms.)

Starting from E, σ, f, we have defined the blobs $\Gamma(x,\ell)$ and computed them up to C_ℓ-equivalence. Next, we introduce a variant. Suppose we repeat our construction of the $\Gamma(x,\ell)$, starting from $E, \sigma, 0$ in place of E, σ, f. Then, in place of the $\Gamma(x,\ell)$, we will obtain a new family of blobs $\Gamma^0(x,\ell) = (\Gamma^0(x,\ell,M))_{M>0}$ in $\mathcal{P}$, determined by E and σ. Also, in place of the ALPs $\mathcal{A}(x,\ell)$, we will obtain a family of ALPs $\mathcal{A}^0(x,\ell)$ of length $\leq D+1$, such that

(10) The blob $\mathcal{K}(\mathcal{A}^0(x,\ell))$ is C_ℓ-equivalent to $\Gamma^0(x,\ell)$ for each $x \in E$, $\ell \geq 0$.

We can compute the $\mathcal{A}^0(x,\ell)$ (for all $x \in E$, $\ell = 0,\dots,\ell_*$) using work at most $CN \log N$ and storage at most CN.

An easy induction on ℓ shows that the $\Gamma^0(x,\ell,M)$ have the form

$$\Gamma^0(x,\ell,M) = M\sigma(x,\ell)$$

for a convex, symmetric set $\sigma(x,\ell) \subseteq \mathcal{P}$, and that $\mathcal{A}^0(x,\ell)$ is a "homogeneous ALP"; see Section 2. Consequently, the blob

$$\mathcal{K}(\mathcal{A}^0(x,\ell)) = (K_M(\mathcal{A}^0(x,\ell)))_{M>0}$$

has the form

$$K_M(\mathcal{A}^0(x,\ell)) = M\sigma(\mathcal{A}^0(x,\ell))$$

for a convex, centrally symmetric polyhedron $\sigma(\mathcal{A}^0(x,\ell)) \subseteq \mathcal{P}$, arising from $\mathcal{A}^0(x,\ell)$ as in (5), (6) in Section 2.

The blob equivalence (10) therefore becomes

(11) $c_\ell\, \sigma(\mathcal{A}^0(x,\ell)) \subseteq \sigma(x,\ell) \subseteq C_\ell\, \sigma(\mathcal{A}^0(x,\ell))$ for all $x \in E$, $\ell \geq 0$.

Thus, we have computed the convex sets $\sigma(x,\ell)$ up to C_ℓ-equivalence. (See also Section 13 below for more details regarding the construction of the $\sigma(x,\ell)$.)

11. The Basic Blobs and ALPs: Linear Dependence on Parameters

In the preceding section, we associated to E, σ, f a family of blobs $\Gamma(x, \ell)$ and ALPs $\mathcal{A}(x, \ell)$ $(x \in E, \ell \geq 0)$.

In this section, we suppose that E and σ are held fixed, while $f = f_\xi$ depends linearly on a parameter $\xi \in \mathbb{R}^{\bar{N}}$. We assume that

(1) $\xi \mapsto f_\xi(x)$ is a depth k linear functional on $\mathbb{R}^{\bar{N}}$, for each fixed $x \in E$.

We will also use the notation $\vec{f}(x, \xi) = f_\xi(x)$, mostly in later sections. Here, k and $\bar{N}$ are given, and $\bar{N}$ is assumed to satisfy (2) of Section 6. We ask how the ALPs $\mathcal{A}(x, \ell)$ depend on the parameter ξ. To answer this question, we bring in our results on PALPs.

We start with a few preliminary remarks. In this section, we write C, C', etc. for constants depending only on m and n; while C_ℓ, C'_ℓ, etc. denote constants depending only on ℓ, m, n. We set $D = \dim \mathcal{P}$. Recall the definition of the blobs $\mathcal{B}(x, r)$. Recall from the preceding section that the blob $\mathcal{B}(x, r)$ arises from an obvious ALP in $\mathcal{P}$, which we shall call $\mathcal{A}_{\mathcal{B}(x,r)}$. Since the targets in $\mathcal{A}_{\mathcal{B}(x,r)}$ are all zero, it is trivial to construct a PALP $\underline{\mathcal{A}}_{\mathcal{B}(x,r)}$ of depth zero in $\mathcal{P}$, such that $\underline{\mathcal{A}}_{\mathcal{B}(x,r)}$ and $\mathcal{A}_{\mathcal{B}(x,r)}$ agree at every $\xi \in \mathbb{R}^{\bar{N}}$.

We now discuss the ξ-dependence of the ALPs $\mathcal{A}(x, \ell)$ constructed from E, σ, f when $f = f_\xi$. Let us call these ALPs $\mathcal{A}_\xi(x, \ell)$.

By induction on $\ell \geq 0$, we can construct a family of PALPs $\underline{\mathcal{A}}(x, \ell)$ (for $x \in E, \ell \geq 0$), of depth $C_\ell k$ and length $\leq D + 1$ in $\mathcal{P}$, with the following property:

$(2)_\ell$ Let $x \in E$ and $\xi \in \mathbb{R}^{\bar{N}}$. Then $\underline{\mathcal{A}}(x, \ell)$ agrees with $\mathcal{A}_\xi(x, \ell)$ at ξ.

To see this, we first recall that

$$(3) \qquad \Gamma(x, 0, M) = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M \text{ for } |\alpha| \leq m - 1, \\ \text{and } |P(x) - f(x)| \leq M\sigma(x)\},$$

and that $\mathcal{A}(x, 0)$ is the obvious ALP giving rise to the blob (3). Thus, for $\ell = 0$, $(2)_\ell$ holds for the PALP $\underline{\mathcal{A}}(x, 0)$ in $\mathcal{P}$, defined as follows: The functionals for $\underline{\mathcal{A}}(x, 0)$ are $\lambda_\alpha : P \mapsto \partial^\alpha P(x)$ for $|\alpha| \leq m - 1$, and $\lambda_{\text{extra}} : P \mapsto P(x)$.

The targets corresponding to these functionals are 0 for the λ_α, and $f_\xi(x)$ for λ_{extra}.

The tolerances corresponding to the above functionals are 1 for the λ_α, and $\sigma(x)$ for λ_{extra}.

Note that $\underline{\mathcal{A}}(x, 0)$ has depth k.

Thus, we have constructed $\underline{\mathcal{A}}(x, 0)$ having the desired properties.

Next, fix $\ell \geq 0$, and suppose we have already constructed PALPs $\underline{\mathcal{A}}(x, \ell)$ of depth $C_\ell k$ and length $\leq D+1$ in $\mathcal{P}$, for each $x \in E$, satisfying property $(2)_\ell$. We will construct PALPs $\underline{\mathcal{A}}(x, \ell+1)$ of depth $C'_\ell k$ and length $\leq D$ in $\mathcal{P}$, for each $x \in E$, satisfying $(2)_{\ell+1}$. To do so, we recall Steps $1', \ldots, 5'$ in the preceding section. We used these five steps to pass from the $\mathcal{A}(x, \ell)$ $(x \in E)$ to the $\mathcal{A}(x, \ell+1)$ $(x \in E)$. To implement Steps $1', \ldots, 5'$, we used our Algorithms ALP1, ALP6, ALP7 in an obvious way. We can now carry out the analogous five steps to pass from the PALPs $\underline{\mathcal{A}}(x, \ell)$ $(x \in E)$ to the PALPs $\underline{\mathcal{A}}(x, \ell+1)$ $(x \in E)$. We simply use Algorithms PALP1, PALP6, PALP7 in place of ALP1, ALP6, ALP7. From $(2)_\ell$ and the defining properties of Algorithms PALP1, PALP6, PALP7, we obtain the desired property $(2)_{\ell+1}$ for the PALPs $\underline{\mathcal{A}}(x, \ell+1)$ $(x \in E)$, and we see also that the $\underline{\mathcal{A}}(x, \ell+1)$ have depth $C'_\ell k$.

This completes the induction on ℓ.

Thus we can compute the $\underline{\mathcal{A}}(x, \ell)$, by an analogue of our earlier computation of the $\mathcal{A}(x, \ell)$. We shall need the $\underline{\mathcal{A}}(x, \ell)$ only for $0 \leq \ell \leq \ell_*$; as before, ℓ_* is an integer constant, depending only on m and n.

The work and storage needed to compute the PALPs $\underline{\mathcal{A}}(x, \ell)$ $(x \in E, 0 \leq \ell \leq \ell_*)$ are at most $C(k+1)$ times the corresponding work and storage for the $\mathcal{A}(x, \ell)$. (The factor $C(k+1)$ arises, because we work with targets that are depth-Ck functionals instead of real numbers.) Thus, the $\underline{\mathcal{A}}(x, \ell)$ $(x \in E, 0 \leq \ell \leq \ell_*)$ can be computed using work at most $C(k+1)N \log N$, and storage at most $C(k+1)N$.

12. Whitney t-Convexity

In this section we start by proving some properties of the useful blob $\mathcal{B}(x, r) = (MB(x, r))_{M>0}$ defined in Section 1. Recall that for $x \in \mathbb{R}^n, r \geq 0$ we set

$$(1) \qquad B(x, r) = \left\{ P \in \mathcal{P} : |\partial^\beta P(x)| \leq r^{m-|\beta|} \text{ for } |\beta| \leq m-1 \right\},$$

and for $y \in \mathbb{R}^n$ we define $B(x, y) = B(x, |x-y|)$. From (1) we immediately conclude that for $x \in \mathbb{R}^n$ and $r, A \geq 0$,

$$(2) \qquad B(x, Ar) \subseteq \max\{A^m, A\} \cdot B(x, r) \, .$$

In this section $c, C, \tilde{C}$ etc. stand for constants depending only on m and n. For two polynomials $P, Q \in \mathcal{P}$, we denote by $P \odot_x Q$ the product of P and Q as $(m-1)$-jets at x. That is, $P \odot_x Q$ is the one and only $S \in \mathcal{P}$ such that

$$\partial^\alpha S(x) = \partial^\alpha (PQ)(x) \quad \text{for} \quad |\alpha| \leq m-1 \, .$$

For $\Omega_1, \Omega_2 \subset \mathcal{P}$ and $x \in \mathbb{R}^n$ we also write

$$\Omega_1 \odot_x \Omega_2 = \{P \odot_x Q : P \in \Omega_1, Q \in \Omega_2\}.$$

Lemma 1. *Let* $x, y \in \mathbb{R}^n, r > 0$*. Then,*

$$B(x,r) \subseteq CB(y, r + |x-y|), \tag{3}$$

and in particular $B(x,y)$ *is* C*-equivalent to* $B(y,x)$.

Suppose $r \le \bar{r}$*. Then,*

$$B(x,r) \odot_x B(x,\bar{r}) \subseteq C\bar{r}^m B(x,r)\,. \tag{4}$$

Let $P, Q \in B(x,r)$*, and suppose* $|x-y| < r$*. Then,*

$$(P \odot_y Q) - (P \odot_x Q) \in Cr^m B(y,x)\,. \tag{5}$$

Proof. Start with verifying (3). Let $P \in B(x,r)$ be a polynomial. Then, for any $|\alpha| \le m-1$ we have

$$|\partial^\alpha P(x)| \le r^{m-|\alpha|}\,.$$

By Taylor's theorem

$$\begin{aligned}|\partial^\alpha P(y)| &= \left| \sum_{|\beta| \le m-1-|\alpha|} \frac{\partial^{\alpha+\beta}P(x)}{\beta!}(y-x)^\beta \right| \\ &\le C \sum_\beta r^{m-(|\alpha|+|\beta|)}|x-y|^{|\beta|} \le C'(r+|x-y|)^{m-|\alpha|}\end{aligned}$$

and (3) follows from the definition of $B(y, r+|x-y|)$.

Next, we establish (4). Let $P \in B(x,r), Q \in B(x,\bar{r})$. Then, for any $|\alpha| \le 2(m-1)$,

$$\begin{aligned}&|\partial^\alpha(PQ)(x)| = \\ &= \left| \sum_{\beta, \alpha-\beta \in \mathcal{M}} \frac{\alpha!}{\beta!(\alpha-\beta)!}(\partial^\beta P)(x) \cdot (\partial^{\alpha-\beta}Q)(x) \right| \le C' \sum_{\beta,\alpha-\beta\in\mathcal{M}} r^{m-|\beta|}\,\bar{r}^{m-|\alpha|+|\beta|} \\ &\le C\bar{r}^m r^{m-|\alpha|}\end{aligned} \tag{6}$$

since $r \le \bar{r}$. Here, $\mathcal{M}$ denotes the set of all multi-indices γ of order $|\gamma| \le m-1$. On the other hand,

$$\partial^\alpha(PQ)(x) = \partial^\alpha(P \odot_x Q)(x) \quad \text{for } |\alpha| \le m-1\,. \tag{7}$$

From (6) and (7) we conclude (4).

It remains to prove (5). To that end, let $P, Q \in B(x,r)$ and $y \in \mathbb{R}^n$ be such that $|x-y| < r$. By (6) for any $|\beta| \le 2(m-1)$,

$$\left|\partial^\beta(PQ)(x)\right| < Cr^{2m-|\beta|}.$$

Note also that $\partial^\beta(PQ - P \odot_x Q)(x) = 0$ for $|\beta| \le m-1$.

Therefore, for any $|\alpha| \leq m-1$,

$$(8)\ |\partial^\alpha(PQ - P \odot_x Q)(y)| = \left| \sum_\beta \frac{1}{\beta!} \partial^{\alpha+\beta}(PQ - P \odot_x Q)(x)(y-x)^\beta \right|$$
$$= \left| \sum_{m \leq |\beta|+|\alpha| \leq 2(m-1)} \frac{1}{\beta!} \partial^{\alpha+\beta}(PQ)(x) \cdot (y-x)^\beta \right|$$
$$\leq C \sum_{m \leq |\beta|+|\alpha| \leq 2(m-1)} r^{2m-(|\alpha|+|\beta|)} |y-x|^{|\beta|} \leq C' r^m |y-x|^{m-|\alpha|}$$

since $r > |y-x|$. According to the definition of $P \odot_y Q$, the inequality (8) implies that for any $|\alpha| \leq m-1$,

$$|\partial^\alpha [(P \odot_y Q) - (P \odot_x Q)](y)| \leq C r^m |x-y|^{m-|\alpha|}.$$

Hence (5) follows. ■

Remark. For $P, Q \in \mathcal{P}^+$ and $x \in \mathbb{R}^n$, we write $P \odot_x^+ Q$ to denote the unique polynomial in $\mathcal{P}^+$ for which $\partial^\beta(P \odot_x^+ Q - PQ)(x) = 0$ for all $|\beta| \leq m$. We will also make use of the following fact, whose proof is completely analogous to that of (4). Suppose $x \in \mathbb{R}^n$ and $r \leq \bar{r}$. Then,

$$(9)\qquad B^+(x,r) \odot_x^+ B^+(x,\bar{r}) \subseteq C\bar{r}^m \cdot B^+(x,r).$$

We will need the following

Definition. *Let $x \in \mathbb{R}^n$, $A \geq 1$ and let σ be a convex, symmetric, non-empty subset of $\mathcal{P}$. We say that "σ is Whitney t-convex at x, with Whitney constant A" if for any $r > 0$,*

$$(10)\qquad [\sigma \cap B(x,r)] \odot_x B(x,r) \subseteq A r^m \sigma.$$

The above definition is an instance of "Whitney ω-convexity"; see [14, 16]. (However, [14, 16] require (10) or its variants, only for $r \leq 1$.) A basic example of a Whitney t-convex set is $B(x,r)$; for any $x \in \mathbb{R}^n, r > 0$ the set $B(x,r)$ is Whitney t-convex at x with Whitney constant $C > 0$, depending only on m and n. This follows from (4).

Also, if σ_1, σ_2 are Whitney t-convex at x with Whitney constants A_1, A_2 respectively, then $\sigma_1 \cap \sigma_2$ is Whitney t-convex at x with Whitney constant $\max\{A_1, A_2\}$. We thus conclude the following lemma.

Lemma 2. *Let $\sigma \subset \mathcal{P}, x \in \mathbb{R}^n, A \geq 1, r > 0$. Suppose that σ is Whitney t-convex at x with Whitney constant A. Then*

$$\sigma \cap B(x,r)$$

is Whitney t-convex at x with Whitney constant CA, where C is a constant depending only on m and n.

Another useful observation is as follows. Let $\sigma, \sigma' \subset \mathcal{P}$ be convex symmetric sets, and let $a, b > 0$ and $A \geq 1$ be constants. Suppose $\frac{1}{a}\sigma \subseteq \sigma' \subseteq b\sigma$, and suppose σ is Whitney t-convex at x with Whitney constant A. Then, as may be easily verified from the definition (10), the set σ' is also Whitney t-convex at x, with Whitney constant $a \max\{1, b\} \cdot A$.

We view Whitney convex sets as quantitative analogues of ideals in $\mathcal{P}$ with respect to $\odot_x$. For instance, ideals with respect to $\odot_x$ are always Whitney t-convex at x with Whitney constant A, for any $A \geq 1$.

In the proof of the next lemma we will use the following elementary observation. Suppose that A, K, T are symmetric convex sets in a vector space V. Then,

$$(11) \quad K \subseteq T \quad \Longrightarrow \quad (A + K) \cap T \subseteq (A \cap 2T) + K.$$

Indeed, if $x \in (A + K) \cap T$, then for some $k \in K$ we have that $x - k \in A$. Also, $x - k \in T - K \subseteq 2T$, and hence $x \in (A \cap 2T) + K$.

Lemma 3. *Let $x, y \in \mathbb{R}^n, A \geq 1$, and assume that $\sigma \subset \mathcal{P}$ is Whitney t-convex at y with Whitney constant A. Then, for any $\delta > |x - y|$,*

$$\sigma + B(y, \delta)$$

is Whitney t-convex at x with Whitney constant CA, where C depends solely on m and n.

Proof. Let $r > 0$. According to (10) we need to show that

$$(12) \qquad \{[\sigma + B(y, \delta)] \cap B(x, r)\} \odot_x B(x, r) \ \subseteq \ CAr^m [\sigma + B(y, \delta)] \,.$$

Assume first that $r < \delta$. Then as $|x - y| < \delta$, (3) gives

$$(13) \qquad B(x, r) \subseteq B(x, \delta) \subseteq CB(y, \delta) \subseteq C\,[\sigma + B(y, \delta)]\,,$$

since $0 \in \sigma$ as σ is non-empty, convex and centrally-symmetric. Combining (4) with (13) we get that,

$$\begin{aligned} &\{[\sigma + B(y, \delta)] \cap B(x, r)\} \odot_x B(x, r) \\ &\qquad \subseteq B(x, r) \odot_x B(x, r) \subseteq C' r^m B(x, r) \subseteq \tilde{C} r^m [\sigma + B(y, \delta)] \end{aligned}$$

and (12) follows for the case $r < \delta$. Suppose now that

$$r \geq \delta\,.$$

Then,

$$(14) \qquad [\sigma + B(x, \delta)] \cap B(x, r) \subseteq [\sigma \cap 2B(x, r)] + B(x, \delta).$$

Indeed, (14) follows from (11) since $B(x, \delta) \subseteq B(x, r)$. The sets $B(x, \delta)$ and $B(y, \delta)$ are C-equivalent by (3), since $|x - y| < \delta$. Therefore, (14) implies

$$(15) \qquad [\sigma + B(y, \delta)] \cap B(x, r) \subseteq C\,[\sigma \cap B(x, r)] + C' B(y, \delta).$$

We will deal separately with each summand in the right-hand side of (15). Since $r \geq \delta \geq |x-y|$, we know by (3) that $B(x,r)$ and $B(y,r)$ are C-equivalent. We use (5) and then the aforementioned equivalence to get that

$$\begin{aligned}(16)\qquad [\sigma \cap B(x,r)] \odot_x B(x,r) &\subseteq [\sigma \cap B(x,r)] \odot_y B(x,r) + C' r^m B(x,y)\\ &\subseteq C\,[\sigma \cap B(y,r)] \odot_y B(y,r) + C' r^m B(x,y)\,.\end{aligned}$$

Recall that σ is assumed to be Whitney t-convex at y with Whitney constant A. Hence, from (16) and (10) we obtain

$$(17)\ \ [\sigma \cap B(x,r)] \odot_x B(x,r) \subseteq CAr^m\sigma + Cr^m B(x,y) \subseteq CAr^m\sigma + C' r^m B(y,\delta),$$

since $B(x,y) \subseteq CB(y,\delta)$ by (3). Next, we once more use the fact that $B(x,\delta)$ and $B(y,\delta)$ are C-equivalent. Thus,

$$(18)\qquad B(y,\delta) \odot_x B(x,r) \subseteq CB(x,\delta) \odot_x B(x,r).$$

By (18) and (4), since $r \geq \delta$,

$$(19)\qquad B(y,\delta) \odot_x B(x,r) \subseteq C' r^m B(x,\delta) \subseteq \tilde{C} r^m B(y,\delta).$$

By combining (15), (17) and (19), we see that

$$\big\{[\sigma + B(y,\delta)] \cap B(x,r)\big\} \odot_x B(x,r) \subseteq C\,[Ar^m\sigma + r^m B(y,\delta)] + C' r^m B(y,\delta),$$

and thus,

$$\big\{[\sigma + B(y,\delta)] \cap B(x,r)\big\} \odot_x B(x,r) \subseteq CAr^m\,[\sigma + B(y,\delta)]\,.$$

This is precisely our desired inclusion (12). The proof is complete. ■

13. Properties of the Γ's and σ's

In this section we establish the basic mathematical properties of the blobs $\Gamma(x,\ell)$ and the convex sets $\sigma(x,\ell)$. Those properties are as follows. We write C_ℓ for a constant depending only on ℓ, m, n.

Property 0:

(a) Let $F \in C^m(\mathbb{R}^n)$ and $M > 0$ be given. Assume that

$$\| F \|_{C^m(\mathbb{R}^n)} \leq M \text{ and } |F(x) - f(x)| \leq M\sigma(x) \text{ for all } x \in E\,.$$

Then $J_x(F) \in \Gamma(x,\ell,C_\ell M)$ for all $x \in E$, $\ell \geq 0$.

(b) Let $F \in C^m(\mathbb{R}^n)$ be such that

$$\| F \|_{C^m(\mathbb{R}^n)} \leq 1 \text{ and } |F(x)| \leq \sigma(x) \text{ for all } x \in E\,.$$

Then $J_x(F) \in C_\ell \sigma(x,\ell)$ for all $x \in E$, $\ell \geq 0$.

Property 1: For any $x \in E$, $\ell \geq 0$, $M > 0$, we have

(a) $\Gamma(x,\ell,M) + M\sigma(x,\ell) \subseteq \Gamma(x,\ell,C_\ell M)$, and

(b) $\Gamma(x,\ell,M) - \Gamma(x,\ell,M) \subseteq C_\ell M\sigma(x,\ell)$.

Here, $A + B$ and $A - B$ denote the Minkowski sum and difference, i.e., $A + B = \{P + Q : P \in A, Q \in B\}$ and $A - B = \{P - Q : P \in A, Q \in B\}$.

Property 2:

(a) Let $x, y \in E$, $\ell \geq 1, M > 0$. Then,

$$\Gamma(x,\ell,M) \subset \Gamma(y,\ell-1,C_\ell M) + C_\ell M B(x,y).$$

(b) Let $x, y \in E$, $\ell \geq 1$. Then,

$$\sigma(x,\ell) \subset C_\ell\,[\sigma(y,\ell-1) + B(x,y)]\,.$$

Property 3: For each $x \in E, \ell \geq 0$, the set $\sigma(x,\ell)$ is Whitney t-convex at x, with Whitney constant C_ℓ.

Property 4: For each $x \in E, \ell \geq 0, M > 0$, we have

$$\Gamma(x,\ell,M) \subset \Gamma(x,\ell-1,C_\ell M),$$

$$\sigma(x,\ell) \subset C_\ell \sigma(x,\ell-1).$$

Property 0(a) was proven in Lemma 2 of Section 7 in [19]. Property 2(a) was also proven in [19]; see (5) and (8) of Section 7 in [19]. In order to establish Property 0(b) and Property 2(b), recall from Section 10 that $\Gamma^0(x,\ell,M) = M\sigma(x,\ell)$ and $\Gamma^0(x,\ell)$ is the blob that arises when $f \equiv 0$. Thus, Property 0(b) is a particular case of Property 0(a), and Property 2(b) is a particular case of Property 2(a). It remains to establish Properties 1, 3 and 4. Property 4 is trivial, as will be explained below. Properties 1 and 3 will follow by a straightforward induction on ℓ, making use of Lemma 3 from the preceding section for the proof of Property 3. We supply details.

Recalling our construction of the $\Gamma(x,\ell)$ and $\sigma(x,\ell)$, we see that the $\sigma(x,\ell)$ arise by the following induction on ℓ.

For $\ell = 0$, we set

$$\sigma(x,0) = \{P \in \mathcal{P} : |P(x)| \leq \sigma(x) \text{ and } |\partial^\beta P(x)| \leq 1 \ \forall |\beta| \leq m-1\}.$$

For the inductive step, fix $\ell \geq 0$, and suppose we have defined $\sigma(x,\ell)$ for each $x \in E$. We will define $\sigma(x,\ell+1)$ for each $x \in E$. To do so, we use the Callahan-Kosaraju decomposition for E, with $\varkappa = 1/2$. We retain the notation of Section 9 (except that C here denotes a constant depending only on m and n). We construct the sets $\sigma(x,\ell+1)$ in five steps:

Step 1°: For each $A \in \mathcal{T}$, we define

$$\sigma(A,\ell) = \bigcap_{x \in A} \left[\sigma(x,\ell) + B(x, \mathrm{diam}_\infty(A))\right].$$

Step 2°: For each $(\Lambda_1,\Lambda_2) \in \mathcal{L}$, and for $i = 1,2$, we define

$$\sigma_i(\Lambda_i,\ell) = \bigcap_{A \in \Lambda_i} \left[\sigma(A,\ell) + B(x_A, \mathrm{diam}_\infty(\cup\Lambda_i))\right].$$

Step 3°: For each $(\Lambda_1,\Lambda_2) \in \mathcal{L}$, we define

$$\bar{\sigma}(\Lambda_1,\Lambda_2,\ell) = \sigma_1(\Lambda_1,\ell) \cap \left[\sigma_2(\Lambda_2,\ell) + B(x_{\Lambda_1}, x_{\Lambda_2})\right].$$

Step 4°: For each $A \in \mathcal{T}$, we define

$$\sigma'(A,\ell+1) = \bigcap_{\substack{(\Lambda_1,\Lambda_2) \in \mathcal{L} \\ \Lambda_1 \ni A}} \bar{\sigma}(\Lambda_1,\Lambda_2,\ell).$$

Step 5°: For each $x \in E$, we define

$$\sigma(x,\ell+1) = \sigma(x,\ell) \cap \bigcap_{\substack{A \in \mathcal{T} \\ A \ni x}} \sigma'(A,\ell+1).$$

This completes the induction defining the convex symmetric sets $\sigma(x,\ell)$ $(x \in E, \ell \geq 0)$.

Property 4 is now obvious from inspection of Step 5°, and also of Step 5 from Section 10. Now we will give the induction on ℓ proving Properties 1 and 3.

For $\ell = 0$, we denote ad-hoc,

$$\tilde{\Gamma}(x,0,M) = \{P \in \mathcal{P} : |P(x) - f(x)| \leq M\,\sigma(x)\}$$

and

$$\tilde{\sigma}(x,0) = \{P \in \mathcal{P} : |P(x)| \leq \sigma(x)\}.$$

Then,

(1) $\quad \Gamma(x,0,M) = \tilde{\Gamma}(x,0,M) \cap MB(x,1)$ and $\sigma(x,0) = \tilde{\sigma}(x,0) \cap B(x,1)$.

Property 1 is obvious for $\ell = 0$. It is also straightforward to verify that $\tilde{\sigma}(x, 0)$ is Whitney t-convex at x with Whitney constant 1. Using (1) and Lemma 2 from the preceding section, we conclude that $\sigma(x, 0)$ is Whitney t-convex at x with Whitney constant C. Thus Property 3 holds for $\ell = 0$.

For the induction step, fix $\ell \geq 0$, and suppose Properties 1 and 3 hold for ℓ. We will prove those properties for $\ell + 1$. We begin with Property 1. It is elementary to see that for any sets $A_1, ..., A_k, B_1, ..., B_k$ in a vector space V we have

$$(*) \qquad \bigcap_{i=1}^{k} A_i \pm \bigcap_{i=1}^{k} B_i \subset \bigcap_{i=1}^{k} (A_i \pm B_i).$$

Since Property 1 holds for ℓ, inspection of $(*)$ and of Steps 1 and 1° in the definitions of Γ, σ shows that we have an obvious analogue of Property 1 for the blob $\Gamma(A, \ell)$ and the convex set $\sigma(A, \ell)$.
Similarly, inspection of $(*)$ and Steps 2 and 2° shows that the obvious analogue of Property 1 holds for $\Gamma_i(\Lambda_i, \ell)$ and $\sigma_i(\Lambda_i, \ell)$.
Then, inspecting Steps 3 and 3°, followed by inspecting Steps 4 and 4° and finally Steps 5 and 5°, yields that the obvious analogues of Property 1 hold for $\bar{\Gamma}(\Lambda_1, \Lambda_2, \ell)$ and $\bar{\sigma}(\Lambda_1, \Lambda_2, \ell)$, and then for $\Gamma'(A, \ell + 1)$ and $\sigma'(A, \ell + 1)$, and finally for the blob $\Gamma(x, \ell + 1)$ and the convex set $\sigma(x, \ell + 1)$.
This completes the inductive step, and establishes Property 1.
For the inductive step in the proof of Property 3, we bring in Lemma 3 from the preceding section. (We call it "the t-convexity lemma" here.) Since $\sigma(x, \ell)$ is Whitney t-convex at x with Whitney constant C_ℓ, the t-convexity lemma shows that $\sigma(x, \ell) + B(x, \mathsf{diam}_\infty(A))$ is Whitney t-convex at any $y \in A$, with Whitney constant C_ℓ, whenever $x \in A$.

Taking the intersection over all $x \in A$ and comparing with Step 1°, we see that $\sigma(A, \ell)$ is Whitney t-convex at each $y \in A$, with Whitney constant C_ℓ.

So, another application of the t-convexity lemma shows that $\sigma(A, \ell) + B(x_A, \mathsf{diam}_\infty(\cup\Lambda_i))$ is Whitney t-convex at any $y \in \cup\Lambda_i$, with Whitney constant C_ℓ, whenever $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, $i = 1$ or 2, and $A \in \Lambda_i$. Taking the intersection over all $A \in \Lambda_i$, and comparing with Step 2°, we see that

(2) $\sigma_i(\Lambda_i, \ell)$ is Whitney t-convex at each $y \in \cup\Lambda_i$, with Whitney constant C_ℓ.

Again, using the t-convexity lemma, as well as the fact that $\cup\Lambda_1$ and $\cup\Lambda_2$ are $\varkappa$-separated with $\varkappa = 1/2$, we learn from (2) that

(3) $\sigma_2(\Lambda_2, \ell) + B(x_{\Lambda_2}, x_{\Lambda_1})$ is Whitney t-convex at each point of $\cup\Lambda_1$, with Whitney constant C_ℓ, whenever $(\Lambda_1, \Lambda_2) \in \mathcal{L}$.

Since $\sigma_2(\Lambda_2, \ell) + B(x_{\Lambda_2}, x_{\Lambda_1})$ and $\sigma_2(\Lambda_2, \ell) + B(x_{\Lambda_1}, x_{\Lambda_2})$ are C-equivalent with C depending only on m and n, it follows from (3) that

(4) $\sigma_2(\Lambda_2, \ell) + B(x_{\Lambda_1}, x_{\Lambda_2})$ is Whitney t-convex at each point of $\cup\Lambda_1$, with Whitney constant C_ℓ, whenever $(\Lambda_1, \Lambda_2) \in \mathcal{L}$.

(See the observation immediately after Lemma 2 in the preceding section.) Comparing (2) and (4) with Step 3°, we see that

(5) $\bar{\sigma}(\Lambda_1, \Lambda_2, \ell)$ is Whitney t-convex at each point of $\cup\Lambda_1$, with Whitney constant C_ℓ, whenever $(\Lambda_1, \Lambda_2) \in \mathcal{L}$.

Thanks to (5) and our induction hypothesis (Whitney t-convexity of $\sigma(x, \ell)$), we learn by inspection of Steps 4° and 5° that $\sigma(x, \ell+1)$ is Whitney t-convex at x, with Whitney constant $C_{\ell+1}$, for each $x \in E$.

This completes the inductive step in the proof of Property 3.

We have established Properties 0, ..., 4.

14. On Sets of Multi-indices

We introduce and recall some notation, to be used for the rest of this paper. For multi-indices α, β, we denote by $\delta_{\alpha\beta}$ the Kronecker delta, equal to 1 if $\alpha = \beta$, and 0 otherwise. We write $\mathcal{M}$ for the set of all multi-indices $\alpha = (\alpha_1, \dots, \alpha_n)$ of order $|\alpha| = \alpha_1 + \cdots + \alpha_n \leq m-1$. We define an order relation on $\mathcal{M}$, as follows.

Let $\alpha = (\alpha_1, \dots, \alpha_n)$ and $\beta = (\beta_1, \dots, \beta_n)$ be distinct elements of $\mathcal{M}$. Then we cannot have $\alpha_1 + \cdots + \alpha_k = \beta_1 + \cdots + \beta_k$ for all $k = 1, \dots, n$. Let $\bar{k}$ be the largest k for which $\alpha_1 + \cdots + \alpha_k \neq \beta_1 + \cdots + \beta_k$. Then we say that $\alpha < \beta$ if and only if $\alpha_1 + \cdots + \alpha_{\bar{k}} < \beta_1 + \cdots + \beta_{\bar{k}}$. It is easy to check that $<$ is a linear order relation.

We also define an order relation between subsets of $\mathcal{M}$. Let $\mathcal{A}, \mathcal{B}$ be two distinct subsets of $\mathcal{M}$, and let α denote the least element of the symmetric difference $(\mathcal{A} \smallsetminus \mathcal{B}) \cup (\mathcal{B} \smallsetminus \mathcal{A})$, with respect to our order relation on $\mathcal{M}$. Then we say that $\mathcal{A} < \mathcal{B}$ if and only if α belongs to $\mathcal{A}$. Again, one checks easily that this defines a linear order relation. Note that $\mathcal{M}$ is minimal, and the empty set $\emptyset$ is maximal under this order. Note also that $\mathcal{A} \subset \mathcal{B}$ implies $\mathcal{B} \leq \mathcal{A}$.

For $\mathcal{A} \subseteq \mathcal{M}$, we define

$$\ell(\mathcal{A}) = 1 + 4 \cdot \#\{\mathcal{A}' \in \mathcal{M} : \mathcal{A}' < \mathcal{A}\},$$

where $\#S$ denotes the number of elements in the set S. Also, we set

$$\ell_* = \ell(\emptyset) + 1.$$

Thus, $1 \leq \ell(\mathcal{A}) < \ell_*$ for each $\mathcal{A} \subseteq \mathcal{M}$. Clearly, ℓ_* is a constant depending only on m and n.

We will make use of the following elementary result from [13].

Lemma 1. *Let $\mathcal{A} \subset \mathcal{M}$ and let $\phi : \mathcal{A} \to \mathcal{M}$ be a map with the following properties:*

(1) $\phi(\alpha) \leq \alpha$ *for all* $\alpha \in \mathcal{A}$*; and*

(2) *For each* $\alpha \in \mathcal{A}$*, either* $\phi(\alpha) = \alpha$ *or* $\phi(\alpha) \notin \mathcal{A}$*.*

Then $\phi(\mathcal{A}) \leq \mathcal{A}$*, with equality if and only if* ϕ *is the identity map.*

15. Finding Neighbors

In this section, we write c, C, C', etc. to denote constants depending only on m and n. Unfortunately, in this section $\mathcal{A}$ will denote a subset of $\mathcal{M}$, while $\mathcal{A}(x, \ell)$ and $\mathcal{A}^{\#}$ will denote ALPs in $\mathcal{P}$.

Suppose we are given $P_0 \in \mathcal{P}$, $\mathcal{A} \subseteq \mathcal{M}$, $x \in E$. We want to compute a polynomial $P \in \mathcal{P}$ with the following property.

(I) Let $P' \in \mathcal{P}$ and $M > 0$ be given. Suppose P' and M satisfy:

(a)′ $\partial^\beta(P' - P_0)(x) = 0$ for all $\beta \in \mathcal{A}$; and

(b)′ $P' \in \Gamma(x, \ell(\mathcal{A}) - 1, M)$.

Then P and M satisfy:

(a) $\partial^\beta(P - P_0)(x) = 0$ for all $\beta \in \mathcal{A}$; and

(b) $P \in \Gamma(x, \ell(\mathcal{A}) - 1, CM)$.

(Recall that C depends only on m and n.)

We will give an algorithm, called Find-Neighbor$(P_0, \mathcal{A}, x)$, that returns such a polynomial P, with work and storage at most C'. We assume that we have already performed the one-time work of finding an ALP $\mathcal{A}(x, \ell(\mathcal{A}) - 1)$ in $\mathcal{P}$, such that

(1) The blobs $\Gamma(x, \ell(\mathcal{A}) - 1)$ and $\mathcal{K}(\mathcal{A}(x, \ell(\mathcal{A}) - 1))$ are C-equivalent.

See Section 10 for the computation of the $\mathcal{A}(x, \ell)$. Recall in particular from that section, that length $(\mathcal{A}(x, \ell(\mathcal{A}) - 1)) \leq D + 1$, where $D = \dim \mathcal{P}$. Thanks to (1), we can replace $\Gamma(x, \ell(\mathcal{A}) - 1, M)$ and $\Gamma(x, \ell(\mathcal{A}) - 1, CM)$ in (I) by $K_M(\mathcal{A}(x, \ell(\mathcal{A}) - 1))$ and $K_{CM}(\mathcal{A}(x, \ell(\mathcal{A}) - 1))$, respectively, without affecting the validity of (I). In dealing with (I), we will assume this substitution has been made.

We have already computed $\mathcal{A}(x, \ell(\mathcal{A}) - 1)$. Thus,

(2) $\mathcal{A}(x, \ell(\mathcal{A}) - 1) = [(\lambda_1, \ldots, \lambda_L), (b_1, \ldots, b_L), (\sigma_1, \ldots, \sigma_L), M_*]$,

with known $\lambda_\ell, b_\ell, \sigma_\ell$, M_*, and with $L \leq D + 1$.

Here, the λ_ℓ are (real) linear functionals on $\mathcal{P}$. Recall that $P' \in K_M (\mathcal{A}(x, \ell(\mathcal{A}) - 1))$ is equivalent to the assertions

$$|\lambda_\ell(P') - b_\ell| \le M\sigma_\ell \qquad \text{for } \ell = 1, \dots, L\text{; and } M \ge M_*.$$

The condition $P \in K_{CM}(\mathcal{A}(x, \ell(\mathcal{A}) - 1))$ may be similarly expressed in terms of the $\lambda_\ell, b_\ell, \sigma_\ell, M_*$. Hence, our desired property (I) is equivalent to the following:

(II) Let $P' \in \mathcal{P}$ and $M > 0$ be given. Suppose P' and M satisfy:

(3) $\partial^\beta(P' - P_0)(x) = 0$ for $\beta \in \mathcal{A}$;

(4) $|\lambda_\ell(P') - b_\ell| \le M\sigma_\ell$ for $\ell = 1, \dots, L$; and

(5) $M \ge M_*$.

Then P and M satisfy:

(3′) $\partial^\beta(P - P_0)(x) = 0$ for $\beta \in \mathcal{A}$;

(4′) $|\lambda_\ell(P) - b_\ell| \le CM\sigma_\ell$ for $\ell = 1, \dots, L$; and

(5′) $CM \ge M_*$.

In view of $(3), \dots, (5)$, it is natural to define an auxiliary blob $\mathcal{K}^\# = (K^\#_M)_{M>0}$ in $\mathcal{P}$, by setting

(6) $K^\#_M = \{P' \in \mathcal{P}$: Conditions (3) and (4) hold $\}$ for $M \ge M_*$, and

(7) $K^\#_M = \emptyset$ for $M < M_*$.

We can then rewrite condition (II) in terms of the blob $\mathcal{K}^\#$. In fact, (II) asserts that, whenever $P' \in K^\#_M$, we must have $P \in K^\#_{CM}$. That is, (II) asserts that P is a C-original vector for the blob $\mathcal{K}^\#$.

On the other hand, a glance at (3), (4) and (6), (7) shows that $\mathcal{K}^\#$ has the form $\mathcal{K}(\mathcal{A}^\#)$ for an obvious ALP $\mathcal{A}^\#$.

The ALP $\mathcal{A}^\#$ may be easily read off from known data ($P_0, \mathcal{A}, x$, and the $\lambda_\ell, b_\ell, \sigma_\ell, M_*$). Note also that length $(\mathcal{A}^\#) = \#(\mathcal{A}) + L \le C$.

Thus, the task of finding $P \in \mathcal{P}$ satisfying (I) amounts to finding a C-original vector for the blob $\mathcal{K}(\mathcal{A}^\#)$ arising from a known ALP $\mathcal{A}^\#$ of length $\le C$. This task may be performed with work and storage at most C', by using Algorithm ALP3 from Section 5.

This completes our discussion of the algorithm Find-Neighbor $(P_0, \mathcal{A}, x)$.

16. Neighbors Depending Linearly on Parameters

The polynomial P returned by the algorithm Find-Neighbor $(P_0, \mathcal{A}, x)$ from the preceding section depends not only on $P_0, \mathcal{A}, x$, but also on E, σ, and f. In this section, we investigate the linear dependence of P on P_0 and f, with the remaining inputs held fixed.

Again, we write c, C, C', etc. to denote constants depending only on m and n.

We suppose we are given a map $\vec{f} : E \times \mathbb{R}^{\bar{N}} \to \mathbb{R}$, such that $\xi \mapsto \vec{f}(x, \xi)$ is a depth k linear functional on $\mathbb{R}^{\bar{N}}$, for each fixed $x \in E$. For fixed $\xi \in \mathbb{R}^{\bar{N}}$, we write f_ξ to denote the function $x \mapsto \vec{f}(x, \xi)$ on E. We suppose that $\bar{N}$ satisfies (2) from Section 6.

It is convenient to introduce the following definition. Let $\vec{P} : \mathbb{R}^{\bar{N}} \to \mathcal{P}$ be a linear map. We say that $\vec{P}$ has "depth k" if for each linear functional $\lambda : \mathcal{P} \to \mathbb{R}$, the functional $\lambda \circ \vec{P}$ on $\mathbb{R}^{\bar{N}}$ has depth k. We call such a $\vec{P}$ a "depth k parametrized polynomial". Note that a depth k parametrized polynomial takes up storage at most $C(k+1)$.

Recall from Section 11 that we have constructed PALPs $\underline{\mathcal{A}}(x, \ell)$ for $x \in E$, $0 \leq \ell \leq \ell_*$. The PALPs $\underline{\mathcal{A}}(x, \ell)$ depend on $\vec{f}$. We assume here that the PALPs $\underline{\mathcal{A}}(x, \ell)$ have already been computed.

Our goal here is to exhibit an algorithm

(0) Find-Parametrized-Neighbor $(\vec{P}_0, \mathcal{A}, x)$ with the following properties.

(1) The inputs of algorithm (0) are a depth-k parametrized polynomial $\vec{P}_0$, a subset $\mathcal{A} \subseteq \mathcal{M}$ and a point $x \in E$.

(2) The output of algorithm (0) is a depth-Ck parametrized polynomial $\vec{P}$.

(3) Let $\vec{P}$ be the parametrized polynomial returned by algorithm (0) for inputs $\vec{P}_0$, $\mathcal{A}, x$. Let $\xi \in \mathbb{R}^{\bar{N}}$ be given. Set $P_0 = \vec{P}_0(\xi)$, $P = \vec{P}(\xi)$, and $f = f_\xi$. Then P is the polynomial returned by the algorithm Find-Neighbor $(P_0, \mathcal{A}, x)$ with initial data E, σ, f.

(4) The algorithm (0) uses the PALP $\underline{\mathcal{A}}(x, \ell(\mathcal{A}) - 1)$ arising from $\vec{f}$. Once $\underline{\mathcal{A}}(x, \ell(\mathcal{A}) - 1)$ is known, the algorithm (0) uses work and storage at most $C(k+1)$.

Thanks to (3), the algorithm Find-Parametrized-Neighbor captures the ξ-dependence of the output of Find-Neighbor, when the inputs P_0 and f depend linearly (with depth k) on a parameter $\xi \in \mathbb{R}^{\bar{N}}$.

To exhibit the algorithm (0), we just carry over our previous algorithm Find-Neighbor into the setting of PALPs. Where we used the ALP $\mathcal{A}(x, \ell(\mathcal{A}) - 1)$ in the preceding section, we now use the PALP $\underline{\mathcal{A}}(x, \ell(\mathcal{A}) - 1)$.

For $\xi \in \mathbb{R}^{\bar{N}}$, consider the ALP $\mathcal{A}(x, \ell(\mathcal{A}) - 1)$ that is constructed from E, σ, f when $f = f_\xi$. Recall from Section 11 that this ALP is denoted by $\mathcal{A}_\xi(x, \ell(\mathcal{A}) - 1)$. Recall from $(2)_\ell$ of Section 11 that $\underline{\mathcal{A}}(x, \ell(\mathcal{A}) - 1)$ agrees with $\mathcal{A}_\xi(x, \ell(\mathcal{A}) - 1)$ at ξ, for any $\xi \in \mathbb{R}^n$.

Whereas $\mathcal{A}(x, \ell(\mathcal{A}) - 1) = [(\lambda_1, \dots, \lambda_L), (b_1, \dots, b_L), (\sigma_1, \dots, \sigma_L), M_*]$, we now have

$$\underline{\mathcal{A}}(x, \ell(\mathcal{A}) - 1) = [(\underline{\lambda}_1, \dots, \underline{\lambda}_L), (\underline{b}_1, \dots, \underline{b}_L), (\underline{\sigma}_1, \dots, \underline{\sigma}_L)] .$$

The λ_ℓ and $\underline{\lambda}_\ell$ are linear functionals on $\mathcal{P}$, and the σ_ℓ and $\underline{\sigma}_\ell$ are non-negative numbers. The b_ℓ are real numbers, whereas the $\underline{b}_\ell$ are depth-Ck linear functionals on $\mathbb{R}^{\bar{N}}$. Note that $\underline{\mathcal{A}}(x, \ell(\mathcal{A}) - 1)$ contains no "threshold" analogous to M_*

Where we used the $\lambda_\ell, b_\ell, \sigma_\ell, M_*$ to define the ALP $\mathcal{A}^\#$ in the preceding section, we now use the $\underline{\lambda}_\ell, \underline{b}_\ell, \underline{\sigma}_\ell$ to define an analogous PALP $\underline{\mathcal{A}}^\#$, whose detailed construction we leave to the reader.

Where we applied Algorithm ALP3 to the ALP $\mathcal{A}^\#$ in the preceding section, we now apply Algorithm PALP3 to obtain the depth $C'k$ parametrized polynomial $\vec{P}$. (See Section 6.)

Thus, we have exhibited the algorithm (0). The verification of properties (1), . . ., (4) is routine. In particular, property (3) follows easily, once we recall the defining property of Algorithm PALP3, together with $(2)_\ell$ from Section 11.

Chapter III - Lengthscales and Calderón-Zygmund decompositions

17. Picking Constants

In the sequel, we will make use of two large constants, $A_0, p_\# > 1$, that depend solely on m and n. The constant $p_\#$ will be an integer, and we will choose A_0 to be an integer power of two.

These constants, A_0 and $p_\#$, will not be specified right now; instead, throughout the manuscript, we will stipulate some lower bounds on $A_0, p_\#$. These lower bounds are always by constants depending only on m and n. These bounds will appear later in the text, specifically in (23), (31) and (41) in Section 31, in (9), (15), (25) and (34) in Section 32 and in (9), (14), (19), (25), (71) and (81) from Section 33. (In the Appendix, we specify additional, similar lower bounds for A_0 and $p_\#$.) For concreteness, we set A_0 and $p_\#$ to be the minimal integer powers of two that satisfy the aforementioned lower bounds.

For $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$ we write $\mathcal{A}^+$ to denote the successor of $\mathcal{A}$ in our order relation on sets of multi-indices, that was defined in Section 14. Similarly, for $\mathcal{A} \subsetneq \mathcal{M}$, we write $\mathcal{A}^-$ to denote the predecessor of $\mathcal{A}$ in our order relation. (Recall that $\emptyset$ is maximal and $\mathcal{M}$ is minimal with respect to our order.) Next, we will define for each $\mathcal{A} \subseteq \mathcal{M}$ a constant $A_1(\mathcal{A})$ as follows:

(1) $\quad A_1(\emptyset) = A_0^{p\#}, \quad$ and $\; A_1(\mathcal{A}) = \left(A_0^2 A_1(\mathcal{A}^+)\right)^{p\#}$ for $\mathcal{A} \subseteq \mathcal{M}, \mathcal{A} \neq \emptyset$.

Once the constants $A_1(\mathcal{A})$ are defined for all $\mathcal{A} \subseteq \mathcal{M}$, we set

(2) $\quad A_2 = (A_0 A_1(\mathcal{M}))^{p\#}$.

Note that $A_0, A_1(\mathcal{A})$ ($\mathcal{A} \subseteq \mathcal{M}$) and A_2 are all integer powers of two.

Finally, we will define constants $A_3(\mathcal{A})$, for $\mathcal{A} \subseteq \mathcal{M}$, as follows:

(3) $\quad A_3(\mathcal{M}) = A_0^2 A_1(\mathcal{M}), \quad$ and $A_3(\mathcal{A}) = A_0 A_2^m A_3(\mathcal{A}^-)$ for $\mathcal{A} \subset \mathcal{M}, \mathcal{A} \neq \mathcal{M}$.

The constants $A_3(\mathcal{A})$ will be used only much later, in Section 29 and Section 33. This finishes the specifications of all constants we need in the sequel.

18. The Basic Lengthscales

Fix $\mathcal{A} \subseteq \mathcal{M}$, and let $A_1(\mathcal{A})$ be the constant from the preceding section. We recall the integer $\ell(\mathcal{A}) = 1 + 4 \cdot \#\{\mathcal{A}' \subseteq \mathcal{M} : \mathcal{A}' < \mathcal{A}\}$, satisfying $1 \leq \ell(\mathcal{A}) < \ell_*$. Also, from Section 10, we recall the convex symmetric set $\sigma(x, \ell) \subseteq \mathcal{P}$ and the homogeneous ALPs $\mathcal{A}^0(x, \ell)$ in $\mathcal{P}$, defined for $x \in E$, $\ell \geq 0$. From (11) in that section, we have

(1) $\quad c_\ell \sigma(\mathcal{A}^0(x, \ell)) \subseteq \sigma(x, \ell) \subseteq C_\ell \sigma(\mathcal{A}^0(x, \ell))$ for all $x \in E$, $\ell \geq 0$,

with c_ℓ, C_ℓ depending only on ℓ, m, n. Recall that $\mathcal{A}^0(x, \ell)$ has the following form, since it is a homogeneous ALP:

(2) $\quad \mathcal{A}^0(x, \ell) = [(\lambda_1, \dots, \lambda_L), (0, \dots, 0), (\sigma_1, \dots, \sigma_L), 0]$.

Here, each λ_i is a linear functional on $\mathcal{P}$, and each σ_i is a non-negative real number. The λ_i and σ_i, and the length L, may all depend on x and ℓ. By definition, we have

(3) $\quad \sigma(\mathcal{A}^0(x, \ell)) = \{P \in \mathcal{P} : |\lambda_i(P)| \leq \sigma_i$ for $i = 1, \dots, L\}$

with $\mathcal{A}^0(x, \ell)$ given by (2). Recall that $L =$ length $\mathcal{A}^0(x, \ell) \leq D + 1$, where $D = \dim \mathcal{P}$.

In Section 10, we saw that the homogeneous ALPs $\mathcal{A}^0(x,\ell)$ (for all $x \in E$, $0 \le \ell \le \ell_*$) can be computed with work $CN \log N$ and storage CN, with C depending only on m, n. We assume here that these homogeneous ALPs have been computed as part of the one-time work.

The goal of this section is to compute, for each $x \in E$, a lengthscale $\delta(x,\mathcal{A}) \in [0,\infty]$, with the following properties:

(OK1) Let δ be given, with $0 < \delta < \delta(x,\mathcal{A})$. Then there exist $P_\alpha \in \mathcal{P}$, indexed by $\alpha \in \mathcal{A}$, such that:

(a) $\partial^\beta P_\alpha(x) = \delta_{\beta\alpha}$ for $\beta, \alpha \in \mathcal{A}$;

(b) $|\partial^\beta P_\alpha(x)| \le CA_1(\mathcal{A})\delta^{|\alpha|-|\beta|}$ for $\alpha \in \mathcal{A}, \beta \in \mathcal{M}$, $\beta \ge \alpha$; and

(c) $\delta^{m-|\alpha|}P_\alpha \in CA_1(\mathcal{A}) \cdot \sigma(x,\ell(\mathcal{A}))$ for $\alpha \in \mathcal{A}$.

(OK2) Let $\delta > 0$ be given, and suppose there exist $P_\alpha \in \mathcal{P}$, indexed by $\alpha \in \mathcal{A}$, such that:

(a) $\partial^\beta P_\alpha(x) = \delta_{\beta\alpha}$ for $\beta, \alpha \in \mathcal{A}$;

(b) $|\partial^\beta P_\alpha(x)| \le cA_1(\mathcal{A})\delta^{|\alpha|-|\beta|}$ for $\alpha \in \mathcal{A}, \beta \in \mathcal{M}$, $\beta \ge \alpha$; and

(c) $\delta^{m-|\alpha|}P_\alpha \in cA_1(\mathcal{A}) \cdot \sigma(x,\ell(\mathcal{A}))$ for $\alpha \in \mathcal{A}$.

Then $0 < \delta < \delta(x,\mathcal{A})$.

In (OK1), (OK2), and for the rest of this section, c and C denote constants depending only on m and n.

We will compute all the $\delta(x,\mathcal{A})$ ($x \in E, \mathcal{A} \subseteq \mathcal{M}$) with work and storage at most CN (once we know the $\mathcal{A}^0(x,\ell(\mathcal{A}))$). We begin by applying (1), (2), (3) with $\ell = \ell(\mathcal{A})$. Since $0 \le \ell(\mathcal{A}) \le \ell_*$, the constants c_ℓ and C_ℓ in (1) may be taken to depend only on m and n, once we set $\ell = \ell(\mathcal{A})$. Hence (1) shows that $\sigma(x,\ell(\mathcal{A}))$ may be replaced by $\sigma(\mathcal{A}^0(x,\ell(\mathcal{A})))$ in (OK1) and (OK2) without affecting their validity. Applying (3), with $\ell = \ell(\mathcal{A})$, we see that our desired properties (OK1) and (OK2) are equivalent to the following.

(OK1′) Let δ be given, with $0 < \delta < \delta(x,\mathcal{A})$. Then there exist $P_\alpha \in \mathcal{P}$, indexed by $\alpha \in \mathcal{A}$, such that

$$\left[\begin{array}{l} |\partial^\beta P_\alpha(x) - \delta_{\beta\alpha}| \le 0 \text{ for } \beta, \alpha \in \mathcal{A}\,; \\[2ex] |\partial^\beta P_\alpha(x)| \le CA_1(\mathcal{A})\ \delta^{|\alpha|-|\beta|} \text{ for } \alpha \in \mathcal{A}, \beta \in \mathcal{M},\ \beta \ge \alpha\,; \\[2ex] |\lambda_\ell(P_\alpha)| \le C\,A_1(\mathcal{A})\sigma_\ell \cdot \delta^{|\alpha|-m} \text{ for } \alpha \in \mathcal{A},\ 1 \le \ell \le L\,. \end{array}\right]$$

(OK2′) Let $\delta > 0$ be given, and suppose there exist $P_\alpha \in \mathcal{P}$, indexed by $\alpha \in \mathcal{A}$, such that

$$\left[\begin{array}{l} |\partial^\beta P_\alpha(x) - \delta_{\beta\alpha}| \leq 0 \text{ for } \beta, \alpha \in \mathcal{A}\,; \\ \\ |\partial^\beta P_\alpha(x)| \leq cA_1(\mathcal{A})\delta^{|\alpha|-|\beta|} \text{ for } \alpha \in \mathcal{A}, \beta \in \mathcal{M}, \beta \geq \alpha\,; \\ \\ |\lambda_\ell(P_\alpha)| \leq cA_1(\mathcal{A})\sigma_\ell \cdot \delta^{|\alpha|-m} \text{ for } \alpha \in \mathcal{A}\,, 1 \leq \ell \leq L\,. \end{array}\right]$$

Then $0 < \delta < \delta(x, \mathcal{A})$.

We already know the λ_ℓ and σ_ℓ, and we have $L \leq D + 1$. Also, on the right-hand sides of (OK1′) and (OK2′), the powers of δ all involve exponents between $-m$ and 0. Consequently, the task of finding $\delta(x, \mathcal{A})$ with properties (OK1′) and (OK2′) is a special case of the problem solved in Section 8. Applying Lemma 1 from that section, we see that a single $\delta(x, \mathcal{A})$ with properties (OK1′) and (OK2′) may be computed using work and storage at most C. Hence, we obtain the following result.

Lemma 1. *Let $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$, and $A_1(\mathcal{A})$ be given. Assuming we already know the homogeneous* ALP $\mathcal{A}^0(x, \ell(\mathcal{A}))$*, we can compute a number $\delta(x, \mathcal{A}) \in [0, \infty]$, satisfying* (OK1) *and* (OK2)*. The computation takes work and storage less than a constant C depending only on m and n.*

The lengthscales $\delta(x, \mathcal{A})$ provided by Lemma 1 belong to $[0, \infty]$, and conclusion (OK1) applies to $0 < \delta < \delta(x, \mathcal{A})$. However, if $\delta(x, \mathcal{A}) \in (0, \infty)$, then (OK1) applies also to $\delta = \delta(x, \mathcal{A})$. To see this, we simply apply (OK1) for $\delta = \frac{1}{2}\delta(x, \mathcal{A})$.

From now on, we assume that the lengthscales $\delta(x, \mathcal{A})$ have already been computed and stored, as part of the one-time work. The total work and storage required for the computation are at most CN, given that we have precomputed the ALPs $\mathcal{A}^0(x, \ell(\mathcal{A}))$.

19. Dyadic Cubes: Notation

A “dyadic cube” in $\mathbb{R}^n$ is a Cartesian product $Q = I_1 \times \cdots \times I_n \subset \mathbb{R}^n$, where each I_j has the form $[2^s t, 2^s(t+1))$ for integers s, t, and where the I_j all have the same length.

To “bisect” a dyadic cube $Q \subset \mathbb{R}^n$ is to partition it into 2^n congruent subcubes in the obvious way. For each dyadic cube Q, there is a unique dyadic cube Q^+ such that Q is among the cubes obtained by bisecting Q^+. We call Q^+ the dyadic “parent” of Q.

Let $Q = I_1 \times \cdots \times I_n \subset \mathbb{R}^n$ be a dyadic cube, with each I_j written in the form $[a_j - \frac{h}{2}, a_j + \frac{h}{2})$. Then we write $\delta_Q =$ sidelength (Q) for h; and, for $r \geq 1$, we write rQ to denote

$$\prod_{j=1}^{n} \left[a_j - r\frac{h}{2}, \quad a_j + r\frac{h}{2} \right) \subset \mathbb{R}^n,$$

the enlargement of Q by factor r. There is a slight inconsistency problem with this notation; recall that for $r > 0$ and $\Omega \subset \mathcal{P}$ we have used the notation $r\Omega = \{rP : P \in \Omega\}$. As a rule, whenever writing $r\Omega$ we mean $\{rP : P \in \Omega\}$, unless $\Omega = Q$ is a dyadic cube. When Q is a dyadic cube, rQ stands for the enlargement of Q by factor r, as defined above.

We write Q^* to denote $5Q$, Q^{**} to denote $25Q$ and Q^{***} to denote $125Q$.

There is a constant $c_G > 0$ (say, $c_G = 1/32$), with the following property:

(0) Let Q, Q' be dyadic cubes, with $(1 + 2c_G)Q \cap (1 + 2c_G)\,Q' \neq \emptyset$.

Then:

(a) If $\delta_{Q'} \leq \frac{1}{2}\delta_Q$, then $(Q')^* \subset Q^*$;

(b) If $\delta_{Q'} \leq 2\delta_Q$, then $Q' \subset Q^*$; and

(c) If $\frac{1}{2}\delta_Q \leq \delta_{Q'} \leq 2\delta_Q$, then the closures of Q and Q' contain a point in common.

For the rest of this paper, c_G denotes the above constant even if c, C, C', etc. denote constants that may change from one occurrence to the next. We take c_G to be a power of 2.

20. Calderón-Zygmund Cubes: Definitions

Recall that for $x \in E$ and $\mathcal{A} \subseteq \mathcal{M}$, we have defined a "lengthscale" $\delta(x, \mathcal{A}) \in [0, \infty]$. Here, we will use the $\delta(x, \mathcal{A})$, and the constant A_2, to partition $\mathbb{R}^n$ into "Calderón-Zygmund cubes". The constant A_2 was defined in Section 17. The only property of A_2 that will be used in this section is that

(0) $A_2 \geq 1$ is a power of 2.

We write $\mathcal{D}(A_2)$ to denote the collection of all dyadic cubes Q such that $\delta_Q \leq A_2^{-1}$. Note that, if $Q \in \mathcal{D}(A_2)$, then either $Q^+ \in \mathcal{D}(A_2)$ or $\delta_Q = A_2^{-1}$.

In the next few sections, except for Lemma 5 in Section 21 below, we make no use of the properties of the $\delta(x, \mathcal{A})$. We may regard them simply as arbitrary given numbers in $[0, \infty]$.

Let $Q \in \mathcal{D}(A_2)$ (i.e., let Q be a dyadic cube of sidelength $\le A_2^{-1}$), and let $\mathcal{A} \subseteq \mathcal{M}$ be given.
We say that Q is "$\mathsf{OK}(\mathcal{A})$" if we have

(1) $\quad A_2\delta_Q \le \delta(x, \mathcal{A})$ for all $x \in Q^* \cap E$.

Also, for $Q \in \mathcal{D}(A_2)$ and $\mathcal{A} \subseteq \mathcal{M}$, we say that Q is "almost $\mathsf{OK}(\mathcal{A})$" if we have either

(2) $\quad \#(E \cap Q^*) \le 1$, or

(3) $\quad Q$ is $\mathsf{OK}(\mathcal{A}')$ for some $\mathcal{A}' \le \mathcal{A}$.

Note that Q can be $\mathsf{OK}(\mathcal{A})$ or almost $\mathsf{OK}(\mathcal{A})$, only if $\delta_Q \le A_2^{-1}$. Also, note that every almost $\mathsf{OK}(\mathcal{A})$ cube Q is contained in some maximal almost $\mathsf{OK}(\mathcal{A})$ cube Q'. (This follows from the observation that there are only finitely many $Q' \in \mathcal{D}(A_2)$ containing Q.)

Finally, for $Q \in \mathcal{D}(A_2)$ and $\mathcal{A} \subseteq \mathcal{M}$, we say that $Q \in \mathsf{CZ}(\mathcal{A})$ (Q is a "Calderón-Zygmund" or "CZ" cube) if the following hold:

(4) $\quad Q$ is almost $\mathsf{OK}(\mathcal{A})$; but

(5) $\quad$ No cube $Q' \in \mathcal{D}(A_2)$ that properly contains Q is almost $\mathsf{OK}(\mathcal{A})$.

Again note that Q cannot be a CZ cube unless $\delta_Q \le A_2^{-1}$.

21. Calderón-Zygmund Cubes: Basic Properties

In this section, we give the basic properties of the CZ cubes defined in the preceding section. We write c, C, C', etc., to denote constants depending only on the dimension n. We recall the constant c_G from Section 19.

Lemma 1. *For each $\mathcal{A} \subseteq \mathcal{M}$, the collection $\mathsf{CZ}(\mathcal{A})$ forms a locally finite partition of $\mathbb{R}^n$ into dyadic cubes.*

Proof. Fix $\mathcal{A} \subseteq \mathcal{M}$, and recall that any two dyadic cubes Q, Q' satisfy either $Q \subseteq Q'$, $Q' \subseteq Q$, or $Q \cap Q' = \emptyset$. Immediately from the definition, we see that one cube from $\mathsf{CZ}(\mathcal{A})$ cannot be properly contained in another cube from $\mathsf{CZ}(\mathcal{A})$. Consequently, the cubes in $\mathsf{CZ}(\mathcal{A})$ are pairwise disjoint.

On the other hand, any sufficiently small dyadic cube $Q \subset \mathbb{R}^n$ is almost $\mathsf{OK}(\mathcal{A})$, since $Q \in \mathcal{D}(A_2)$ and $\#(E \cap Q^*) \le 1$. Such a cube Q must therefore be contained in a maximal almost $\mathsf{OK}(\mathcal{A})$-cube Q'. The cube Q' belongs to $\mathsf{CZ}(\mathcal{A})$. It follows that $\mathsf{CZ}(\mathcal{A})$ forms a covering of $\mathbb{R}^n$, and that the sidelengths of the cubes in $\mathsf{CZ}(\mathcal{A})$ are bounded from below. The proof of the lemma is complete. ■

Lemma 2 ("Good Geometry"). *If* $Q, Q' \in CZ(\mathcal{A})$*, with* $(1+2c_G)Q \cap (1+2c_G)Q' \neq \emptyset$*, then* $\frac{1}{2}\delta_Q \leq \delta_{Q'} \leq 2\delta_Q$*.*

Proof. Suppose not. Without loss of generality, we may suppose $\delta_Q \leq \delta_{Q'}$. Since δ_Q and $\delta_{Q'}$ are powers of 2, it follows that

(1) $\quad \delta_Q \leq \frac{1}{4}\delta_{Q'}$.

Since $Q' \in CZ(\mathcal{A}) \subseteq \mathcal{D}(A_2)$, it follows from (1) that

(2) $\quad Q^+ \in \mathcal{D}(A_2)$, and

(3) $\quad \delta_{Q^+} \leq \frac{1}{2}\delta_{Q'}$.

We have also

$$(1+2c_G)Q^+ \cap (1+2c_G)Q' \supseteq (1+2c_G)Q \cap (1+2c_G)Q' \neq \emptyset.$$

Together with (3), this yields

(4) $\quad (Q^+)^* \subset (Q')^*$,

thanks to the defining property of c_G (see (0)(a) in Section 19). Now, Q' is almost $OK(\mathcal{A})$, since it belongs to $CZ(\mathcal{A})$. Hence, either

(5) $\quad \#(E \cap (Q')^*) \leq 1$,

or there exists $\mathcal{A}' \leq \mathcal{A}$ such that

(6) $\quad A_2\delta_{Q'} \leq \delta(x, \mathcal{A}')$ for all $x \in (Q')^* \cap E$.

It follows from (3) and (4), that (5) implies

(7) $\quad \#(E \cap (Q^+)^*) \leq 1$;

and (6) implies

(8) $\quad A_2\delta_{Q^+} \leq \delta(x, \mathcal{A}')$ for all $x \in (Q^+)^* \cap E$.

Hence, Q^+ satisfies either (7) or (8) (with $\mathcal{A}' \leq \mathcal{A}$). That is, Q^+ is almost $OK(\mathcal{A})$, thanks also to (2).

On the other hand, (2) shows that Q^+ is a cube in $\mathcal{D}(A_2)$ properly containing Q. Since $Q \in CZ(\mathcal{A})$, the definition of $CZ(\mathcal{A})$ shows that Q^+ cannot be almost $OK(\mathcal{A})$.

This contradiction completes the proof of Lemma 2. ■

Corollary. *Fix* $\mathcal{A} \subseteq \mathcal{M}$*. Then any given* $\underline{x} \in \mathbb{R}^n$ *can belong to at most* C *of the cubes* $(1+c_G)Q$ *for* $Q \in CZ(\mathcal{A})$*.*

Proof. Let $\underline{Q}$ be the cube in $CZ(\mathcal{A})$ that contains $\underline{x}$. If $Q \in CZ(\mathcal{A})$ and $\underline{x} \in (1+c_G)Q$, then by Lemma 2, we have

(9) $\quad \frac{1}{2}\delta_{\underline{Q}} \leq \delta_Q \leq 2\delta_{\underline{Q}}$ and $\underline{x} \in (1+c_G)Q$.

There are at most C dyadic cubes Q satisfying (9). ■

Lemma 3. *The cover* $\mathsf{CZ}(\mathcal{A}')$ *refines the cover* $\mathsf{CZ}(\mathcal{A})$ *if* $\mathcal{A}' < \mathcal{A}$. *That is, whenever* $Q' \in \mathsf{CZ}(\mathcal{A}')$ *and* $Q \in \mathsf{CZ}(\mathcal{A})$ *with* $Q' \cap Q \neq \emptyset$, *we have* $Q' \subseteq Q$.

Proof. Let $\mathcal{A}, \mathcal{A}', Q, Q'$ be as in the hypotheses of Lemma 3. Then Q' is almost $\mathsf{OK}(\mathcal{A}')$, which implies trivially that Q' is almost $\mathsf{OK}(\mathcal{A})$. Hence, Q' is contained in a maximal almost $\mathsf{OK}(\mathcal{A})$-cube Q''. By definition, we have $Q'' \in \mathsf{CZ}(\mathcal{A})$. Also, $Q \in \mathsf{CZ}(\mathcal{A})$, and $Q'' \cap Q \supseteq Q' \cap Q \neq \emptyset$. Therefore, $Q = Q''$, by Lemma 1. Thus, $Q' \subseteq Q'' = Q$, proving Lemma 3. ■

Lemma 4. *Let* $Q \in \mathsf{CZ}(\mathcal{A})$. *Then either* $\delta_Q = A_2^{-1}$ *or* $Q^{**} \cap E \neq \emptyset$.

Proof. Suppose $\delta_Q \neq A_2^{-1}$ and $Q^{**} \cap E = \emptyset$. We have $Q \in \mathsf{CZ}(\mathcal{A}) \subseteq \mathcal{D}(A_2)$, and $\delta_Q \neq A_2^{-1}$. Hence, $Q^+ \in \mathcal{D}(A_2)$. Also, $(Q^+)^* \cap E \subseteq Q^{**} \cap E = \emptyset$. Hence, by definition, Q^+ is almost $\mathsf{OK}(\mathcal{A})$. On the other hand, $Q^+ \in \mathcal{D}(A_2)$ and Q^+ properly contains Q. Since $Q \in \mathsf{CZ}(\mathcal{A})$, it follows from the definition of $\mathsf{CZ}(\mathcal{A})$ that Q^+ cannot be almost $\mathsf{OK}(\mathcal{A})$. This contradiction completes the proof of Lemma 4. ■

Lemma 5. $\mathsf{CZ}(\emptyset)$ *consists of all dyadic cubes of sidelength* A_2^{-1}.

Proof. Let $x \in E$. By Property (OK2) in Section 18, we have $\delta(x, \emptyset) = \infty$, because the hypotheses (a), (b), (c) of (OK2) hold vacuously for $\mathcal{A} = \emptyset$.

It follows that every dyadic cube $Q \in \mathcal{D}(A_2)$ is $\mathsf{OK}(\emptyset)$, and hence almost $\mathsf{OK}(\emptyset)$. Consequently, the cubes of $\mathsf{CZ}(\emptyset)$ are precisely the maximal cubes in $\mathcal{D}(A_2)$. These are precisely the dyadic cubes of sidelength A_2^{-1}, since A_2 is a power of 2. ■

Lemma 6. *Let* $Q \in \mathsf{CZ}(\mathcal{A})$ *and* $Q' \in \mathsf{CZ}(\mathcal{A}')$, *with* $\mathcal{A}' < \mathcal{A}$.

If $(1 + c_G)\, Q' \cap Q^{***} \neq \emptyset$, *then* $\delta_{Q'} \leq C\delta_Q$, *where* C *depends only on* n.

Proof. If not, then for a large enough constant C depending only on n, we have

$$(10) \quad \delta_{Q'} > C\delta_Q.$$

Since $(1 + c_G)Q' \cap Q^{***} \neq \emptyset$, it follows that $Q \subset (1 + 2c_G)Q'$. Also, since $Q' \in \mathsf{CZ}(\mathcal{A}')$ with $\mathcal{A}' < \mathcal{A}$, there exists $\widetilde{Q} \in \mathsf{CZ}(\mathcal{A})$ with $Q' \subseteq \widetilde{Q}$. (See Lemma 3.) We have $\widetilde{Q}, Q \in \mathsf{CZ}(\mathcal{A})$, with

$$(1 + 2c_G)\widetilde{Q} \cap (1 + 2c_G)\, Q \supseteq (1 + 2c_G)\, Q' \cap Q = Q \neq \emptyset\,.$$

Consequently, Lemma 2 yields

$$(11) \quad \tfrac{1}{2}\delta_Q \leq \delta_{\widetilde{Q}} \leq 2\delta_Q\,.$$

On the other hand, (10) gives $\delta_{\widetilde{Q}} \geq \delta_{Q'} > C\delta_Q$, contradicting (11). ■

Lemma 7. *Let* $Q \in \mathsf{CZ}(\mathcal{A})$ *and* $Q' \in \mathsf{CZ}(\mathcal{A}')$*, with* $\mathcal{A}' < \mathcal{A}$ *and* $(1 + c_G)Q \cap (1 + c_G)Q' \neq \emptyset$*. Then* $Q' \subset Q^*$*.*

Proof. By Lemma 3, there exists $\widetilde{Q} \in \mathsf{CZ}(\mathcal{A})$ with $Q' \subset \widetilde{Q}$. We have $Q, \widetilde{Q} \in \mathsf{CZ}(\mathcal{A})$, and $(1 + c_G)Q \cap (1 + c_G)\,\widetilde{Q} \neq \emptyset$. Hence Lemma 2 gives $\frac{1}{2}\delta_Q \leq \delta_{\widetilde{Q}} \leq 2\delta_Q$, and consequently $\delta_{Q'} \leq 2\delta_Q$. Since also $(1 + c_G)Q \cap (1 + c_G)Q' \neq \emptyset$, it follows from (0)(b) of Section 19 that $Q' \subset Q^*$. ■

Lemma 8. *Let* $Q_0 \in \mathsf{CZ}(\mathcal{A}_0)$ *and* $\widehat{Q} \in \mathsf{CZ}(\hat{\mathcal{A}})$*, with* $\hat{\mathcal{A}} < \mathcal{A}_0$*. Suppose that* $\delta_{\widehat{Q}} = A_2^{-1}$ *and* $(1 + c_G)Q_0 \cap (1 + c_G)\widehat{Q} \neq \emptyset$*. Then there exists* $\widetilde{Q} \in \mathsf{CZ}(\hat{\mathcal{A}})$*, such that* $\widetilde{Q} \subseteq Q_0$ *and* $(1 + c_G)\widetilde{Q} \cap (1 + c_G)\widehat{Q} \neq \emptyset$*.*

Proof. Lemma 3 shows that $\widehat{Q} \in \mathsf{CZ}(\mathcal{A}_0)$, since $\delta_{\widehat{Q}} = A_2^{-1}$ is already the largest possible sidelength for any cube in $\mathsf{CZ}(\mathcal{A}_0)$. Consequently, Lemma 2 gives $\frac{1}{2}\delta_{Q_0} \leq \delta_{\widehat{Q}} \leq 2\delta_{Q_0}$, and therefore the closures of Q_0 and $\widehat{Q}$ contain a point in common. (See (0)(c) in Section 19.) It follows that there exists a point $x_0 \in Q_0 \cap (1 + c_G)\widehat{Q}$. Let $\widetilde{Q}$ be the cube in $\mathsf{CZ}(\hat{\mathcal{A}})$ containing x_0. Then $\widetilde{Q} \subseteq Q_0$ by Lemma 3, and $x_0 \in (1 + c_G)\widetilde{Q} \cap (1 + c_G)\widehat{Q}$. ■

22. Calderón-Zygmund Cubes: Sidelengths I

In this section, we prove a few lemmas on the sidelengths of the Calderón-Zygmund cubes. In a later section, these lemmas will be used to give an efficient computation of the CZ cube containing a given point.

We write c, C, C' here, to denote constants depending only on the dimension n. We recall the constant c_G from Section 19.

We use the following definitions.

- For $x \in \mathbb{R}^n$, let $\delta_{\mathrm{nbr}}(x) = \inf\{r > 0$: At least two distinct elements of E lie within distance r of $x\}$.
- For $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathcal{M}$, let $\check{\delta}(x, \mathcal{A}) = \min_{y \in E}\{\max(|x - y|, A_2^{-1}\delta(y, \mathcal{A}))\}$.
- For $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathcal{M}$, let $\delta^{\#}(x, \mathcal{A}) = \max\{\delta_{\mathrm{nbr}}(x), \max_{\mathcal{A}' \leq \mathcal{A}} \check{\delta}(x, \mathcal{A}')\}$.
- For $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathcal{M}$, let $\delta_{\mathsf{CZ}}(x, \mathcal{A}) = \delta_Q$ for the cube $Q \in \mathsf{CZ}(\mathcal{A})$ that contains x.

The quantity $\check{\delta}(x, \mathcal{A})$ is related to the definition of $\mathsf{OK}(\mathcal{A})$, while $\delta^{\#}(x, \mathcal{A})$ is more connected with the definition of almost $\mathsf{OK}(\mathcal{A})$. Our goal here is to compute the order of magnitude of $\delta_{\mathsf{CZ}}(x, \mathcal{A})$.

Lemma 1. *For* $x \in \mathbb{R}^n$ *and* $\mathcal{A} \subseteq \mathcal{M}$*, we have*

$$c \min\{A_2^{-1}, \delta^{\#}(x, \mathcal{A})\} \leq \delta_{\mathsf{CZ}}(x, \mathcal{A}) \leq C \min\{A_2^{-1}, \delta^{\#}(x, \mathcal{A})\}.$$

Proof. First we show that

(1) $\quad \delta_{CZ}(x,\mathcal{A}) \;\le\; C \min\{A_2^{-1}, \delta^{\#}(x,\mathcal{A})\}.$

Let $Q \in CZ(\mathcal{A})$ with $x \in Q$. We must show that $\delta_Q \le C \min\{A_2^{-1}, \delta^{\#}(x,\mathcal{A})\}$. Since $Q \in CZ(\mathcal{A}) \subseteq \mathcal{D}(A_2)$, we know that $\delta_Q \le A_2^{-1}$. Hence, to prove (1), it is enough to show that

(2) $\quad \delta_Q \;\le\; C\delta^{\#}(x,\mathcal{A}).$

We know that Q is almost $OK(\mathcal{A})$. Hence, either

(3) $\quad \#(E \cap Q^*) \;\le\; 1,$

or else, for some $\mathcal{A}' \le \mathcal{A}$, we have

(4) $\quad A_2^{-1}\delta(y,\mathcal{A}') \;\ge\; \delta_Q$ for all $y \in E \cap Q^*$.

Recall that $x \in Q$. If (3) holds, then by definition of $\delta_{nbr}(x)$, we have $\delta_{nbr}(x) \ge c\delta_Q$, which implies (2), by the definition of $\delta^{\#}(x,\mathcal{A})$.

On the other hand, suppose (4) holds for some given $\mathcal{A}' \le \mathcal{A}$.

For $y \in E \cap Q^*$, we have $\max\{|x-y|, A_2^{-1}\delta(y,\mathcal{A}')\} \ge \delta_Q$ by (4).

For $y \in E \setminus Q^*$, we have $\max\{|x-y|, A_2^{-1}\delta(y,\mathcal{A}')\} \ge |x-y| \ge c\delta_Q$.

Hence,

$$\min_{y\in E} \left\{ \max(|x-y|, A_2^{-1}\delta(y,\mathcal{A}'))\right\} \ge c\delta_Q,$$

i.e., $\check{\delta}(x,\mathcal{A}') \ge c\delta_Q$ with $\mathcal{A}' \le \mathcal{A}$. Consequently, (2) holds, by definition of $\delta^{\#}(x,\mathcal{A})$.

We have shown that (2) holds in either of the two cases (3), (4). This completes the proof of (1).

Next, we show that

(5) $\quad \delta_{CZ}(x,\mathcal{A}) \ge c \cdot \min\{A_2^{-1}, \delta^{\#}(x,\mathcal{A})\}.$

To see this, let μ be a small positive constant to be picked later, and let Q be a dyadic cube containing x, with

(6) $\quad \tfrac{1}{2}\mu \cdot \min\{A_2^{-1},\, \delta^{\#}(x,\mathcal{A})\} \;\le\; \delta_Q < \mu \cdot \min\{A_2^{-1},\, \delta^{\#}(x,\mathcal{A})\}.$

(Note that $\delta^{\#}(x,\mathcal{A}) \neq 0$.) Under certain assumptions on μ, to be specified below, we will show that

(7) $\quad Q$ is almost $OK(\mathcal{A})$.

To prove (7), we require that

(8) $\quad \mu \;<\; 1.$

Then, note that (6) implies $\delta_Q < A_2^{-1}$, hence $Q \in \mathcal{D}(A_2)$. Consequently, our desired result (7) is equivalent to the following assertion:

(9) $\quad \#(E \cap Q^*) \leq 1,$

or else, for some $\mathcal{A}' \leq \mathcal{A}$, we have

(10) $\quad A_2^{-1}\,\delta(y, \mathcal{A}') \geq \delta_Q$ for all $y \in E \cap Q^*$.

From (6), we obtain $\delta_Q < \mu\,\delta^{\#}(x, \mathcal{A})$. By definition of $\delta^{\#}(x, \mathcal{A})$, this means that either

(11) $\quad \delta_Q < \mu\delta_{\mathrm{nbr}}(x),$

or else, for some $\mathcal{A}' \leq \mathcal{A}$, we have

(12) $\quad \delta_Q < \mu\,\check{\delta}(x, \mathcal{A}').$

Under a suitable assumption on μ, we will show that (11) implies (9), and that (12) implies (10). Since we know that either (11) or (12) holds, this will tell us that either (9) or (10) is satisfied, completing the proof of (7).

To see that (11) implies (9), we assume that

(13) $\mu < c$ for a small enough constant c depending only on the dimension n.

Assuming (11), we see that at most one point of E lies inside a ball of radius δ_Q/μ centered at x. By (13), and since $x \in Q$, we conclude that $\#(E \cap Q^*) \leq 1$. Thus, under the assumption (13), indeed (11) implies (9).

Next we check that (12) implies (10), under the assumption that μ satisfies (13). In fact, (12) tells us that

(14) $\quad \delta_Q \leq \mu \cdot \max\left(|x - y|,\, A_2^{-1}\,\delta(y, \mathcal{A}')\right)$ for any $y \in E$.

In particular, for $y \in E \cap Q^*$, we have $|x - y| \leq C\delta_Q$, and hence (14) implies that

$$\delta_Q \leq \max\left(C\mu\delta_Q,\, A_2^{-1}\,\delta(y, \mathcal{A}')\right) \leq \max\left(\tfrac{1}{2}\,\delta_Q,\, A_2^{-1}\,\delta(y, \mathcal{A}')\right),$$

which immediately yields (10). Thus, as claimed, (11) implies (9), and (12) implies (10). This completes the proof of (7), under the assumptions (8) and (13) on μ.

We now take μ to be a constant, depending only on n, and small enough to satisfy (8) and (13). Then (7) holds. Consequently, Q is contained in a maximal almost $\mathrm{OK}(\mathcal{A})$-cube Q'. By definition, $Q' \in \mathrm{CZ}(\mathcal{A})$, and therefore $\delta_Q \leq \delta_{Q'} = \delta_{\mathrm{CZ}}(x, \mathcal{A})$. Together with (6), this yields

(15) $\quad \delta_{\mathrm{CZ}}(x, \mathcal{A}) \geq \frac{1}{2}\mu \cdot \min\{A_2^{-1},\, \delta^{\#}(x, \mathcal{A})\}.$

Since μ now depends only on the dimension n, (15) is equivalent to the desired estimate (5). The proof of (5) is complete.

Lemma 1 is now proven, since its conclusions are our known results (1) and (5). ■

We will apply Lemma 1 for $x \in E$. To find the order of magnitude of $\delta_{CZ}(x, \mathcal{A})$ when $x \notin E$, we will use Lemmas 2, 3, 4 below.

Lemma 2. *For a large enough* C_1*, depending only on the dimension* n*, the following holds.*
Let $x \in \mathbb{R}^n$ *and* $y \in E$*, with* $|x-y| \le 2\mathsf{dist}(x,E)$*. Fix* $\mathcal{A} \subseteq \mathcal{M}$*. If*

$$\delta_{CZ}(y,\mathcal{A}) > C_1|x-y|,$$

then

$$\frac{1}{2}\delta_{CZ}(y,\mathcal{A}) \le \delta_{CZ}(x,\mathcal{A}) \le 2\delta_{CZ}(y,\mathcal{A}).$$

Proof. Let Q, Q' be the cubes in $CZ(\mathcal{A})$ containing y, x, respectively. Then

$$\delta_{CZ}(x,\mathcal{A}) = \delta_{Q'} \quad \text{and} \quad \delta_{CZ}(y,\mathcal{A}) = \delta_Q.$$

Our hypothesis gives $\delta_Q > C_1|x-y|$, with $y \in Q$. If we take C_1 large enough (depending only on n), this implies that $x \in (1+c_G)Q$. Since also $x \in Q'$, we have

$$(1+c_G)Q \cap (1+c_G)\,Q' \neq \emptyset, \quad \text{with } Q, Q' \in CZ(\mathcal{A}).$$

Lemma 2 from the preceding section gives $\frac{1}{2}\delta_Q \le \delta_{Q'} \le 2\delta_Q$, which is the conclusion of the present Lemma. ■

Fix the constant C_1 from Lemma 2.

Lemma 3. *Let* $x \in \mathbb{R}^n$*,* $y \in E$*, with* $0 < |x-y| \le 2\mathsf{dist}(x,E)$*. Fix* $\mathcal{A} \subseteq \mathcal{M}$*. If*

$$\delta_{CZ}(y,\mathcal{A}) \le 2C_1|x-y| \quad \textit{and} \quad |x-y| \le A_2^{-1},$$

then

$$\text{(16)} \qquad c|x-y| \le \delta_{CZ}(x,\mathcal{A}) \le C|x-y|.$$

Proof. Let $0 < \mu < 1/2$ be a small constant depending only on n, and let $\widehat{Q}$ be a dyadic cube containing x, with sidelength $\mu|x-y| < \delta_{\widehat{Q}} \le 2\mu|x-y|$. (Note that $|x-y| \neq 0$.)

As $|x-y| \le A_2^{-1}$ we have $\delta_{\widehat{Q}} \le A_2^{-1}$, i.e., $\widehat{Q} \in \mathcal{D}(A_2)$. Also, no points of E lie inside a ball of radius $|x-y|/4$ centered at x. Since $\delta_{\widehat{Q}} \le 2\mu|x-y|$ and $x \in \widehat{Q}$, then $\widehat{Q}^* \cap E = \emptyset$, provided that μ is a sufficiently small constant depending only on n. This shows that $\widehat{Q}$ is almost $OK(\mathcal{A})$. Consequently, $\widehat{Q} \subseteq \widehat{Q}'$ for a maximal almost $OK(\mathcal{A})$-cube $\widehat{Q}'$. By definition, $\widehat{Q}' \in CZ(\mathcal{A})$, and also $\widehat{Q}' \supseteq \widehat{Q} \ni x$; hence, $\delta_{CZ}(x,\mathcal{A}) = \delta_{\widehat{Q}'} \ge \delta_{\widehat{Q}} > \mu|x-y|$, proving half of (16). We establish the other half by contradiction. Thus, suppose

$$\text{(17)} \qquad \delta_{CZ}(x,\mathcal{A}) > C'|x-y|$$

for a large enough C' (depending only on n).

Let Q, Q' be the cubes in $CZ(\mathcal{A})$ that contain y, x, respectively. Then $\delta_{CZ}(x, \mathcal{A}) = \delta_{Q'}$, $\delta_{CZ}(y, \mathcal{A}) = \delta_Q$, and (17) gives

(18) $\delta_{Q'} > C'|x - y|$, with $x \in Q'$.

If C' is large enough, then (18) implies that $y \in (1 + c_G)Q'$. On the other hand, $y \in Q$. Hence,

$$(1 + c_G)Q \cap (1 + c_G)\, Q' \neq \emptyset, \quad \text{with} \quad Q, Q' \in CZ(\mathcal{A}).$$

Lemma 2 from the preceding section now gives $\frac{1}{2}\delta_Q \leq \delta_{Q'} \leq 2\delta_Q$, and therefore,

(19) $\delta_{CZ}(y, \mathcal{A}) = \delta_Q > \frac{1}{2}C'|x - y|$,

by (18). If C' is large enough, then (19) contradicts our hypothesis that $\delta_{CZ}(y, \mathcal{A}) \leq 2C_1|x - y|$. Hence, if we take C' large enough, depending only on n (say, $C' = 4C_1$), then (17) cannot hold. This proves the remaining half of (16), completing the proof of Lemma 3. ■

Lemma 4. *Let* $x \in \mathbb{R}^n$, $y \in E$, *with* $|x - y| \leq 2\,\mathsf{dist}(x, E)$. *If* $|x - y| \geq \frac{1}{2}A_2^{-1}$, *then*

$$cA_2^{-1} \leq \delta_{CZ}(x, \mathcal{A}) \leq A_2^{-1}.$$

Proof. The hypothesis immediately implies that $\mathsf{dist}(x, E) \geq \frac{1}{4}A_2^{-1}$. Hence, there is a dyadic cube Q containing x, with

(20) $cA_2^{-1} \leq \delta_Q \leq A_2^{-1}$, and $E \cap Q^* = \emptyset$.

From (20) we have $Q \in \mathcal{D}(A_2)$ and $\#(E \cap Q^*) = 0$, hence Q is almost $OK(\mathcal{A})$. Therefore Q is contained in a maximal almost $OK(\mathcal{A})$ cube Q'. By definition, we have $Q' \in CZ(\mathcal{A})$. Since also $Q' \supseteq Q \ni x$, it follows that

$$\delta_{CZ}(x, \mathcal{A}) = \delta_{Q'} \geq \delta_Q \geq cA_2^{-1},$$

by (20). On the other hand, since $Q' \in CZ(\mathcal{A}) \subseteq \mathcal{D}(A_2)$, we have

$$\delta_{CZ}(x, \mathcal{A}) = \delta_{Q'} \leq A_2^{-1}.$$

The proof of Lemma 4 is complete. ■

Lemmas 2, 3 and 4 reduce the computation of the order of magnitude of $\delta_{CZ}(x, \mathcal{A})$ for $x \in \mathbb{R}^n \setminus E$ to the computation of an “approximate nearest neighbor” $y \in E$, and the determination of the order of magnitude of $\delta_{CZ}(y, \mathcal{A})$ for $y \in E$.

23. BBD Trees

In this section, we recall some of the results of Arya, Mount, Netanyahu, Silverman and Wu from [1].

We work with a subset

(1) $\quad E \subset \mathbb{R}^n$, with $\#(E) = N \geq 2$.

We write c, C, C', etc. to denote constants depending only on the dimension n, and we write $X = O(Y)$ to denote the inequality $|X| \leq CY$.

Given E as in (1), and given $x \in \mathbb{R}^n$, we can enumerate the points of E as $x_1, x_2, \dots, x_N$, in such a way that

$$|x - x_1| \leq |x - x_2| \leq \cdots \leq |x - x_N|.$$

We then write $d_k(x, E)$ to denote $|x - x_k|$. Thus, $d_k(x, E)$ is the distance from x to its k^{th} nearest neighbor in the set E.

The following result is contained in [1].

Theorem BBD1. *There exists an algorithm with the following properties:*

- *The initial input consists of the set* E.
- *After receiving the initial input* E*, the algorithm performs one-time work* $O(N \log N)$ *using storage* $O(N)$.
- *After the one-time work is done, the algorithm answers queries, with work* $O(\log N)$ *per query.*
- *A query consists of a point* $x \in \mathbb{R}^n$.
- *The answer to a query* x *consists of two distinct points* $\tilde{x}_1, \tilde{x}_2 \in E$*, with*

$$|x - \tilde{x}_1| \leq 2d_1(x, E) \quad \textit{and} \quad |x - \tilde{x}_2| \leq 2d_2(x, E).$$

Here, we have stated merely what we need; the algorithm in [1] is more general and more precise than Theorem BBD1.

The proof of (the stronger version of) Theorem BBD1 in [1] is based on a data structure called a "balanced box decomposition tree", or "BBD tree". We will need to use BBD trees of a certain kind. Let us recall their definition.

A "dyadic cuboid" is a subset of $\mathbb{R}^n$, of the form

(2) $\quad 2^k \cdot \left([a_1, a_1{+}1) \times \cdots \times [a_i, a_i{+}1) \times [a_{i+1}, a_{i+1} + \tfrac{1}{2}) \times \cdots \times [a_n, a_n{+}\tfrac{1}{2})\right)$

for integers $k, i, a_1, \dots, a_i$; and integers or half-integers $a_{i+1}, \dots, a_n$. (Here, $1 \leq i \leq n$. If $i = n$, then (2) means simply $2^k([a_1, a_1{+}1) \times \cdots \times [a_n, a_n{+}1))$, which is an arbitrary dyadic cube.)

Dyadic cuboids have the following useful properties:

- Given dyadic cuboids $Q, Q' \subset \mathbb{R}^n$, we have either $Q \subseteq Q'$, $Q' \subseteq Q$, or $Q \cap Q' = \emptyset$.
- Any dyadic cuboid Q may be partitioned into two congruent dyadic cuboids Q_1, Q_2; if Q is given by (2), then we form Q_1 and Q_2 by bisecting $[a_i, a_i + 1)$. We say that Q_1 and Q_2 arise by "bisecting" Q. This differs from our use of the word "bisecting" for dyadic cubes.

Given $E \subset \mathbb{R}^n$ as in (1), a "BBD Tree" for E is a tree T with the following properties.

BBD1: Each node of T other than the root is either a dyadic cuboid, or otherwise a set of the form $Q \setminus Q'$, where $Q' \subset Q$, and Q, Q' are dyadic cuboids. The latter set is called a "punctured dyadic cuboid".

BBD2: Any node other than the root has either two children ("an internal node"), or zero children ("a leaf"). An internal node is the disjoint union of its two children.

BBD3: The root of T is a disjoint union of at most 2^n dyadic cubes that contains the entire set E. The root has at most 2^n children, that are all dyadic cubes. The root is the disjoint union of its children.

BBD4: A node $A \in T$ is a leaf if and only if $\#(A \cap E) \leq 1$. For any leaf A, we mark whether $A \cap E$ is empty or not.

BBD5: Each node $A \in T$ is marked with a "representative" $x_A \in E$ that satisfies $x_A \in A$ in case $A \cap E \neq \emptyset$.

BBD6: The tree has height $O(\log N)$, and the number of nodes is $O(N)$.

BBD7: Let $A \in T$ be a node other than the root. Assume that A is an internal node. Then the children of A arise as follows:

(I) If A is a dyadic cuboid Q, then either

- The children of A are the two dyadic cuboids Q_1, Q_2 that arise by "bisecting" Q (a "split"); or
- There exists a dyadic cuboid $\widetilde{Q} \subset Q$, such that the children of A are $\widetilde{Q}$ and $Q \setminus \widetilde{Q}$ (a "puncture").

(Formally, it is possible to view a split as a puncture. We choose not to do so. Whenever we say "a puncture", we mean one which is not a split.)

(II) If A is a punctured cuboid $Q \setminus Q'$, then either

- The children of A are $Q_1 \setminus Q'$ and Q_2, where Q_1 and Q_2 arise by "bisecting" Q, and $Q' \subseteq Q_1$ (a "split"); or
- There exists a dyadic cuboid $\widetilde{Q}$, with $Q' \subset \widetilde{Q} \subset Q$, such that the children of A are $Q \setminus \widetilde{Q}$ and $\widetilde{Q} \setminus Q'$ ("enlarging the hole").

(Again, whenever we say that the children of A arise by "enlarging the hole", we mean in particular that they do not arise by a "split".)

Note that it may happen that the entire set E is not contained in a single dyadic cube; this is the reason why we allow the root to be a disjoint union of dyadic cubes. (To avoid trivalities, the set E is assumed in [1] to be contained in a single large dyadic cube.)

The main result in [1], and the main step in the proof of Theorem BBD1, is as follows.

Theorem BBD2. *There exists an algorithm that computes a* BBD *tree for a given* $E \subset \mathbb{R}^n$ *as in* (1), *with work* $O(N \log N)$ *and storage* $O(N)$.

The differences between our definition of the BBD tree and the definition in [1] are minor and non-essential. We will compute a BBD tree for E, as part of our one-time work.

Remark. Formally, any dyadic cuboid may be viewed as $Q \backslash Q'$ where Q, Q' are non-empty dyadic cuboids. We would like to emphasize that whenever we say that $A \subset \mathbb{R}^n$ is a punctured dyadic cuboid, we mean in particular that A is not a dyadic cuboid. Thus, a (non-empty) punctured dyadic cuboid is uniquely represented as $Q \setminus Q'$ for dyadic cuboids Q, Q'.

24. Calderón-Zygmund Cubes: Sidelengths II

Our goal in this section is to give an algorithm to compute, for each $\mathcal{A} \subseteq \mathcal{M}$ and $x \in E$, the order of magnitude of $\delta_{CZ}(x, \mathcal{A})$. Recall that $\delta_{CZ}(x, \mathcal{A})$ was defined in Section 22 as δ_Q, where Q is the cube in $CZ(\mathcal{A})$ that contains x.

Note that in this section, we compute the order of magnitude of $\delta_{CZ}(x, \mathcal{A})$ only for $x \in E$. In a later section, we will deal with general $x \in \mathbb{R}^n$.

We assume here that the lengthscales $\delta(x, \mathcal{A}) \in [0, \infty]$ have already been computed, for all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

We recall from Section 22 the definitions of $\delta_{nbr}(x)$, $\check{\delta}(x, \mathcal{A})$, $\delta^{\#}(x, \mathcal{A})$, $\delta_{CZ}(x, \mathcal{A})$. We will compute the orders of magnitude of these quantities, for all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

We write C, C', etc., here to denote constants depending only on m and n.

First of all, by applying Theorem BBD1 from Section 23, we can compute numbers $\underline{\delta}_{nbr}(x)$ for all $x \in E$, satisfying

(1) $\quad \frac{1}{2}\, \underline{\delta}_{nbr}(x) \;\leq\; \delta_{nbr}(x) \;\leq\; \underline{\delta}_{nbr}(x)$ for all $x \in E$;

the computation of the $\underline{\delta}_{nbr}(x)$ for all $x \in E$ uses work at most $\mathsf{C}N \log N$ and storage at most $\mathsf{C}N$.

Next, we turn our attention to the numbers $\check{\delta}(x, \mathcal{A})$ for $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$. We fix a Callahan-Kosaraju decomposition $(\mathcal{T}, \mathcal{L})$, with parameter $\varkappa = 1/2$. (See Section 9 for the notation.) Recall that it takes work at most $\mathsf{C}N \log N$ and storage at most $\mathsf{C}N$ to compute $(\mathcal{T}, \mathcal{L})$.

Recall also that, for each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, we have computed and stored "representatives" $x'_{\Lambda_1} \in \cup \Lambda_1$ and $x''_{\Lambda_2} \in \cup \Lambda_2$. Fix $\mathcal{A} \subseteq \mathcal{M}$. We perform the following computations.

<u>Step 1</u>: For each $A'' \in \mathcal{T}$, we compute $\delta''(A'') := \min\limits_{y \in A''} \delta(y, \mathcal{A})$.

<u>Step 2</u>: For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, we compute $\delta(\Lambda_1, \Lambda_2) := \min\limits_{A'' \in \Lambda_2} \delta''(A'')$.

<u>Step 3</u>: For each $A' \in \mathcal{T}$, we compute

$$\delta'(A') := \min_{\substack{(\Lambda_1, \Lambda_2) \in \mathcal{L} \\ \Lambda_1 \ni A'}} \left\{ \max(|x'_{\Lambda_1} - x''_{\Lambda_2}|, A_2^{-1} \delta(\Lambda_1, \Lambda_2)) \right\}.$$

<u>Step 4</u>: For each $x \in E$, we compute $\delta_{\min}(x) := \min\limits_{\substack{A' \in \mathcal{T} \\ A' \ni x}} \delta'(A')$.

The properties of the Callahan-Kosaraju decomposition guarantee that we can carry out the above computations, with total work at most $\mathsf{C}N \log N$, and storage at most $\mathsf{C}N$. (See (4), (5), (6) in Section 9.)

By inspection of Steps 1 and 2 above, we have

(2) $\quad \delta(\Lambda_1, \Lambda_2) \;=\; \min\limits_{y \in \cup \Lambda_2} \delta(y, \mathcal{A})$, for each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$.

We recall the definition

(3) $\quad \check{\delta}(x, \mathcal{A}) \;=\; \min\limits_{y \in E} \left\{ \max(|x - y|, A_2^{-1} \delta(y, \mathcal{A})) \right\}$ for any $x \in \mathbb{R}^n$.

The next lemma shows that the order of magnitude of $\check{\delta}(x, \mathcal{A})$ is known, for all $x \in E$, once we have computed the $\delta_{\min}(x)$ $(x \in E)$.

Lemma 1. *For all* $x \in E$*, we have*

(4) $\quad \frac{1}{2} \min\{\delta_{\min}(x),\, A_2^{-1} \delta(x, \mathcal{A})\} \;\leq\; \check{\delta}(x, \mathcal{A}) \leq 2 \min\{\delta_{\min}(x),\, A_2^{-1} \delta(x, \mathcal{A})\}$.

Proof. By definition of $\delta_{\min}(x)$, we have

(5) $\delta_{\min}(x) = \delta'(A')$ for an $A' \in \mathcal{T}$ with $x \in A'$.

Fix such an A'. By definition of $\delta'(A')$, we have

(6) $\delta'(A') = \max\left\{|x'_{\Lambda_1} - x''_{\Lambda_2}|,\ A_2^{-1}\delta(\Lambda_1, \Lambda_2)\right\}$, for some $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ with $A' \in \Lambda_1$.

Fix such a (Λ_1, Λ_2). Combining (2), (5), (6), we see that

(7) $\delta_{\min}(x) = \max\left\{|x'_{\Lambda_1} - x''_{\Lambda_2}|,\ A_2^{-1}\delta(\bar{y}, \mathcal{A})\right\}$, for some $\bar{y} \in \cup\Lambda_2$.

We have now $x, x'_{\Lambda_1} \in \cup\Lambda_1$ and $\bar{y}, x''_{\Lambda_2} \in \cup\Lambda_2$, with $(\Lambda_1, \Lambda_2) \in \mathcal{L}$. Since $(\mathcal{T}, \mathcal{L})$ is a Callahan-Kosaraju decomposition with $\varkappa = 1/2$, it follows that $|x'_{\Lambda_1} - x''_{\Lambda_2}| \geq \frac{1}{2}|x - \bar{y}|$, and therefore (7) yields

(8) $\delta_{\min}(x) \geq \frac{1}{2}\max\left(|x - \bar{y}|,\ A_2^{-1}\delta(\bar{y}, \mathcal{A})\right)$, with $\bar{y} \in \cup\Lambda_2 \subseteq E$.

Comparing (8) with (3), we see that

(9) $\delta_{\min}(x) \geq \frac{1}{2}\check{\delta}(x, \mathcal{A})$.

From (3) we also obtain (trivially)

(10) $A_2^{-1}\delta(x, \mathcal{A}) \geq \check{\delta}(x, \mathcal{A})$.

Estimates (9), (10) together imply the upper bound for $\check{\delta}(x, \mathcal{A})$ in (4).

We turn our attention to the lower bound. Fix $\bar{y} \in E$ to achieve the minimum in (3). Thus,

(11) $\check{\delta}(x, \mathcal{A}) = \max\left\{|x - \bar{y}|, A_2^{-1}\delta(\bar{y}, \mathcal{A})\right\}$.

If $\bar{y} = x$, then (11) gives $\check{\delta}(x, \mathcal{A}) = A_2^{-1}\delta(x, \mathcal{A})$, in which case the lower bound for $\check{\delta}(x, \mathcal{A})$ in (4) is obvious. Suppose instead that $\bar{y} \neq x$. By the defining property of the Callahan-Kosaraju decomposition, there exists $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ with $x \in \cup\Lambda_1$ and $\bar{y} \in \cup\Lambda_2$. Fix such a (Λ_1, Λ_2), and fix also $A', A'' \in \mathcal{T}$, with $x \in A'$, $A' \in \Lambda_1$, $\bar{y} \in A''$, $A'' \in \Lambda_2$. (Such A', A'' exist, since $x \in \cup\Lambda_1$ and $\bar{y} \in \cup\Lambda_2$.) We have $x, x'_{\Lambda_1} \in \cup\Lambda_1$, and $\bar{y}$, $x''_{\Lambda_2} \in \cup\Lambda_2$, with $(\Lambda_1, \Lambda_2) \in \mathcal{L}$. Another appeal to the defining properties of a Callahan-Kosaraju decomposition (with $\varkappa = 1/2$) yields

(12) $|x'_{\Lambda_1} - x''_{\Lambda_2}| \leq 2|x - \bar{y}|$.

We inspect the algorithm for the computation of $\delta_{\min}(x)$:

(13) By Step 4, $\delta_{\min}(x) \leq \delta'(A')$;

(14) By Step 3, $\delta'(A') \leq \max\left(|x'_{\Lambda_1} - x''_{\Lambda_2}|, A_2^{-1}\delta(\Lambda_1, \Lambda_2)\right)$;

(15) By Step 2, $\delta(\Lambda_1, \Lambda_2) \leq \delta''(A'')$; and

(16) By Step 1, $\delta''(A'') \leq \delta(\bar{y}, \mathcal{A})$.

(Here, we use the fact that $x \in A'$, $A' \in \Lambda_1$, $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, $A'' \in \Lambda_2$, and $\bar{y} \in A''$.)
Combining (13),. . ., (16), we obtain $\delta_{\min}(x) \leq \max\left(|x'_{\Lambda_1} - x''_{\Lambda_2}|, A_2^{-1}\, \delta(\bar{y}, \mathcal{A})\right)$. Together with (12), this implies that $\delta_{\min}(x) \leq 2\max(|x - \bar{y}|, A_2^{-1}\delta(\bar{y}, \mathcal{A}))$, and therefore (11) yields $\frac{1}{2}\,\delta_{\min}(x) \leq \check{\delta}(x, \mathcal{A})$.

This immediately implies the lower bound for $\check{\delta}(x, \mathcal{A})$ in (4). Thus, that lower bound holds in all cases. The proof of Lemma 1 is complete. ■

We now let $\mathcal{A} \subseteq \mathcal{M}$ vary, and we write $\delta_{\min}(x, \mathcal{A})$ in place of $\delta_{\min}(x)$. We can compute all the $\delta_{\min}(x, \mathcal{A})$ for $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$ with work at most $CN \log N$ and storage at most CN. Lemma 1 tells us that

(17) $$\frac{1}{2}\check{\delta}(x, \mathcal{A}) \leq \min\left\{\delta_{\min}(x, \mathcal{A})\,, A_2^{-1}\delta(x, \mathcal{A})\right\} \leq 2\check{\delta}(x, \mathcal{A}).$$

for all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

Recall that, for any $x \in E$ and $\mathcal{A} \subseteq \mathcal{M}$, we defined

(18) $$\delta^{\#}(x, \mathcal{A}) = \max\left\{\delta_{\text{nbr}}(x), \max_{\mathcal{A}' \leq \mathcal{A}} \check{\delta}(x, \mathcal{A}')\right\}.$$

Let us define

(19) $\underline{\delta}^{\#}(x, \mathcal{A}) = \max\left\{\underline{\delta}_{\text{nbr}}(x), \max_{\mathcal{A}' \leq \mathcal{A}} \left[\min\{\delta_{\min}(x, \mathcal{A}'), A_2^{-1}\delta(x, \mathcal{A}')\}\right]\right\}$, for $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

For each given $x, \mathcal{A}$, we can compute $\underline{\delta}^{\#}(x, \mathcal{A})$ from the known quantities $\underline{\delta}_{\text{nbr}}(x)$, $\delta_{\min}(x, \mathcal{A}')$, $\delta(x, \mathcal{A}')$ $(\mathcal{A}' \leq \mathcal{A})$, with work at most C. Hence, we may compute and store all the $\underline{\delta}^{\#}(x, \mathcal{A})$ $(x \in E, \mathcal{A} \subseteq \mathcal{M})$ with work and storage at most CN. Moreover, comparing (18) with (19), and applying (1) and (17), we learn that

(20) $\frac{1}{2}\,\underline{\delta}^{\#}(x, \mathcal{A}) \leq \delta^{\#}(x, \mathcal{A}) \leq 2\underline{\delta}^{\#}(x, \mathcal{A})$ for all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

Finally, we set

(21) $\underline{\delta}_{CZ}(x, \mathcal{A}) = \min\left\{A_2^{-1}, \underline{\delta}^{\#}(x, \mathcal{A})\right\}$ for $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

Thus,

(22) We can compute all the quantities (21) starting from the set E, and the lengthscales $\delta(x, \mathcal{A})$ $(x \in E$, $\mathcal{A} \subseteq \mathcal{M})$, with work at most $CN \log N$, and storage at most CN.

Moreover, by comparing (20) and (21) with Lemma 1 from Section 22, we discover that

(23) $c\underline{\delta}_{CZ}(x,\mathcal{A}) \leq \delta_{CZ}(x,\mathcal{A}) \leq C\underline{\delta}_{CZ}(x,\mathcal{A})$ for all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

We record (21), (22), (23) as a lemma.

Lemma 2. *With work at most* $CN \log N$ *and storage at most* CN*, we can compute numbers* $\underline{\delta}_{CZ}(x,\mathcal{A})$ *(for all* $x \in E$*,* $\mathcal{A} \subseteq \mathcal{M}$*), having the following property:*

Let $x \in E$ *and* $\mathcal{A} \subseteq \mathcal{M}$ *be given, and let* Q *be the cube in* $CZ(\mathcal{A})$ *that contains* x*. Then* $c\underline{\delta}_{CZ}(x,\mathcal{A}) \leq \delta_Q \leq C\underline{\delta}_{CZ}(x,\mathcal{A})$*. That is,*

$$c\underline{\delta}_{CZ}(x,\mathcal{A}) \leq \delta_{CZ}(x,\mathcal{A}) \leq C\underline{\delta}_{CZ}(x,\mathcal{A}).$$

Here, c *and* C *depend only on* m *and* n*.*

Thus, we have succeeded in computing the order of magnitude of the sidelengths of the CZ cubes containing points of E.

25. Recognizing a CZ Cube

The goal of this section is to give an efficient algorithm to recognize whether a given dyadic cube Q belongs to $CZ(\mathcal{A})$ for a given subset $\mathcal{A} \subseteq \mathcal{M}$. We use the BBD tree for the set E. Let us keep the notation from the Section 23, except that in this section C, C', etc. stand for constants depending only on m and n.

We suppose that we have already precomputed the lengthscales $\delta(x,\mathcal{A}) \in [0,\infty]$, for all $x \in E$ and $\mathcal{A} \subseteq \mathcal{M}$. We are given the constant A_2.

For any subsets $\mathcal{A} \subseteq \mathcal{M}$ and $\Omega \subset \mathbb{R}^n$, we write $\delta_{\min}(\Omega,\mathcal{A})$ to denote the minimum of $\delta(x,\mathcal{A})$ over all $x \in E \cap \Omega$. (If $E \cap \Omega = \emptyset$, then we set $\delta_{\min}(\Omega,\mathcal{A}) = \infty$.)
Note that, if Ω is partitioned into $\Omega_1, \ldots, \Omega_s$, then

(1) $\delta_{\min}(\Omega,\mathcal{A}) = \min\limits_{i=1,\ldots,s} \delta_{\min}(\Omega_i,\mathcal{A})$.

Recall that the leaves of our BBD tree T are marked. In particular, for each leaf A, either $E \cap A = \emptyset$ or $\#(E \cap A) = 1$. The leaf A is marked to show whether $E \cap A$ is empty, and to exhibit the (unique) element $x_A \in E \cap A$, in case $E \cap A$ is non-empty. Recall also that each internal node A other than the root is the disjoint union of its two children A_1 and A_2. The root is the disjoint union of its children, and it has at most 2^n children. Consequently, a trivial "bottom-up" recursive algorithm (using (1)) allows us to compute the quantities

(2) $\delta_{\min}(A,\mathcal{A})$ for all $\mathcal{A} \subseteq \mathcal{M}$, and also $\#(E \cap A)$,

for each node A of the BBD tree.

The work used to carry out the trivial algorithm, and thus to mark each node $\mathsf{A} \in \mathsf{T}$ with the information (2), is at most $\mathsf{C} \cdot \#\{\text{nodes of } \mathsf{T}\} \leq \mathsf{C}'\mathsf{N}$. We assume from now on that the nodes of T are marked with the information (2).

We introduce the notion of the "hull" of a node A. If A is a cuboid Q, then we define hull $(A) = \mathsf{Q}$. Otherwise, A is a punctured cuboid $\mathsf{Q} \setminus \mathsf{Q}'$ and we define hull $(A) = \mathsf{Q}$.

We will base our algorithms on the following elementary result.

Proposition 1. *Let $\mathsf{A} \in \mathsf{T}$ be an internal node other than the root, let $\widehat{\mathsf{Q}}$ be a dyadic cuboid, and suppose that $\widehat{\mathsf{Q}} \subseteq$ hull (A). Then the set $\{\mathsf{A}_1, \mathsf{A}_2\}$, consisting of the children of A, may be partitioned into three subsets X_{in}, $\mathsf{X}_{\mathsf{out}}$, $\mathsf{X}_{\mathsf{hard}}$, with the following properties*:

(a) *All $\mathsf{A}' \in \mathsf{X}_{\mathsf{in}}$ satisfy $\mathsf{A}' \subset \widehat{\mathsf{Q}}$.*

(b) *All $\mathsf{A}' \in \mathsf{X}_{\mathsf{out}}$ satisfy $\mathsf{A}' \cap \widehat{\mathsf{Q}} = \emptyset$.*

(c) *All $\mathsf{A}'' \in \mathsf{X}_{\mathsf{hard}}$ satisfy $\widehat{\mathsf{Q}} \subseteq$ hull (A'').*

(d) *$\mathsf{X}_{\mathsf{hard}}$ is empty or consists of a single node A''.*

Proof. The node A is an internal node, and is nor the root neither a leaf. Therefore, A is non-empty, and is either a dyadic cuboid or else a punctured dyadic cuboid. Let $\mathsf{Q} = \text{hull}(A)$. We will define a set Q' as follows. If A is a cuboid, we set $\mathsf{Q}' = \emptyset$. Otherwise, A is a punctured cuboid and we define Q' so that $A = \mathsf{Q} \setminus \mathsf{Q}'$. In both cases, $A = \mathsf{Q} \setminus \mathsf{Q}'$, and Q' is properly contained in $\mathsf{Q} =$ hull (A). We proceed by cases.

Case 1: Suppose A is split. Let Q_1 and Q_2 be the two dyadic cuboids obtained by "bisecting" Q. Then Q' is contained in either Q_1 or Q_2. Without loss of generality, say $\mathsf{Q}' \subseteq \mathsf{Q}_1$. Then the children of A are $A_1 = \mathsf{Q}_1 \setminus \mathsf{Q}'$ and $A_2 = \mathsf{Q}_2$. The hull of the cuboid A_2 is Q_2.

Since $\widehat{\mathsf{Q}} \subseteq$ hull $(A) = \mathsf{Q}$, we have either $\widehat{\mathsf{Q}} = \mathsf{Q}$, $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}_2$, or $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}_1$.

- If $\widehat{\mathsf{Q}} = \mathsf{Q}$, then we take $\mathsf{X}_{\mathsf{in}} = \{\mathsf{A}_1, \mathsf{A}_2\}$, $\mathsf{X}_{\mathsf{out}} = \emptyset$, $\mathsf{X}_{\mathsf{hard}} = \emptyset$.
- If $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}_2$, then we take $\mathsf{X}_{\mathsf{in}} = \emptyset$, $\mathsf{X}_{\mathsf{out}} = \{\mathsf{A}_1\}$, $\mathsf{X}_{\mathsf{hard}} = \{\mathsf{A}_2\}$.
- If $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}_1$ and hull$(\mathsf{A}_1) = \mathsf{Q}_1$, then we take $\mathsf{X}_{\mathsf{in}} = \emptyset$, $\mathsf{X}_{\mathsf{out}} = \{\mathsf{A}_2\}$, $\mathsf{X}_{\mathsf{hard}} = \{\mathsf{A}_1\}$.

In each of the three sub-cases above, properties (a),. . .,(d) hold. We still need to handle the case where $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}_1$ but hull$(\mathsf{A}_1) \neq \mathsf{Q}_1$. In this case, necessarily $A_1 = \text{hull}(A_1) = \mathsf{Q}_1 \setminus \mathsf{Q}'$ is actually a cuboid (see Figure 2). Therefore $\mathsf{Q}_1 = A_1 \cup \mathsf{Q}'$, a disjoint union of cuboids. Since $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}_1$ then either $\widehat{\mathsf{Q}} = \mathsf{Q}_1, \widehat{\mathsf{Q}} \subseteq A_1$ or $\widehat{\mathsf{Q}} \subseteq \mathsf{Q}'$.

- If $\widehat{\mathsf{Q}} = \mathsf{Q}_1$, then we take $\mathsf{X}_{\mathsf{in}} = \{\mathsf{A}_1\}$, $\mathsf{X}_{\mathsf{out}} = \{\mathsf{A}_2\}$ and $\mathsf{X}_{\mathsf{hard}} = \emptyset$.

- If $\widehat{Q} \subseteq A_1$, then we take $X_{\mathsf{in}} = \emptyset$, $X_{\mathsf{out}} = \{A_2\}$ and $X_{\mathsf{hard}} = \{A_1\}$.
- If $\widehat{Q} \subseteq Q'$, then we take $X_{\mathsf{in}} = \emptyset$, $X_{\mathsf{out}} = \{A_1, A_2\}$ and $X_{\mathsf{hard}} = \emptyset$.

In each of the three sub-cases above, properties (a),. . .,(d) hold. Thus the Proposition holds in Case 1.

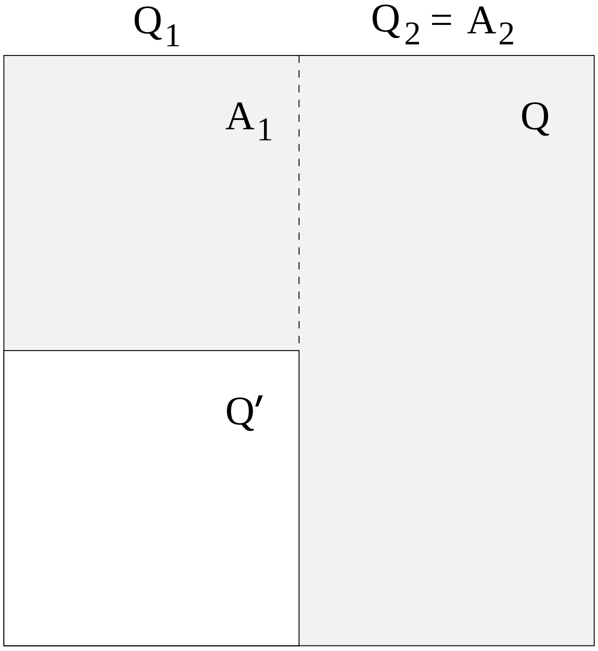

Figure 2

Case 2: Suppose A is punctured, or else its hole is enlarged. Then there is a dyadic cuboid $\widetilde{Q}$, such that $Q' \subset \widetilde{Q} \subset Q$, and the children of A are $A_1 = Q \setminus \widetilde{Q}$, $A_2 = \widetilde{Q} \setminus Q'$. Since the operation here is not a split, then $\text{hull}(A_1) = Q$. Recall that $\widehat{Q} \subset Q = \text{hull}(A)$. We have either $\widetilde{Q} \subsetneq \widehat{Q}$, $\widetilde{Q} \cap \widehat{Q} = \emptyset$ or $\widehat{Q} \subseteq \widetilde{Q}$.

- If $\widetilde{Q} \subsetneq \widehat{Q}$, then we take $X_{\mathsf{in}} = \{A_2\}$, $X_{\mathsf{out}} = \emptyset$, $X_{\mathsf{hard}} = \{A_1\}$.
- If $\widetilde{Q} \cap \widehat{Q} = \emptyset$, then we take $X_{\mathsf{in}} = \emptyset$, $X_{\mathsf{out}} = \{A_2\}$, $X_{\mathsf{hard}} = \{A_1\}$.
- If $\widehat{Q} \subseteq \widetilde{Q}$ and $\text{hull}(A_2) = \widetilde{Q}$, then we take $X_{\mathsf{in}} = \emptyset$, $X_{\mathsf{out}} = \{A_1\}$, $X_{\mathsf{hard}} = \{A_2\}$.

In each of the three sub-cases above, properties (a),. . .,(d) hold. We still need to consider the case where $\widehat{Q} \subseteq \widetilde{Q}$ and $\text{hull}(A_2) \neq \widetilde{Q}$ (see Figure 3). In this case $A_2 = \text{hull}(A_2) = \widetilde{Q} \setminus Q'$ is a cuboid. Hence $\widetilde{Q} = A_2 \cup Q'$ is a disjoint union of cuboids. Since $\widehat{Q} \subseteq \widetilde{Q}$ then either $\widehat{Q} = \widetilde{Q}, \widehat{Q} \subseteq A_2$ or $\widehat{Q} \subseteq Q'$.

- If $\widehat{Q} = \widetilde{Q}$, then we take $X_{\mathsf{in}} = \{A_2\}$, $X_{\mathsf{out}} = \{A_1\}$ and $X_{\mathsf{hard}} = \emptyset$.
- If $\widehat{Q} \subset A_2$, then we take $X_{\mathsf{in}} = \emptyset$, $X_{\mathsf{out}} = \{A_1\}$ and $X_{\mathsf{hard}} = \{A_2\}$.
- If $\widehat{Q} \subset Q'$, then we take $X_{\mathsf{in}} = \emptyset$, $X_{\mathsf{out}} = \{A_1, A_2\}$ and $X_{\mathsf{hard}} = \emptyset$.

In each of the three sub-cases above, properties (a),...,(d) hold. Thus, the Proposition holds also in Case 2. We conclude that the Proposition is proven in all cases. ■

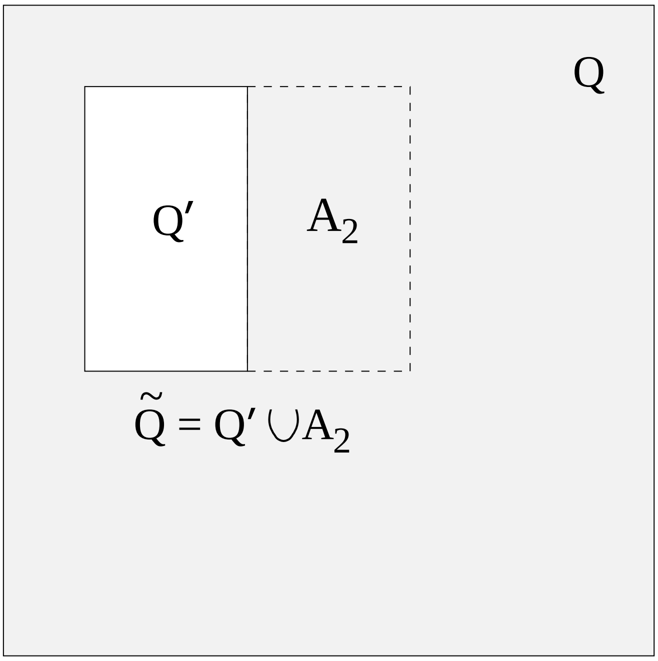

Figure 3

Given an internal node $A \in T$ other than the root, and a dyadic cuboid $\widehat{Q} \subset \text{hull}\,(A)$, the computation of $X_{\mathsf{in}}, X_{\mathsf{out}}$ and X_{hard} as in Proposition 1 is straightforward, and requires no more than C computer operations. We exploit Proposition 1 in the following algorithm.

Algorithm RCZ0: Given a node $A \in T$ other than the root and a dyadic cuboid $\widehat{Q} \subseteq \text{hull}\,(A)$, we compute the quantities

$$(3) \qquad \#(E \cap \widehat{Q} \cap A), \text{ and } \delta_{\min}(\widehat{Q} \cap A, \mathcal{A}) \text{ for all } \mathcal{A} \subseteq \mathcal{M}.$$

Explanation: If A is a leaf, then $E \cap A$ is empty or a singleton $\{x_A\}$; the marking of A indicates which case occurs, as well as x_A (in case $E \cap A \neq \emptyset$). Hence, it is trivial to compute the quantities (3) when A is a leaf.

Suppose A is an internal node that is not the root. Recall that A is the disjoint union of its children. We partition the set of children of A into X_{in}, X_{out} and X_{hard}, as in Proposition 1.

Either X_{hard} is empty, or $X_{\mathsf{hard}} = \{A''\}$ for a single node A'', with $\widehat{Q} \subseteq$ hull (A'').

If X_{hard} is empty, then $\widehat{Q} \cap A$ is the disjoint union of the A' in X_{in}. Consequently,

$$(4) \qquad \#(E \cap \widehat{Q} \cap A) = \sum_{A' \in X_{\mathsf{in}}} \#(E \cap A'), \text{ and}$$

$$(5) \qquad \delta_{\min}(\widehat{Q} \cap A, \mathcal{A}) = \min_{A' \in X_{\mathsf{in}}} \delta_{\min}(A', \mathcal{A}) \text{ for all } \mathcal{A} \subseteq \mathcal{M},$$

thanks to (1). The right-hand sides of (4), (5) may be trivially computed from our markings (2) for the nodes $A' \in X_{\mathsf{in}}$. Hence, (4) and (5) yield the desired information (3), in case X_{hard} is empty.

If $X_{\mathsf{hard}} = \{A''\}$, then $\widehat{Q} \cap A$ is the disjoint union of $\widehat{Q} \cap A''$ and all the A' in X_{in}. Consequently,

$$(6) \quad \#(E \cap \widehat{Q} \cap A) = \#(E \cap \widehat{Q} \cap A'') + \sum_{A' \in X_{\mathsf{in}}} \#(E \cap A'), \text{ and}$$

$$(7) \quad \delta_{\min}(\widehat{Q} \cap A, \mathcal{A}) = \min \left\{ \delta_{\min}(\widehat{Q} \cap A'', \mathcal{A}), \min_{A' \in X_{\mathsf{in}}} \delta_{\min}(A', \mathcal{A}) \right\} \text{ for all } \mathcal{A} \subseteq \mathcal{M}.$$

As with (4) and (5), the right-hand sides of (6) and (7) may be computed easily from the markings (2) for $A' \in X_{\mathsf{in}}$, once we know the quantities

$$(8) \qquad \#(E \cap \widehat{Q} \cap A''), \text{ and } \delta_{\min}(\widehat{Q} \cap A'', \mathcal{A}), \text{ for all } \mathcal{A} \subseteq \mathcal{M}.$$

To compute the quantities (8), we apply Algorithm RCZ0 recursively, to the node A'' and the dyadic cuboid $\widehat{Q}$. Note that A'' is a child of A, and that $\widehat{Q} \subseteq$ hull (A''), since $A'' \in X_{\mathsf{hard}}$. (See conclusion (c) of Proposition 1.) Thus, if the recursive call to Algorithm RCZ0 terminates, then we obtain the quantities (8), and substitute them into (6) and (7) to obtain the desired information (3). Apart from recursing, the amount of work we perform while inspecting the node A and the cuboid $\widehat{Q}$ is clearly bounded by C.

This completes our description of Algorithm RCZ0.

Note that Algorithm RCZ0 terminates, and takes work at most $C \log N$. To see this, suppose we apply Algorithm RCZ0 to a given node A_0, different from the root, and a given cuboid $\widehat{Q}$. Then the algorithm recursively calls itself, with A_0 replaced successively by $A_1, A_2, \dots$ where A_ν is a child of $A_{\nu-1}$

for each ν. Since the tree T has height at most $C \log N$, it follows that the algorithm terminates after at most $C \log N$ recursive calls. This implies easily that the work of Algorithm RCZ0 is at most $C \log N$, as claimed.

The next step is as follows.

Algorithm RCZ1: Given a dyadic cuboid $\widehat{Q}$, we compute

$$\text{(9)} \qquad \#(E \cap \widehat{Q}) \text{ and } \delta_{\min}(\widehat{Q}, \mathcal{A}) = \min_{x \in E \cap \widehat{Q}} \delta(x, \mathcal{A}) \text{ for each } \mathcal{A} \subseteq \mathcal{M}.$$

Explanation: The root of the BBD tree T is a disjoint union of dyadic cuboids $A^1, \dots, A^L$ with $L \leq 2^n$. The set E is contained in $A^1 \cup \ldots \cup A^L$. For each $i = 1, \dots, L$, we will compute

$$\text{(10)} \qquad \#(E \cap \widehat{Q} \cap A^i) \text{ and } \delta_{\min}(\widehat{Q} \cap A^i, \mathcal{A}) \text{ for each } \mathcal{A} \subseteq \mathcal{M}.$$

The cubes A^i are disjoint and their union contains E. Hence, once the quantities in (10) are obtained, the computation of the information in (9) is obvious. For each $i = 1, \dots, L$, either $A^i \subseteq \widehat{Q}$, $\widehat{Q} \cap A^i = \emptyset$, or $\widehat{Q} \subseteq A^i$. The information (10) is obtained trivially in the first two cases. In the third case, we have $\widehat{Q} \subseteq A^i = \text{hull}\ (A^i)$, and A^i is a node in T that is not the root. Hence, the desired information (10) may be read off from Algorithm RCZ0 (with $A = A^i$) in the non-trivial case. The work of the algorithm is at most $C \log N$. This concludes our description and analysis of Algorithm RCZ1.

Let $Q \subset \mathbb{R}^n$ be a dyadic cube, and let $\mathcal{A} \subseteq \mathcal{M}$ be given. The cube Q^* is a disjoint union of 5^n dyadic cubes Q_ν, in an obvious way. Applying Algorithm RCZ1 to each of the Q_ν, we can easily compute the quantities

$\#(E \cap Q^*)$, and $\min\{\delta(x, \mathcal{A}') : x \in E \cap Q^*\}$ for all $\mathcal{A}' \leq \mathcal{A}$.

This allows us to decide whether Q is almost $\mathrm{OK}(\mathcal{A})$, with work at most $C \log N$.

It follows easily from the definitions of "$\mathrm{CZ}(\mathcal{A})$" and "almost $\mathrm{OK}(\mathcal{A})$" that a dyadic cube $Q \subset \mathbb{R}^n$ belongs to $\mathrm{CZ}(\mathcal{A})$ if and only if Q is almost $\mathrm{OK}(\mathcal{A})$, but Q^+ is not almost $\mathrm{OK}(\mathcal{A})$. (Here, Q^+ is the dyadic parent of Q. In particular, if $\delta_Q = A_2^{-1}$, then Q^+ cannot be almost $\mathrm{OK}(\mathcal{A})$, since $Q^+ \notin \mathcal{D}(A_2)$.)

Hence, we have proven the following result.

Lemma 1. *After performing* $CN \log N$ *one-time work, with storage* CN*, we can answer queries as follows:*

Given a dyadic cube $Q \subset \mathbb{R}^n$ *and a subset* $\mathcal{A} \subseteq \mathcal{M}$*, we can decide with work* $C \log N$ *whether* $Q \in \mathrm{CZ}(\mathcal{A})$*. Here,* C *is a constant depending only on* m *and* n*.*

Thus, we have carried out the goal of this section.

26. Computing CZ Cubes

In this section, we write c, C, C', etc. to denote constants depending only on m and n. Our goal here is to establish the following result.

CZ Computation Lemma. *After performing one-time work at most* $CN \log N$*, and using storage at most* CN*, we can answer queries in* $C \log N$ *time as follows:*

Given a subset $\mathcal{A} \subseteq \mathcal{M}$ *and a point* $x \in \mathbb{R}^n$*, we exhibit a list of all the cubes* $Q \in CZ(\mathcal{A})$ *such that* $(1 + c_G)Q$ *contains* x.

Proof. We combine the results of several previous sections.

We begin by computing the CZ cubes containing the points of E. Recall from Section 24 (specifically, Lemma 2 in that section), that we can compute and store numbers $\underline{\delta}_{CZ}(x, \mathcal{A})$ for all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$, such that

$$(1) \qquad c\underline{\delta}_{CZ}(x, \mathcal{A}) \leq \delta_{CZ}(x, \mathcal{A}) \leq C\underline{\delta}_{CZ}(x, \mathcal{A}) \text{ for all } x \in E, \mathcal{A} \subseteq \mathcal{M}.$$

The work and storage to produce all the $\underline{\delta}_{CZ}(x, \mathcal{A})$ are at most $CN \log N$ and CN, respectively. We recall that, for each $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathcal{M}$, we have

$$(2) \qquad \delta_{CZ}(x, \mathcal{A}) = \delta_Q \text{ for the } Q \in CZ(\mathcal{A}) \text{ that contains } x.$$

Next, we perform the one-time work associated with Theorem BBD1 in Section 23, as well as that associated with Lemma 1 in Section 25. This one-time work is at most $CN \log N$, and the storage it uses is at most CN.

Fix $x \in E$ and $\mathcal{A} \subseteq \mathcal{M}$. There are at most C' dyadic cubes Q such that

$$(3) \qquad Q \ni x \text{ and } c\underline{\delta}_{CZ}(x, \mathcal{A}) \leq \delta_Q \leq C\underline{\delta}_{CZ}(x, \mathcal{A}).$$

Among these is the $Q \in CZ(\mathcal{A})$ that contains x, as we see from (1), (2). According to Lemma 1 in Section 25, we can test each dyadic cube Q satisfying (3), to decide whether $Q \in CZ(\mathcal{A})$. The one and only survivor will be the cube $Q \in CZ(\mathcal{A})$ containing x, and the total work for the testing is at most $C \log N$.

Looping over all $x \in E$ and $\mathcal{A} \subseteq \mathcal{M}$, we compute (and store), for each such x and $\mathcal{A}$, the cube $Q \in CZ(\mathcal{A})$ containing x. The total work needed is at most $CN \log N$, and the storage needed is at most CN.

Thus, we may suppose that we have precomputed the cube $Q \in CZ(\mathcal{A})$ containing x and the number $\delta_{CZ}(x, \mathcal{A}) = \delta_Q$ for each $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$.

Next, we drop our assumption that $x \in E$. Let $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq M$ be given. We explain how to compute the $CZ(A)$ cube that contains x. Using Theorem BBD1 (from Section 23), we compute, with work at most $C \log N$, a point $y \in E$ such that $|x - y| \leq 2\mathsf{dist}(x, E)$. If $x = y$, then $x \in E$ and the cube $Q \in CZ(\mathcal{A})$ containing x has already been computed and stored in the one-time work. In that case, we may simply retrieve from memory the cube

$Q \in CZ(\mathcal{A})$ that contains x. Assume now that $x \neq y$. We look up the value of $\delta_{CZ}(y, \mathcal{A})$, which has been precomputed above.

We are in position to invoke Lemmas 2, 3, 4 from Section 22. With C_1 as in those Lemmas, we obtain the following results:

(4) If $\delta_{CZ}(y,\mathcal{A}) > C_1|x-y|$, then $\frac{1}{2}\,\delta_{CZ}(y,\mathcal{A}) \le \delta_{CZ}(x,\mathcal{A}) \le 2\,\delta_{CZ}(y,\mathcal{A})$.

(5) If $\delta_{CZ}(y,\mathcal{A}) \le 2C_1|x-y|$ and $0 < |x-y| \le A_2^{-1}$, then $c|x-y| \le \delta_{CZ}(x,\mathcal{A}) \le C|x-y|$.

(6) If $|x-y| \ge \frac{1}{2}A_2^{-1}$, then $cA_2^{-1} \le \delta_{CZ}(x,\mathcal{A}) \le A_2^{-1}$.

At least one of (4), (5), (6) applies, since $x \neq y$. Hence, with work at most C, we can compute a number $\underline{\delta}_{CZ}(x,\mathcal{A})$, such that

(7) $c\underline{\delta}_{CZ}(x,\mathcal{A}) \le \delta_{CZ}(x,\mathcal{A}) \le C\,\underline{\delta}_{CZ}(x,\mathcal{A})$.

There are at most C' dyadic cubes Q such that

(8) $Q \ni x$ and $c\underline{\delta}_{CZ}(x,\mathcal{A}) \le \delta_Q \le C\,\underline{\delta}_{CZ}(x,\mathcal{A})$.

Among these is the cube $Q \in CZ(\mathcal{A})$ containing x, as we see from (7) and (2). Using Lemma 1 from Section 25, we can test each Q satisfying (8), to decide whether $Q \in CZ(\mathcal{A})$. The one and only survivor will be the cube $Q \in CZ(\mathcal{A})$ containing x. The work of all the testing is at most $C \log N$.

Thus, given $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathcal{M}$, we can find, with work at most $C \log N$, the cube $Q \in CZ(\mathcal{A})$ containing x.

Finally, let $x \in \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathcal{M}$ be given, as before. As above, we find the cube $Q \in CZ(\mathcal{A})$ containing x. Now suppose that $Q' \in CZ(\mathcal{A})$, with $(1+c_G)Q' \ni x$. According to Lemma 2 ("Good Geometry") in Section 21, we must have $\frac{1}{2}\,\delta_Q \le \delta_{Q'} \le 2\delta_Q$. Consequently, to exhibit all the $Q' \in CZ(\mathcal{A})$ such that $(1+c_G)Q' \ni x$, it is enough to search among the dyadic cubes Q' such that

(9) $(1+c_G)Q' \ni x$, and $\frac{1}{2}\,\delta_Q \le \delta_{Q'} \le 2\delta_Q$.

There are at most C such dyadic cubes Q'. Using Lemma 1 in Section 25, we can test each of these Q', to see whether $Q' \in CZ(\mathcal{A})$.

Thus, we can output a list of all the $Q' \in CZ(\mathcal{A})$ such that $(1+c_G)Q' \ni x$. The work of all the testing is at most $C \log N$.

The proof of the CZ Computation Lemma is complete. ■

Remark. For $x = (x_1, ..., x_n) \in \mathbb{R}^n$ we write $|x|_{\ell_\infty} = \max_i |x_i|$. Recall Theorem BBD1 from Section 23. Suppose we replace in that theorem the Euclidean norm with the ℓ_∞ norm. That is, suppose that we modify Theorem BBD1, such that the answer to a query would now consist of two distinct points $\tilde{x}_1, \tilde{x}_2 \in E$, with $|x-\tilde{x}_1|_{\ell_\infty} \le 2\min_{y\in E}|x-y|_{\ell_\infty}$, and similarly, $|x-\tilde{x}_2|_{\ell_\infty}$ satisfies the obvious ℓ_∞ analog of the condition from Theorem BBD1. Then it is straightforward to adapt the proof of the CZ computation lemma, including the results it relies on from the preceding sections, to the case where Theorem BBD1 uses the ℓ_∞ metric. The key observation is that $|x|_{\ell_\infty} \leq |x| \leq \sqrt{n}|x|_{\ell_\infty}$ for all $x \in \mathbb{R}^n$, and that this $\sqrt{n}$ factor does not matter much, since our constants are allowed to depend on the dimension. We omit the details. This remark will be used only in the Appendix.

27. Finding Representatives

The goal of this section is to give an algorithm to produce a point in $E \cap Q$, where Q is a given dyadic cube satisfying $E \cap Q \neq \emptyset$. Recall that we can compute $\#(E \cap Q)$ by algorithm $\mathsf{RCZ1}$ from Section 25.

We write c, C, C', etc. here to denote constants depending only on the dimension n, and we write $X = O(Y)$ to indicate that $|X| \leq CY$.

We proceed as in Section 25, using the BBD tree T. We retain the notation of that section. Recall (from Section 23) that each node $A \in T$ is marked to indicate whether $A \cap E$ is empty or not. Additionally, recall that with any node $A \in T$ we store a "representative" $x_A \in E$ that satisfies $x_A \in E \cap A$ in case $E \cap A \neq \emptyset$. We will use the x_A below.

Algorithm REP0: Given a node $A \in T$ other than the root, and a dyadic cuboid $\widehat{Q} \subseteq \text{hull}\ (A)$ such that

(0) $\widehat{Q} \cap A \cap E \neq \emptyset$,

we produce a "representative" $x_{\widehat{Q},A} \in \widehat{Q} \cap A \cap E$.

Explanation: Note that our algorithm does not check whether (0) holds; it simply runs, exhibiting a representative in case (0) holds, and doing who-knows-what otherwise.

If A is a leaf, then $A \cap E$ is either empty (which cannot occur here, thanks to (0)), or else the singleton $\{x_A\}$. Hence, we may simply return $x_{\widehat{Q},A} = x_A$ in case A is a leaf.

Suppose A is an internal node other than the root. We partition the set of children of A into subsets $X_{\mathsf{in}}, X_{\mathsf{out}}, X_{\mathsf{hard}}$, as in Proposition 1 in Section 25. According to Proposition 1, we have either

(1) X_{hard} is empty, and $\widehat{Q} \cap A$ is the union of the A' in X_{in}, or else

(2) $X_{\mathsf{hard}} = \{A''\}$ for a node A'' (a child of A), and $\widehat{Q} \cap A$ is the union of $\widehat{Q} \cap A''$ with the nodes A' in X_{in}.

We check whether there exists an $A' \in X_{\mathsf{in}}$ with $E \cap A' \neq \emptyset$. If so, then we simply return $x_{\widehat{Q},A} = x_{A'}$ for such an A'. We then have $x_{\widehat{Q},A} \in E \cap A' = \widehat{Q} \cap A' \cap E \subseteq \widehat{Q} \cap A \cap E$, thanks to the defining properties of $x_{A'}$, A', and X_{in}.

Otherwise (that is, if $E \cap A' = \emptyset$ for all $A' \in X_{\mathsf{in}}$) then, by (0), we cannot be in case (1) above; we must be in case (2), and, moreover, $\widehat{Q} \cap E \cap A = \widehat{Q} \cap E \cap A'' \neq \emptyset$. In this case, since $A'' \in X_{\mathsf{hard}}$, we have $\widehat{Q} \subseteq$ hull (A''), by conclusion (c) of Proposition 1 in Section 25. Hence, we may find a point $x_{\widehat{Q},A} \in \widehat{Q} \cap E \cap A$ by recursively calling Algorithm REP0 for the node A'' and the cuboid $\widehat{Q}$.

This concludes our description of Algorithm REP0. Since A'' is a child of A above, and since the BBD tree has height $O(\log N)$, it follows that Algorithm REP0 terminates, and that the work of the algorithm is $O(\log N)$, once we have constructed the BBD tree.

Algorithm REP1: Given a dyadic cuboid $\widehat{Q}$, we decide whether $E \cap \widehat{Q} = \emptyset$; and if $E \cap \widehat{Q} \neq \emptyset$, then we exhibit a "representative" $x_{\widehat{Q}} \in E \cap \widehat{Q}$.

Explanation: The root of the BBD tree T is a disjoint union of dyadic cuboids $A^1, ..., A^L$ with $L \leq 2^n$. The set E is contained in $\bigcup_{i=1}^{L} A^i$. For each dyadic cuboid A^i, we detect whether A^i intersects $\widehat{Q}$. Since A^i and $\widehat{Q}$ are dyadic cuboids, then also $\widehat{Q} \cap A^i$ is a dyadic cuboid, whenever non-empty. For each i such that $\widehat{Q} \cap A^i$ is non-empty, we apply Algorithm RCZ1 for the dyadic cuboid $\widehat{Q} \cap A^i$ to compute $\#(E \cap \widehat{Q} \cap A^i)$. If $\#(E \cap \widehat{Q} \cap A^i) = 0$ for all i, we conclude that $E \cap \widehat{Q} = \emptyset$. Otherwise, for some i we have $E \cap \widehat{Q} \cap A^i \neq \emptyset$ with $\widehat{Q} \cap A^i$ being a dyadic cuboid contained in $A^i = \text{hull}(A^i)$. Hence, we may exhibit a point in $E \cap \widehat{Q}$ by running Algorithm REP0 for the node $A^i \in T$ and the dyadic cuboid $\widehat{Q} \cap A^i$. After one-time work $O(N \log N)$ with storage $O(N)$, Algorithm REP1 requires work $O(\log N)$.

Let Q be a dyadic cube. Then Q^* is a disjoint union of 5^n obvious dyadic cubes, and Q^{**} is a disjoint union of 25^n obvious dyadic cubes. Hence, by applying Algorithm REP1, we can perform the following computations.

Algorithm Is-Cube-Empty(Q): Given a dyadic cube Q, we decide whether $E \cap Q^{**} = \emptyset$. We return "yes" if $E \cap Q^{**} = \emptyset$ and "no" otherwise.

Algorithm Find-Representative(Q): Given a dyadic cube Q such that $E \cap Q^{**} \neq \emptyset$, we return a point $x_Q \in E \cap Q^{**}$, with the property that $x_Q \in E \cap Q^*$ if $E \cap Q^* \neq \emptyset$.

The algorithm Find-Representative(Q) is guaranteed to function properly only when its input Q is a dyadic cube such that $E \cap Q^{**} \neq \emptyset$. We make no claim regarding Find-Representative(Q) when its input fails to be a dyadic cube with $E \cap Q^{**} \neq \emptyset$. After one-time work $O(N \log N)$ with storage $O(N)$, the execution of the algorithms Is-Cube-Empty(Q) and Find-Representative(Q) requires work $O(\log N)$.

28. Partitions of Unity

In this section, C, c stand for constants depending only on m and n. Recall that $J_x^+(F)$ denotes the m-jet of the function $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ at the point $x \in \mathbb{R}^n$. Let Q be a dyadic cube in $\mathbb{R}^n$. Let $\tilde{\theta}_Q \in C^m(\mathbb{R}^n)$ be a function such that

(1) $0 \le \tilde{\theta}_Q \le 1$ on $\mathbb{R}^n$,

(2) $\tilde{\theta}_Q \ge c$ on Q,

(3) $\tilde{\theta}_Q = 0$ outside $(1 + \frac{c_G}{2})Q$,

(4) $|\partial^\beta \tilde{\theta}_Q(x)| \le C\delta_Q^{-|\beta|}$ for $x \in \mathbb{R}^n, |\beta| \le m$.

It is easy to satisfy conditions (1),..., (4), e.g., by taking $\tilde{\theta}_Q$ to be an appropriate spline. (See, e.g., (3) of Section 54.) Furthermore, we assume that $\tilde{\theta}_Q$ is picked so that the following query can be answered in work at most C:

Algorithm PU1: ("Find jet of $\tilde{\theta}_Q$") Given a dyadic cube Q, and a point $x \in \mathbb{R}^n$, compute the m-jet $J_x^+(\tilde{\theta}_Q)$.

Given $\mathcal{A} \subset \mathcal{M}$, the Calderón-Zygmund decomposition $CZ(\mathcal{A})$ is a cover of the entire $\mathbb{R}^n$. Consequently (2) implies that

(5) $$\sum_{Q \in CZ(\mathcal{A})} \tilde{\theta}_Q \ge c \text{ on } \mathbb{R}^n.$$

For any $x \in \mathbb{R}^n$, there are at most C cubes $Q \in CZ(\mathcal{A})$ such that $x \in \mathrm{Supp}(\tilde{\theta}_Q)$, according to (3) and to the Corollary to Lemma 2 in Section 21. We conclude that the left-hand side of (5) is finite everywhere. For $Q \in CZ(\mathcal{A})$, we define

$$\theta_Q^{\mathcal{A}} = \tilde{\theta}_Q \Big/ \sum_{\hat{Q} \in CZ(\mathcal{A})} \tilde{\theta}_{\hat{Q}}.$$

The function $\theta_Q^{\mathcal{A}} \in C^m(\mathbb{R}^n)$ is well defined, non-negative and finite by (5), and

(6) $\theta_Q^{\mathcal{A}}(x) = 0$ for $x \notin (1 + c_G/2)Q$.

We also have

(7) $$\sum_{Q \in CZ(\mathcal{A})} \theta_Q^{\mathcal{A}} = 1 \text{ on } \mathbb{R}^n.$$

Therefore, the collection $\theta_Q^{\mathcal{A}}$ $(Q \in CZ(\mathcal{A}))$ constitutes a partition of unity on $\mathbb{R}^n$. Let $\underline{x} \in \mathbb{R}^n$, and let $Q \in CZ(\mathcal{A})$ be such that $\underline{x} \in (1 + c_G)Q$.

According to Lemma 2 from Section 21,

(8) If $\widehat{Q} \in CZ(\mathcal{A}), \underline{x} \in (1 + c_G)\widehat{Q}$ then necessarily $\frac{1}{2}\delta_Q \le \delta_{\widehat{Q}} \le 2\delta_Q$.

Recall that by the Corollary to Lemma 2 in Section 21, there are at most C cubes $\widehat{Q} \in CZ(\mathcal{A})$ such that $\underline{x} \in (1 + c_G)\widehat{Q}$. We conclude from (3), (4) and (8), that for any $|\beta| \le m$,

$$\left|\partial^\beta \left[\sum_{\widehat{Q} \in CZ(\mathcal{A})} \tilde{\theta}_{\widehat{Q}}\right](\underline{x})\right| \le \sum_{\substack{\widehat{Q} \in CZ(\mathcal{A}) \\ \underline{x} \in (1+c_G)\widehat{Q}}} C\delta_{\widehat{Q}}^{-|\beta|} < C'\delta_Q^{-|\beta|}. \tag{9}$$

Combining (9) with (5), we see that for any $Q \in CZ(\mathcal{A})$,

$$\left|\partial^\beta \theta_Q^{\mathcal{A}}(\underline{x})\right| \le C'\delta_Q^{-|\beta|} \text{ for all } \underline{x} \in \mathbb{R}^n,\ |\beta| \le m. \tag{10}$$

Note that (6), (7) and (10) are the standard properties of partitions of unity that are usually used in C^m-extension problems.

Algorithm PU2: ("Find jet of $\theta_Q^{\mathcal{A}}$") Given $\mathcal{A} \subset \mathcal{M}$, a dyadic cube $Q \in CZ(\mathcal{A})$ and a point $x \in \mathbb{R}^n$, compute $J_x^+(\theta_Q^{\mathcal{A}})$.

Explanation: First, we apply the *CZ computation lemma* from Section 26. According to that lemma, presuming a one-time work, we can produce in $C \log N$ time, the list of all cubes $\widehat{Q} \in CZ(\mathcal{A})$ such that $x \in (1 + c_G)\widehat{Q}$. Denote this list by L. By the Corollary to Lemma 2 in Section 21, $\sharp(L) < C$. According to (3),

$$J_x^+\Bigg(\sum_{\widehat{Q} \in CZ(\mathcal{A})} \tilde{\theta}_{\widehat{Q}}\Bigg) = \sum_{\widehat{Q} \in L} J_x^+(\tilde{\theta}_{\widehat{Q}}).$$

Thus, using Algorithm PU1, we can compute both $J_x^+(\tilde{\theta}_Q)$ and

$$J_x^+\Bigg(\sum_{\widehat{Q} \in CZ(\mathcal{A})} \tilde{\theta}_{\widehat{Q}}\Bigg)$$

within C computer operations. From these two m-jets, we can read off $J_x^+(\theta_Q^{\mathcal{A}})$. This algorithm uses at most $C \log N$ work and storage. The one-time work required is $CN \log N$ operations and CN storage. Here, C is a constant depending only on m and n.

In the special case $\mathcal{A} = \emptyset$, we can produce the above list L within C computer operations, thanks to Lemma 5 in Section 21. Hence, it takes only C operations to execute Algorithm PU2 when $\mathcal{A} = \emptyset$.

Chapter IV - Main Algorithm

29. The Main Algorithm and the Main Lemma

In this section we present the main procedure of our algorithm. Recall that $J^+_x(F)$ denotes the m-jet of the function F at the point x and that $\mathcal{P}^+$ is the space of all polynomials of degree m on $\mathbb{R}^n$. Recall that for two polynomials $P, Q \in \mathcal{P}^+$ and $x \in \mathbb{R}^n$, we denote by $P \odot^+_x Q$ the unique polynomial in $\mathcal{P}^+$ for which $\partial^\beta(P \odot^+_x Q - PQ)(x) = 0$ for $|\beta| \leq m$. In this section we denote by C, C' constants depending only on m and n.

In Section 14, we introduced an order relation $<$ on subsets of $\mathcal{M}$, the set of multi-indices of order at most $m-1$. The minimal subset of $\mathcal{M}$ under $<$ is $\mathcal{M}$ itself. For any proper subset $\mathcal{A} \subset \mathcal{M}$, recall that we write $\mathcal{A}^-$ to denote the predecessor of $\mathcal{A}$ under the order $<$.

Next, we will present a procedure for the computation of a certain polynomial, to be denoted by $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$. A standard convention in computer programming, is that text between /* . . . */ is not part of the actual algorithm, but rather serves to ease the reading of the algorithm.

The Main Algorithm Procedure $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$.

```
/* Returns a polynomial in $\mathcal{P}^+$, to be viewed as a jet at $\underline{x}$.
  Defined for $\mathcal{A}_0 \subset \mathcal{M}, Q_0 \in CZ(\mathcal{A}_0), x_0 \in E \cap Q_0^{**}, P_0 \in \mathcal{P}, \underline{x} \in (1+c_G)Q_0$. */
 Line 1:  If $\mathcal{A}_0 = \mathcal{M}$ then define $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0) := P_0$, else
 Line 2:  { Let $\mathcal{A}'$ be the least $\mathcal{A} \subset \mathcal{M}$ such that $Q_0 \in CZ(\mathcal{A}')$.
 Line 3:  If $\mathcal{A}' < \mathcal{A}_0$, then define $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0) := f_{\underline{x}}(\mathcal{A}', Q_0, x_0, P_0)$,
 Line 4:  else
 Line 5:    { Produce a list $Q_1, ..., Q_{k_{max}}$ of all the cubes
 Line 6:        $Q \in CZ(\mathcal{A}_0^-)$ such that $\underline{x} \in (1+c_G)Q$.
 Line 7:        For each $k = 1, ..., k_{max}$ do the following:
 Line 8:          { If $E \cap Q_k^{**} = \emptyset$, then set $f_k := P_0$, else
 Line 9:            { If $x_0 \in Q_k^*$, then set $x_k := x_0$ and $P_k := P_0$, else
 Line 10:              { Define $x_k := \text{Find-Representative}(Q_k)$.
 Line 11:                Define $P_k := \text{Find-Neighbor}(P_0, \mathcal{A}_0, x_k)$.
 Line 12:              } /* Now we have found $x_k, P_k$ in all cases */
 Line 13:              Define $f_k := f_{\underline{x}}(\mathcal{A}_0^-, Q_k, x_k, P_k)$.
 Line 14:            } /* Now we have found $f_k$ in all cases */
 Line 15:          } /* End of the k-loop starting in line 7 */
 Line 16:         Define $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0) := \sum_{k=1}^{k_{max}} J^+_{\underline{x}}\left(\theta^{\mathcal{A}_0^-}_{Q_k}\right) \odot^+_{\underline{x}} f_k$.
 Line 17:     }
 Line 18:   }
```

Thanks to the algorithms from the previous sections, we can carry out the above algorithm for computing $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$. Let us elaborate on the execution of the *Main Algorithm.* Recall that the letters $C, C', \tilde{C}$ stand for various constants depending only on m and n.

We can execute `Line 2`, thanks to Lemma 1 from Section 25. The amount of work needed here is $C \log N$, presuming one-time work of $CN \log N$ time and CN storage.

We can execute `Line 3` since, recursively, we can evaluate $f_{\underline{x}}(\mathcal{A}', Q_0, x_0, P_0)$ when $\mathcal{A}' < \mathcal{A}_0$ and $Q_0 \in CZ(\mathcal{A}'), x_0 \in E \cap Q_0^{**}, P_0 \in \mathcal{P}, \underline{x} \in (1 + c_G)Q_0$.

We can execute `Lines 5-6` according to the *CZ Computation Lemma* from Section 26. The work needed here is $C \log N$, presuming one-time work of time $CN \log N$ and storage CN.

Note that $k_{max} \leq C$ according to the Corollary to Lemma 2 from Section 21. Hence the loop in `Lines 8-15` is executed at most a constant number of times.

We can execute `Lines 8-10` by applying the algorithms Is-Cube-Empty and Find-Representative from Section 27. We need $C \log N$ operations for the execution of `Lines 8-10`, presuming the standard $CN \log N$ one-time work.

Regarding `Line 11`, the algorithm Find-Neighbor is discussed in Section 15, and requires C operations, given $CN \log N$ one-time work.

We can execute `Line 13` since, recursively, we can evaluate $f_{\underline{x}}(\mathcal{A}_0^-, Q_k, x_k, P_k)$ as $\mathcal{A}_0^- < \mathcal{A}_0$ and $Q_k \in CZ(\mathcal{A}_0^-), x_k \in E \cap Q_k^{**}, P_k \in \mathcal{P}, \underline{x} \in (1 + c_G)Q_k$.

We can execute `Line 16` thanks to Algorithm PU2 from Section 28. This takes $C \log N$ computer operations. (As a matter of fact, C computer operations suffice here as the cubes $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ with $\underline{x} \in (1 + c_G)\widehat{Q}$ were already computed.)

By an easy induction on $\mathcal{A}$ (with respect to our order relation $<$), it follows that the number of operations needed to execute $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$, once we have done the one-time work, is at most $C \log N$. Let us summarize the above discussion.

Proposition. *For any $\mathcal{A}_0 \subset \mathcal{M}, Q_0 \in CZ(\mathcal{A}_0), x_0 \in E \cap Q_0^{**}, P_0 \in \mathcal{P}, \underline{x} \in (1 + c_G)Q_0$, we can compute the polynomial $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$ with work at most $C \log N$, given that we have previously done one-time work of $CN \log N$ operations and CN storage.*

So far we have shown that the procedure for the computation of $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$ is efficient. Next, we explain why this procedure may actually be useful. The properties of $f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$ are stated in the following lemma, whose proof occupies Sections 30,..., 33. Recall that for $x \in \mathbb{R}^n, \delta > 0$, we have set

$$B^+(x, \delta) = \{P \in \mathcal{P}^+ : |\partial^\beta P(x)| \leq \delta^{m-|\beta|} \text{ for } |\beta| \leq m\}.$$

Recall from Section 10 the basic blobs $\Gamma(x, \ell)$ $(x \in E, \ell \geq 0)$, and recall the constants $\ell(\mathcal{A}_0)$ and $A_3(\mathcal{A}_0)$ from Section 14 and Section 17, respectively.

Let $\mathcal{A}_0 \subset \mathcal{M}$ be a given subset of $\mathcal{M}$.

Main Lemma for $\mathcal{A}_0$:

Suppose that

(1) $Q_0 \in CZ(\mathcal{A}_0)$,

(2) $x_0 \in E \cap Q_0^{**}$ *with* $x_0 \in E \cap Q_0^*$ *when* $E \cap Q_0^* \neq \emptyset$,

(3) $M_0 > 0$,

(4) $P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0)$.

Then, there exists $F \in C^m((1 + c_G)Q_0)$*, with the following properties:*

(5) $J_x^+(F - P_0) \in A_3(\mathcal{A}_0) \cdot M_0 \cdot B^+(x, \delta_{Q_0})$ *for all* $x \in (1 + c_G)Q_0$.

(6) $J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A}_0) \cdot M_0)$ *for all* $x \in E \cap (1 + c_G)Q_0$.

(7) $J_x^+(F) = f_{\underline{x}}(\mathcal{A}_0, Q_0, x_0, P_0)$ *for all* $x \in (1 + c_G)Q_0$.

(8) *If* $x_0 \in (1 + c_G)Q_0$*, then also* $J_{x_0}(F) = P_0$.

Chapter V - Proofs

30. Preparation for the Proof: Collections of Polynomials

This is the first in a sequence of four sections that are dedicated to the proof of the *Main Lemma* from the preceding section. In this section we collect some results, to be used in the next sections, on certain sets of polynomials. In these four sections, the letters $c, C, c', \tilde{C}$, etc. denote some positive constants depending only on m and n; also, $p, \bar{p}, \tilde{p}$, etc. will denote positive integer constants depending only on m and n. The values of these constants are not necessarily the same in different appearances.

We will work in the space $\mathbb{R}^{\sharp(\mathcal{A})}$, for non-empty subsets $\mathcal{A} \subseteq \mathcal{M}$. For $a \in \mathbb{R}^{\sharp(\mathcal{A})}$ we use $a = (a_\alpha)_{\alpha \in \mathcal{A}}$ as coordinates in $\mathbb{R}^{\sharp(\mathcal{A})}$.

For $x \in \mathbb{R}^n$ and $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, let $\pi_{\mathcal{A},x} : \mathcal{P} \to \mathbb{R}^{\sharp(\mathcal{A})}$ be the following linear map:

(1) $\pi_{\mathcal{A},x}(P) = (\partial^\alpha P(x))_{\alpha \in \mathcal{A}}$.

For $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}, \delta > 0$ we also set

(2) $$B_{\mathcal{A}}(\delta) = \left\{ (a_\alpha)_{\alpha \in \mathcal{A}} : \sum_{\alpha \in \mathcal{A}} \delta^{|\alpha| - m} |a_\alpha| \leq 1 \right\} \subset \mathbb{R}^{\sharp(\mathcal{A})}.$$

Recall the definition of $\mathsf{B}(x,\delta)$ from Section 1. It is straightforward to verify that for any $x \in \mathbb{R}^n$,

(3) $\mathsf{B}_{\mathcal{A}}(\delta)$ is C-equivalent to $\pi_{\mathcal{A},x}\{\mathsf{B}(x,\delta)\}$.

Let $x \in \mathbb{R}^n, P \in \mathcal{P}, \mathcal{A} \subseteq \mathcal{M}$. If there exists $\alpha \in \mathcal{A}$ with $\partial^\alpha P(x) \neq 0$, then we set,

(4) $\alpha_{\mathcal{A},x}(P) = \max\{\alpha \in \mathcal{A} : \partial^\alpha P(x) \neq 0\}$,

where the maximum in (4) is taken with respect to our order $<$ on multi-indices (see Section 14). In the case where $\partial^\alpha P(x) = 0$ for all $\alpha \in \mathcal{A}$, we will set $\alpha_{\mathcal{A},x}(P)$ to be the zero multi-index, the minimal element in $\mathcal{M}$. Thus $\alpha_{\mathcal{A},x}(P)$ is defined in all cases. For $x \in \mathbb{R}^n, \emptyset \neq \mathcal{A} \subseteq \mathcal{M}$ and $\delta > 0$ denote

(5) $\mathcal{R}_{\mathcal{A}}(x,\delta) = \left\{P \in \mathcal{P} : \forall \beta \geq \alpha_{\mathcal{A},x}(P),\ |\partial^\beta P(x)| \leq \delta^{m-|\beta|}\right\}$.

The set $\mathcal{R}_{\mathcal{A}}(x,\delta)$ is centrally-symmetric, but it is not necessarily convex. It clearly satisfies

(6) $\mathsf{B}(x,\delta) \subseteq \mathcal{R}_{\mathcal{A}}(x,\delta)$,

for any non-empty $\mathcal{A} \subseteq \mathcal{M}$.

Lemma 1. *Let $\Omega \subset \mathcal{P}$ be a centrally-symmetric convex set, $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, $x \in \mathbb{R}^n$, $\delta > 0, K \geq 1$. Then the following are equivalent*:

(A) $\mathsf{B}_{\mathcal{A}}(\delta) \subseteq K\pi_{\mathcal{A},x}\{\Omega \cap \mathcal{R}_{\mathcal{A}}(x,\delta)\}$.

(B) *There exist polynomials $\{P_\alpha\}_{\alpha\in\mathcal{A}}$ with the following properties: For any $\alpha \in \mathcal{A}$,*

- (i) $\partial^\beta P_\alpha(x) = \delta_{\alpha,\beta}$ *for any* $\beta \in \mathcal{A}$,
- (ii) $|\partial^\beta P_\alpha(x)| \leq K\delta^{|\alpha|-|\beta|}$ *for any* $\beta \in \mathcal{M}$ *with* $\beta \geq \alpha$,
- (iii) $\delta^{m-|\alpha|}P_\alpha \in K\Omega$.

Proof. For $\alpha\in\mathcal{A}$, denote $a_\delta(\alpha) = (\delta^{m-|\alpha|}\delta_{\alpha,\beta})_{\beta\in\mathcal{A}} \in \mathbb{R}^{\sharp(\mathcal{A})}$. According to (2),

(7) $\mathsf{B}_{\mathcal{A}}(\delta) = \mathsf{conv}\{\pm a_\delta(\alpha) : \alpha \in \mathcal{A}\}$,

where conv denotes convex hull. Suppose (A) holds. By (7), for each $\alpha \in \mathcal{A}$ there exists $P'_\alpha \in K[\Omega \cap \mathcal{R}_{\mathcal{A}}(x,\delta)]$ such that $\pi_{\mathcal{A},x}(P'_\alpha) = a_\delta(\alpha)$. That is,

$$\partial^\beta P'_\alpha(x) = \delta^{m-|\alpha|}\delta_{\alpha,\beta} \quad \text{for} \quad \beta \in \mathcal{A}.$$

In particular, $\alpha_{\mathcal{A},x}(P'_\alpha) = \alpha$. Since $P'_\alpha \in K\mathcal{R}_{\mathcal{A}}(x,\delta)$, then $|\partial^\beta P'_\alpha(x)| \leq K\delta^{m-|\beta|}$ for any $\beta \geq \alpha$. We denote $P_\alpha = \delta^{|\alpha|-m}P'_\alpha$. Then $|\partial^\beta P_\alpha(x)| \leq K\delta^{|\alpha|-|\beta|}$ for $\alpha \in \mathcal{A}, \beta \geq \alpha$, and also $\partial^\beta P_\alpha(x) = \delta_{\alpha,\beta}$ when $\alpha, \beta \in \mathcal{A}$. Additionally, $\delta^{m-|\alpha|}P_\alpha \in K\Omega$ for $\alpha \in \mathcal{A}$. Consequently, the polynomials $\{P_\alpha\}_{\alpha\in\mathcal{A}}$ satisfy (i), (ii) and (iii). Thus we proved that (A) implies (B).

To obtain the other direction, suppose that (B) holds. Then there exist polynomials P_α $(\alpha \in \mathcal{A})$ that satisfy (i), (ii) and (iii). Denote $P'_\alpha = \delta^{m-|\alpha|}P_\alpha$. Then by (i), (ii), (iii) we know that $P'_\alpha \in K[\Omega \cap \mathcal{R}_\mathcal{A}(x,\delta)]$, and (i) also implies that $\pi_{\mathcal{A},x}(P'_\alpha) = a_\delta(\alpha)$. Let $(\lambda_\alpha)_{\alpha\in\mathcal{A}}$ be any real numbers with $\sum_\alpha |\lambda_\alpha| \leq 1$. Let

$$P = \sum_{\alpha\in\mathcal{A}} \lambda_\alpha P'_\alpha.$$

If $P \equiv 0$, clearly $P \in K[\Omega \cap \mathcal{R}_\mathcal{A}(x,\delta)]$. Suppose now that $P \not\equiv 0$. Since $K\Omega$ is convex and symmetric, we know that $P \in K\Omega$. In addition, $\alpha_{\mathcal{A},x}(P) = \max\{\alpha \in \mathcal{A} : \lambda_\alpha \neq 0\}$, and for any $\beta \geq \alpha_{\mathcal{A},x}(P)$,

$$|\partial^\beta P(x)| \leq \sum_{\alpha\in\mathcal{A}} |\lambda_\alpha||\partial^\beta P'_\alpha(x)| \leq \sum_{\alpha\in\mathcal{A}} |\lambda_\alpha| K\delta^{m-|\beta|} \leq K\delta^{m-|\beta|},$$

since $P'_\alpha \in K\mathcal{R}_\mathcal{A}(x,\delta)$ and $\alpha_{\mathcal{A},x}(P'_\alpha) = \alpha$. Hence, $P \in K\mathcal{R}_\mathcal{A}(x,\delta)$. To summarize, for any real numbers $(\lambda_\alpha)_{\alpha\in\mathcal{A}}$ with $\sum_\alpha |\lambda_\alpha| \leq 1$, we have

$$\sum_{\alpha\in\mathcal{A}} \lambda_\alpha P'_\alpha \in K[\Omega \cap \mathcal{R}_\mathcal{A}(x,\delta)].$$

Therefore, $\mathit{conv}\{\pm P'_\alpha : \alpha \in \mathcal{A}\} \subseteq K[\Omega \cap \mathcal{R}_\mathcal{A}(x,\delta)]$. Projecting, we obtain that

$$\mathit{conv}\{\pm a_\delta(\alpha) : \alpha\in\mathcal{A}\} = \mathit{conv}\{\pm \pi_{\mathcal{A},x}(P'_\alpha) : \alpha\in\mathcal{A}\} \subseteq K\pi_{\mathcal{A},x}\{\Omega \cap \mathcal{R}_\mathcal{A}(x,\delta)\},$$

and (A) follows from (7). ■

Consider condition (B) from Lemma 1; when δ gets smaller, the condition just becomes easier to satisfy. Thus, if $\delta < \bar{\delta}$ and (B) holds for $\bar{\delta}$, then (B) also holds for δ. By Lemma 1, condition (A) enjoys the same property; if $\delta < \bar{\delta}$ then

(8) $$B_\mathcal{A}(\bar{\delta}) \subseteq K\pi_{\mathcal{A},x}\left\{\Omega \cap \mathcal{R}_\mathcal{A}(x,\bar{\delta})\right\} \quad \Rightarrow \quad B_\mathcal{A}(\delta) \subset K\pi_{\mathcal{A},x}\{\Omega \cap \mathcal{R}_\mathcal{A}(x,\delta)\}.$$

An important property of the sets we consider is related to scaling. Fix $\delta > 0$, and let $\tau_\delta : \mathcal{P} \to \mathcal{P}$ be the map

$$\tau_\delta(P)(x) = \delta^m P\left(\delta^{-1}x\right).$$

It is straightforward to check that for any $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$,

(9) $B(0,\delta) = \tau_\delta\{B(0,1)\}$,

(10) $\mathcal{R}_\mathcal{A}(0,\delta) = \tau_\delta\{\mathcal{R}_\mathcal{A}(0,1)\}$; and

(11) $\forall P \in \mathcal{P}, \quad \pi_{\mathcal{A},0}(\tau_\delta P) \in B_\mathcal{A}(\delta) \Leftrightarrow \pi_{\mathcal{A},0}(P) \in B_\mathcal{A}(1)$.

Also, $(\tau_\delta P) \odot_0 (\tau_\delta Q) = \delta^m \cdot \tau_\delta(P \odot_0 Q)$. From the definition (10) of Section 12, it is straightforward to conclude that $\Omega \subset \mathcal{P}$ is Whitney t-convex at 0 with Whitney constant A, if and only if $\tau_\delta \Omega$ is Whitney t-convex at 0 with Whitney constant A.

Thus, when we study $B(0,\delta), \mathcal{R}_{\mathcal{A}}(0,\delta)$ or $B_{\mathcal{A}}(\delta)$, it is usually enough to focus on the case $\delta = 1$.

Lemma 2. *Let $\Omega \subset \mathcal{P}$ be a centrally-symmetric convex set, $x \in \mathbb{R}^n$, $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, $\delta > 0$, $K \geq 1$. Assume that,*

$$B_{\mathcal{A}}(\delta) \subseteq K\pi_{\mathcal{A},x}\{\Omega \cap \mathcal{R}_{\mathcal{A}}(x,\delta)\}, \tag{12}$$

$$B_{\mathcal{A}}(\delta) \not\subset K\pi_{\mathcal{A},x}\{\Omega \cap B(x,\delta)\}. \tag{13}$$

Then there exists $\bar{\mathcal{A}} \subseteq \mathcal{M}$ such that $\bar{\mathcal{A}} < \mathcal{A}$ and

$$B_{\bar{\mathcal{A}}}(\delta) \subseteq CK^p \pi_{\bar{\mathcal{A}},x}\{\Omega \cap B(x,\delta)\}, \tag{14}$$

where $C, p > 0$ are constants that depend solely on m and n.

Proof. By translating, we may assume that $x = 0$. Furthermore, in view of the scaling relations (9), (10) and (11), it is clear that our assumptions (12), (13), the convexity of Ω and our conclusion (14) are all invariant under the scaling $P(x) \mapsto \delta^m P\left(\delta^{-1}x\right)$ $(P \in \mathcal{P})$. Therefore it is enough to treat the case $\delta = 1$.

We will split the proof into two parts. Part I of the proof uses only the assumption (12), while (13) is exploited in the second part.

<u>Part I:</u> Fix $\alpha \in \mathcal{A}$ (the set $\mathcal{A}$ is non-empty by assumption). Let $a(\alpha) \in \mathbb{R}^{\sharp(\mathcal{A})}$ be the unit vector $a(\alpha) = (\delta_{\alpha,\beta})_{\beta \in \mathcal{A}}$. Then $a(\alpha) \in B_{\mathcal{A}}(1)$, according to (2). By (12), there exists some polynomial

$$P_\alpha \in K\left[\Omega \cap \mathcal{R}_{\mathcal{A}}(0,1)\right] \tag{15}$$

such that $\pi_{\mathcal{A},0}(P_\alpha) = a(\alpha)$. That is, for any $\beta \in \mathcal{A}$,

$$\partial^\beta P_\alpha(0) = \delta_{\alpha,\beta}. \tag{16}$$

For $\beta \in \mathcal{M}$ we define

$$\|\beta\| = \sum_{j=1}^{n} (m+1)^j \left(\sum_{k=1}^{j} \beta_k \right).$$

It is trivial to verify that for $\beta, \bar{\beta} \in \mathcal{M}$, if $\bar{\beta} < \beta$, then $\|\bar{\beta}\| < \|\beta\|$. Let $\alpha \in \mathcal{A}$ and $\beta, \bar{\beta} \in \mathcal{M}$ be such that $\beta \neq \bar{\beta}$ and $\partial^\beta P_\alpha(0) \neq 0, \partial^{\bar{\beta}} P_\alpha(0) \neq 0$. Define

$$I_{\alpha,\beta,\bar{\beta}} = \left\{ k \in \mathbb{Z} : \frac{1}{2Km! \dim \mathcal{P}} < \left| \frac{2^{k\|\bar{\beta}\|} \partial^{\bar{\beta}} P_\alpha(0)}{2^{k\|\beta\|} \partial^\beta P_\alpha(0)} \right| < 2Km! \dim \mathcal{P} \right\}.$$

Since $\beta \neq \bar{\beta}$, then $\|\beta\| - \|\bar{\beta}\|$ is a non-zero integer. Consequently,

$$\sharp(I_{\alpha,\beta,\bar{\beta}}) \leq 1 + \log_2 \left[(2Km! \dim \mathcal{P})^2 \right] < C \log(K+1).$$

The number of different sets of the form $I_{\alpha,\beta,\bar{\beta}}$ is bounded by a constant depending only m and n. Hence,

$$\#\left(\bigcup_{\substack{\alpha \in \mathcal{A}, \beta \neq \bar{\beta} \in \mathcal{M} \\ \partial^{\bar{\beta}} P_\alpha(0) \neq 0, \partial^{\beta} P_\alpha(0) \neq 0}} I_{\alpha,\beta,\bar{\beta}} \right) < C' \log(K+1).$$

Therefore, there exists an integer $k_0 \leq 0$, with $0 < |k_0| \leq \lceil C' \log(K+1) \rceil + 1$, such that $k_0 \notin I_{\alpha,\beta,\bar{\beta}}$ for any relevant $\alpha, \beta, \bar{\beta}$. Denote $\lambda = 2^{k_0}$. Then,

(17) $\quad \left(\frac{c}{K}\right)^p \leq \lambda < 1$

for some constants c, p depending only on m and n. In addition, for any $\alpha \in \mathcal{A}, \beta, \bar{\beta} \in \mathcal{M}$ such that $\beta \neq \bar{\beta}, \partial^{\beta} P_\alpha(0) \neq 0, \partial^{\bar{\beta}} P_\alpha(0) \neq 0$ we have $k_0 \notin I_{\alpha,\beta,\bar{\beta}}$ and thus

(18) $\quad \left| \frac{\lambda^{\|\bar{\beta}\|} \partial^{\bar{\beta}} P_\alpha(0)}{\lambda^{\|\beta\|} \partial^{\beta} P_\alpha(0)} \right| \notin \left(\frac{1}{2Km! \dim \mathcal{P}}, 2Km! \dim \mathcal{P} \right).$

Next, for $\alpha \in \mathcal{A}$, consider the quantity

(19) $\quad M_\alpha = \max_{\beta \in \mathcal{M}} \lambda^{\|\beta\|} |\partial^{\beta} P_\alpha(0)|.$

For any $\alpha \in \mathcal{A}$ we have $\partial^{\alpha} P_\alpha(0) = 1$ by (16), and thus,

(20) $\quad M_\alpha \geq \lambda^{\|\alpha\|} |\partial^{\alpha} P_\alpha(0)| = \lambda^{\|\alpha\|} > 0.$

The numbers whose maximum is considered in (19) are well separated from one another (except for zeros); this is the content of (18). Let $\phi(\alpha) \in \mathcal{M}$ be such that $\beta = \phi(\alpha)$ achieves the maximum in (19). According to (18), for any $\beta \neq \phi(\alpha)$,

(21) $\quad (2Km! \dim \mathcal{P}) \lambda^{\|\beta\|} |\partial^{\beta} P_\alpha(0)| \leq \lambda^{\|\phi(\alpha)\|} |\partial^{\phi(\alpha)} P_\alpha(0)|.$

If $\beta \in \mathcal{A}$ but $\beta \neq \alpha$, then $\lambda^{\|\beta\|} \partial^{\beta} P_\alpha(0) = 0 < M_\alpha$, by (16) and (20). Thus the maximum in (19) cannot be obtained by $\beta \in \mathcal{A}$ with $\beta \neq \alpha$. Consequently,

(22) $\quad \phi(\alpha) \in \mathcal{A} \quad \Longrightarrow \quad \phi(\alpha) = \alpha.$

Next, we use the fact that $P_\alpha \in K\mathcal{R}_{\mathcal{A}}(0,1)$, which we know from (15). Recall the definition (5) of the set $\mathcal{R}_{\mathcal{A}}(0,1)$. From (16) we have that

(23) $\quad \alpha = \alpha_{\mathcal{A},0}(P_\alpha).$

Since $P_\alpha \in K\mathcal{R}_{\mathcal{A}}(0,1)$ then $|\partial^\beta P_\alpha(0)| \leq K$ for all $\beta > \alpha = \alpha_{\mathcal{A},0}(P_\alpha)$. Combining with (16), we find that for all $\beta > \alpha$,

$$\lambda^{\|\beta\|}|\partial^\beta P_\alpha(0)| \leq K\lambda^{\|\beta\|} < K\lambda^{\|\alpha\|}|\partial^\alpha P_\alpha(0)| \tag{24}$$

because $\|\beta\| > \|\alpha\|$ and $0 < \lambda < 1$. Suppose for a moment that $\phi(\alpha) = \beta$ for some $\beta > \alpha$. Then (24) implies that

$$\lambda^{\|\phi(\alpha)\|}|\partial^{\phi(\alpha)} P_\alpha(0)| < K\lambda^{\|\alpha\|}|\partial^\alpha P_\alpha(0)|,$$

in contradiction with (21). Thus our momentary assumption was false, and hence,

$$\forall \alpha \in \mathcal{A}, \quad \phi(\alpha) \leq \alpha. \tag{25}$$

Denote $\bar{\mathcal{A}} = \phi(\mathcal{A})$. According to (22), (25) and Lemma 1 from Section 14, we have that $\bar{\mathcal{A}} \leq \mathcal{A}$. Let $\psi : \bar{\mathcal{A}} \to \mathcal{A}$ be such that $\phi(\psi(\alpha)) = \alpha$ for all $\alpha \in \bar{\mathcal{A}}$. By (21), for any $\alpha \in \bar{\mathcal{A}}$ and $\beta \neq \alpha$,

$$\lambda^{\|\beta\|}|\partial^\beta P_{\psi(\alpha)}(0)| \leq (2m!\dim \mathcal{P})^{-1} \cdot \lambda^{\|\alpha\|}|\partial^\alpha P_{\psi(\alpha)}(0)|. \tag{26}$$

According to (20) and (17), for all $\alpha \in \bar{\mathcal{A}}$,

$$\begin{aligned}\lambda^{\|\alpha\|}|\partial^\alpha P_{\psi(\alpha)}(0)| &= M_{\psi(\alpha)} \geq \lambda^{\|\psi(\alpha)\|}|\partial^{\psi(\alpha)} P_{\psi(\alpha)}(0)| \\ &= \lambda^{\|\psi(\alpha)\|} > \lambda^{\bar{p}} > \tfrac{1}{CK^p}\end{aligned} \tag{27}$$

for some constants $C, p, \bar{p} > 0$ depending only on m and n. The left hand side of (27) is thus non-zero, and for $\alpha \in \bar{\mathcal{A}}$ we may define

$$\bar{P}_\alpha = \left(\lambda^{\|\alpha\|}\partial^\alpha P_{\psi(\alpha)}(0)\right)^{-1} \cdot P_{\psi(\alpha)}. \tag{28}$$

Then by (26) and (28),

$$\lambda^{\|\alpha\|}\partial^\alpha \bar{P}_\alpha(0) = 1, \quad \text{and for } \beta \neq \alpha, \quad \lambda^{\|\beta\|}|\partial^\beta \bar{P}_\alpha(0)| \leq (2m!\dim \mathcal{P})^{-1}. \tag{29}$$

Next, (15), (27) and (28) imply that for any $\alpha \in \bar{\mathcal{A}}$,

$$\bar{P}_\alpha \in CK^{\tilde{p}}\Omega. \tag{30}$$

Consider the matrix $A = \left(\lambda^{\|\beta\|}\partial^\beta \bar{P}_\alpha(0)\right)_{\alpha,\beta \in \bar{\mathcal{A}}}$. By (29), the matrix A has ones on the main diagonal. Furthermore, since $\sharp(\bar{\mathcal{A}}) \leq \dim \mathcal{P}$, then according to (29) the sum of the absolute values of the off-diagonal elements in any row of A does not exceed $\frac{1}{2}$. Hence A is invertible, and the norm of A^{-1} as an operator on $l_\infty(\mathbb{R}^n)$ is not larger than 2. Denote the elements of A^{-1} by $A^{-1} = (a_{\alpha,\beta})_{\alpha,\beta \in \bar{\mathcal{A}}}$. Then,

$$|a_{\alpha,\beta}| \leq 2 \quad \text{for all } \alpha, \beta \in \bar{\mathcal{A}}. \tag{31}$$

Next, we set, for $\alpha \in \bar{\mathcal{A}}$

$$(32)\qquad P'_\alpha = \lambda^{\|\alpha\|} \sum_{\gamma \in \bar{\mathcal{A}}} a_{\gamma,\alpha} \bar{P}_\gamma.$$

By the definition of the inverse matrix A^{-1}, for any $\alpha, \beta \in \bar{\mathcal{A}}$,

$$(33)\qquad \partial^\beta P'_\alpha(0) = \lambda^{\|\alpha\|-\|\beta\|} \sum_{\gamma \in \bar{\mathcal{A}}} a_{\gamma,\alpha} \lambda^{\|\beta\|} \partial^\beta \bar{P}_\gamma(0) = \delta_{\alpha,\beta}.$$

Furthermore, by (29), (31) and (32), for any $\beta \notin \bar{\mathcal{A}}$,

$$(34)\qquad |\partial^\beta P'_\alpha(0)| \le 2\lambda^{\|\alpha\|} \sum_{\gamma \in \bar{\mathcal{A}}} |\partial^\beta \bar{P}_\gamma(0)| \le 2\lambda^{\|\alpha\|} \sum_{\gamma \in \bar{\mathcal{A}}} \frac{1}{\lambda^{\|\beta\|}} \le C\lambda^{\|\alpha\|-\|\beta\|}.$$

According to (33) and (34), for any $\alpha \in \bar{\mathcal{A}}$,

$$(35)\qquad P'_\alpha \in \tfrac{C}{\lambda^c} B(0,1) \subseteq C' K^{\bar{p}} B(0,1),$$

where the second inclusion follows from (17). Recall that Ω is convex and centrally-symmetric. By (17), (30), (31) and (32), we get that for any $\alpha \in \bar{\mathcal{A}}$,

$$(36)\qquad P'_\alpha \in C K^p \Omega.$$

Since Ω and $B(0,1)$ are convex and centrally-symmetric, (35) and (36) imply that

$$\mathrm{conv}\{\pm P'_\alpha\}_{\alpha \in \bar{\mathcal{A}}} \subseteq C K^p [\Omega \cap B(0,1)].$$

Projecting, we get that

$$(37)\qquad \pi_{\bar{\mathcal{A}},0}\{\mathrm{conv}\{\pm P'_\alpha\}_{\alpha \in \bar{\mathcal{A}}}\} \subseteq C K^p \pi_{\bar{\mathcal{A}},0}\{\Omega \cap B(0,1)\}.$$

However, $B_{\bar{\mathcal{A}}}(1) = \pi_{\bar{\mathcal{A}},0}\{\mathrm{conv}\{\pm P'_\alpha\}_{\alpha \in \bar{\mathcal{A}}}\}$ by (33). Therefore (37) implies that

$$(38)\qquad B_{\bar{\mathcal{A}}}(1) \subseteq C K^p \pi_{\bar{\mathcal{A}},0}\{\Omega \cap B(0,1)\}.$$

It only remains to show that $\bar{\mathcal{A}} < \mathcal{A}$; once we have that, (38) implies the conclusion of the lemma. Note that up to now, we did not use our assumption (13). It will play a rôle in the proof that $\bar{\mathcal{A}} < \mathcal{A}$.

Part II: We begin with proving that there exists $\hat{\alpha} \in \mathcal{A}$ with $\phi(\hat{\alpha}) \notin \mathcal{A}$. Assume the opposite, i.e.,

$$(39)\qquad \phi(\alpha) \in \mathcal{A} \quad \text{for all } \alpha \in \mathcal{A}.$$

By (22) we have that $\phi(\alpha) = \alpha$ for all $\alpha \in \mathcal{A}$. Let $\alpha \in \mathcal{A}$ and $\beta < \phi(\alpha) = \alpha$. According to (21) and (16),

$$(40)\qquad \begin{aligned} \lambda^{\|\beta\|} |\partial^\beta P_\alpha(0)| &\le (2K \dim \mathcal{P})^{-1} \cdot \lambda^{\|\alpha\|} |\partial^\alpha P_\alpha(0)| \\ &= (2K \dim \mathcal{P})^{-1} \cdot \lambda^{\|\alpha\|} < K \lambda^{\|\beta\|}, \end{aligned}$$

since $K \ge 1, 0 < \lambda < 1$ and $\|\beta\| < \|\alpha\|$.

Therefore (40) implies that for any $\beta < \alpha$,

(41) $|\partial^\beta P_\alpha(0)| \leq K.$

Recall that $P_\alpha \in K\mathcal{R}_{\mathcal{A}}(0,1)$ by (15). According to (5) and (23), also for any $\beta \geq \alpha$,

(42) $|\partial^\beta P_\alpha(0)| \leq K.$

Combining (41) and (42) we get that $P_\alpha \in KB(0,1)$ for all $\alpha \in \mathcal{A}$. By (15), we conclude that

(43) $\mathrm{conv}\{\pm P_\alpha\}_{\alpha\in\mathcal{A}} \subseteq K\Omega \cap KB(0,1) = K\left[\Omega \cap B(0,1)\right].$

(Recall that $\Omega, B(0,1)$ are convex and centrally-symmetric.) However, putting $\delta = 1$ in (2) and using (16) we obtain

(44) $B_{\mathcal{A}}(1) = \pi_{\mathcal{A},0}\{\mathrm{conv}\{\pm P_\alpha\}_{\alpha\in\mathcal{A}}\}.$

According to (43) and (44) we have that

(45) $B_{\mathcal{A}}(1) \subseteq K\pi_{\mathcal{A},0}\{\Omega \cap B(0,1)\},$

in contradiction with (13). Therefore, our assumption (39) was false, and consequently there exists $\hat{\alpha} \in \mathcal{A}$ with $\phi(\hat{\alpha}) \notin \mathcal{A}$. In particular, $\phi(\hat{\alpha}) \neq \hat{\alpha}$, and hence ϕ is not the identity map. The relations (22) and (25) are exactly the assumptions of Lemma 1 from Section 14. By the conclusion of that lemma, we know that $\bar{\mathcal{A}} = \phi(\mathcal{A}) < \mathcal{A}$, as ϕ is not the identity map. This completes the proof. ■

A set $\mathcal{A} \subseteq \mathcal{M}$ is called a "monotonic set", if for any multi-indices α and β,

$$\alpha \in \mathcal{A},\ |\beta| \leq m - 1 - |\alpha| \quad \Rightarrow \quad \alpha + \beta \in \mathcal{A}.$$

Suppose $\mathcal{A} \subseteq \mathcal{M}$ is a monotonic set. The fundamental property of $\mathcal{A}$ is that for any $P \in \mathcal{P}$ and $x, y \in \mathbb{R}^n$,

(46) $\pi_{\mathcal{A},x}(P) = 0 \quad \Longrightarrow \quad \pi_{\mathcal{A},y}(P) = 0.$

Indeed, (46) follows at once, since for any $\alpha \in \mathcal{A}$,

$$\partial^\alpha P(y) = \sum_{|\beta|\leq m-1-|\alpha|} \frac{\partial^{\alpha+\beta}P(x)}{\beta!}(y-x)^\beta = 0,$$

where the sum vanishes because $\alpha + \beta$ is always in $\mathcal{A}$.

Recall that for two subsets $\Omega_1, \Omega_2 \subset \mathcal{P}$ and $x \in \mathbb{R}^n$, we denote

$$\Omega_1 \odot_x \Omega_2 = \{P \odot_x Q \ :\ P \in \Omega_1, Q \in \Omega_2\}.$$

As before, we write $\mathrm{conv}(\Omega)$ to denote the convex hull of a set $\Omega \subset \mathcal{P}$.

Lemma 3. *Let $\Omega \subseteq B(0,1)$ be a centrally-symmetric convex set, and let $K \geq 1$. Assume that $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, and that*

(47) $B_{\mathcal{A}}(1) \subseteq K\pi_{\mathcal{A},0}\{\Omega\}$.

Then there exists a monotonic set $\mathcal{A}' \subseteq \mathcal{M}$, with $\mathcal{A}' \leq \mathcal{A}$, and

$$B_{\mathcal{A}'}(1) \subseteq CK^p\,\pi_{\mathcal{A}',0}\{\mathrm{conv}\,[\Omega \odot_0 B(0,1)]\}$$

where C, p are constants depending only on m and n.

Proof. Our assumptions (47), and $\Omega \subseteq B(0,1) \subset \mathcal{R}_{\mathcal{A}}(0,1)$ imply that

(48) $B_{\mathcal{A}}(1) \subseteq K\pi_{\mathcal{A},0}\{\Omega \cap \mathcal{R}_{\mathcal{A}}(0,1)\}$.

Now, (48) is precisely the assumption (12) of Lemma 2, in the case $x = 0$, $\delta = 1$. Most of the proof of Lemma 2 used only this assumption, namely Part I of that proof. In particular, the construction of $0 < \lambda < 1$, the set $\bar{\mathcal{A}} \leq \mathcal{A}$ and the polynomials $\bar{P}_\alpha$ in the proof of Lemma 2, was based on (12) only.

We may thus repeat the reasoning from Part I of Lemma 2, based on (48). Therefore we obtain $0 < \lambda < 1$ that satisfies (17), a set $\bar{\mathcal{A}} \leq \mathcal{A}$, and polynomials $\bar{P}_\alpha$ $(\alpha \in \bar{\mathcal{A}})$ that satisfy (28),..., (30). This means that,

(49) $\frac{1}{(CK)^p} < \lambda < 1$,

(50) $\bar{P}_\alpha \in CK^p\Omega$ for any $\alpha \in \bar{\mathcal{A}}$,

and for all $\alpha \in \bar{\mathcal{A}}$,

(51) $\lambda^{\|\alpha\|}\partial^\alpha \bar{P}_\alpha(0) = 1$, and for $\beta \neq \alpha$, $\lambda^{\|\beta\|}|\partial^\beta \bar{P}_\alpha(0)| \leq \frac{1}{2m!\dim \mathcal{P}}$.

Next, we denote

$$\mathcal{A}' = \{\alpha + \gamma \,:\, \alpha \in \bar{\mathcal{A}},\ |\gamma| \leq m - 1 - |\alpha|\},$$

and let

$\alpha'_1 + \gamma_1, ..., \alpha'_{t_{max}} + \gamma_{t_{max}}$ $\quad(\alpha'_t \in \bar{\mathcal{A}}, |\gamma_t| \leq m - 1 - |\alpha'_t|$ for $t = 1, ..., t_{max})$

be an enumeration of $\mathcal{A}'$. The set $\mathcal{A}'$ is clearly monotonic, and satisfies that $\mathcal{A}' \leq \bar{\mathcal{A}}$ (since $\bar{\mathcal{A}} \subseteq \mathcal{A}'$). Since $\bar{\mathcal{A}} \leq \mathcal{A}$, then by transitivity we also have $\mathcal{A}' \leq \mathcal{A}$. For $\gamma \in \mathcal{M}$ we will consider the polynomial $x \mapsto x^\gamma$ on $\mathbb{R}^n$. With a slight abuse of notation, we denote this polynomial by x^γ; for example, we write that $x^\gamma \in \mathcal{P}$. We define polynomials P'_t for $t = 1, ..., t_{max}$ as follows:

(52) $P'_t = \frac{(\alpha'_t)!}{(\alpha'_t + \gamma_t)!} x^{\gamma_t} \odot_0 \bar{P}_{\alpha'_t}$.

The polynomial x^{γ_t} belongs to $CB(0,1)$. According to (50) and (52), we conclude that

(53) $P'_t \in CK^p[\Omega \odot_0 B(0,1)]$.

From (51) and (52) we obtain that for any $\beta \in \mathcal{M}$,

$$(54) \quad \left|\lambda^{\|\beta\|}\partial^{\beta+\gamma_t}P'_t(0) - \delta_{\beta,\alpha'_t}\right| \leq \tfrac{1}{2\dim \mathcal{P}},$$

and that

$$(55) \quad \partial^{\beta}P'_t(0) = 0 \ \text{ whenever } \beta - \gamma_t \not\in \mathcal{M}$$

(i.e., whenever $\beta-\gamma_t$ contains negative coordinates). Note that (54) and (55) together provide bounds for $\partial^{\beta}P'_t(0)$ for all $\beta \in \mathcal{M}$.

Consider now the matrix $A = \left(\lambda^{\|\alpha'_s\|}\partial^{\alpha'_s+\gamma_s}P'_t(0)\right)_{t,s=1,\ldots,t_{max}}$. According to (54) and (55), the matrix A is very close to the identity matrix; the norm of $A - Id$ on $l_\infty(\mathbb{R}^n)$ is bounded by $\frac{1}{2}$. Consequently, the matrix A is invertible, and the inverse matrix $A^{-1} = (a_{t,s})_{t,s=1,\ldots,t_{max}}$ satisfies that

$$(56) \quad |a_{t,s}| \leq 2 \quad \text{for all } t, s = 1, ..., t_{max},$$

and that for all $t, s = 1, ..., t_{max}$,

$$(57) \quad \sum_{r=1}^{t_{max}} \lambda^{\|\alpha'_s\|}\partial^{\alpha'_s+\gamma_s}P'_r(0)a_{r,t} = \delta_{s,t}.$$

Next, we define polynomials P_α for $\alpha \in \mathcal{A}'$ as follows:

$$(58) \quad P_{\alpha'_t+\gamma_t} = \lambda^{\|\alpha'_t\|}\sum_{r=1}^{t_{max}} a_{r,t}P'_r.$$

By (58) and (57), for any $\alpha, \beta \in \mathcal{A}'$, $\alpha = \alpha'_t + \gamma_t, \beta = \alpha'_s + \gamma_s$,

$$(59) \quad \partial^{\beta}P_\alpha(0) = \partial^{\alpha'_s+\gamma_s}P_{\alpha'_t+\gamma_t}(0) = \sum_{r=1}^{t_{max}} \lambda^{\|\alpha'_t\|}\partial^{\alpha'_s+\gamma_s}P'_r(0)a_{r,t} = \delta_{s,t} = \delta_{\alpha,\beta}.$$

The set $conv[\Omega \odot_0 B(0,1)]$ is convex, by definition, and it is also centrally-symmetric. Thus (49), (53), (56) and the definition (58) imply that for any $\alpha \in \mathcal{A}'$,

$$(60) \quad P_\alpha \in 2t_{max} \cdot CK^p \cdot conv\left[\Omega \odot_0 B(0,1)\right] \subseteq C'K^p conv\left[\Omega \odot_0 B(0,1)\right].$$

Combining (60) and the convexity and central-symmetry of $conv[\Omega \odot_0 B(0,1)]$ we get that

$$(61) \quad \pi_{\mathcal{A}',0}\{conv\{\pm P_\alpha\}_{\alpha\in\mathcal{A}'}\} \subseteq CK^p\pi_{\mathcal{A}',0}\{conv[\Omega \odot_0 B(0,1)]\}.$$

According to (59), we know that $B_{\mathcal{A}'}(1) = \pi_{\mathcal{A}',0}\{conv\{\pm P_\alpha\}_{\alpha\in\mathcal{A}'}\}$. Hence by (61),

$$B_{\mathcal{A}'}(1) \subseteq CK^p\pi_{\mathcal{A}',0}\{conv[\Omega \odot_0 B(0,1)]\}.$$

Since $\mathcal{A}'$ is monotonic and $\mathcal{A}' \leq \mathcal{A}$, the lemma is proven. ■

Lemma 4. *Let $\Omega \subset \mathcal{P}$ be a centrally-symmetric convex set, $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, $x \in \mathbb{R}^n$, $\delta > 0, K \geq 1$. Assume that Ω is Whitney t-convex at x with Whitney constant $A > 1$. Assume also that*

$$(62) \quad B_{\mathcal{A}}(\delta) \subseteq K\pi_{\mathcal{A},x}\{\Omega \cap B(x,\delta)\}.$$

Then, there exists $\bar{\mathcal{A}} \subseteq \mathcal{M}$, such that $\bar{\mathcal{A}}$ is monotonic, $\bar{\mathcal{A}} \leq \mathcal{A}$, and

$$(63) \quad B_{\bar{\mathcal{A}}}(\delta) \subseteq CAK^p\pi_{\bar{\mathcal{A}},x}\{\Omega \cap B(x,\delta)\},$$

where $C, p > 0$ are constants that depend solely on m and n.

Proof. By translating and rescaling, according to (9), (10), (11) and the discussion around them, we may assume that $x = 0$, $\delta = 1$. Denote $\Omega' = \Omega \cap B(0,1)$. Based on (62), we may invoke Lemma 3 for the set Ω'. By the conclusion of that lemma, we find $\bar{\mathcal{A}} \subseteq \mathcal{M}$, such that $\bar{\mathcal{A}}$ is monotonic, $\bar{\mathcal{A}} \leq \mathcal{A}$, and

$$(64) \quad B_{\bar{\mathcal{A}}}(1) \subseteq CK^p\pi_{\bar{\mathcal{A}},0}\{\mathrm{conv}[\Omega' \odot_0 B(0,1)]\}.$$

From our assumptions, the set Ω is Whitney t-convex at 0 with Whitney constant A. According to Lemma 2 from Section 12, also $\Omega' = \Omega \cap B(0,1)$ is Whitney t-convex at 0 with Whitney constant CA. This implies that

$$(65) \quad \Omega' \odot_0 B(0,1) = [\Omega' \cap B(0,1)] \odot_0 B(0,1) \subseteq CA\Omega'.$$

By using (64) and (65), we get that

$$(66) \quad B_{\bar{\mathcal{A}}}(1) \subseteq C'AK^p\pi_{\bar{\mathcal{A}},0}\{\mathrm{conv}(\Omega')\} = C'AK^p\pi_{\bar{\mathcal{A}},0}\{\Omega \cap B(0,1)\},$$

since $\mathrm{conv}(\Omega') = \Omega' = \Omega \cap B(0,1)$, and the lemma is proven. ■

We would like our treatment to include also the degenerate case where $\mathcal{A} = \emptyset$. Thus, we will also consider the ridiculous space $\mathbb{R}^{\sharp(\mathcal{A})}$ for $\mathcal{A} = \emptyset$; here the space $\mathbb{R}^{\sharp(\emptyset)}$ $(= \mathbb{R}^0)$ simply means the singleton $\{0\}$. We also define the (trivial) projection $\pi_{\emptyset,x} : \mathcal{P} \to \mathbb{R}^{\sharp(\emptyset)}$ by setting

$$\pi_{\emptyset,x}(P) = 0$$

for all $x \in \mathbb{R}^n, P \in \mathcal{P}$. Also, $B_{\emptyset}(\delta) = \{0\}$ for all $\delta > 0$.

Lemma 5. *Let $\Omega \subset \mathcal{P}$ be a centrally-symmetric convex set, $x \in \mathbb{R}^n, \mathcal{A} \subseteq \mathcal{M}$, $\delta > 0$, $K \geq 1$ be given. Assume that*

$$(67) \quad B_{\mathcal{A}}(\delta) \subseteq K\pi_{\mathcal{A},x}\{\Omega \cap B(x,\delta)\}, \quad \textit{and}$$

$$(68) \quad 0 \in \pi_{\mathcal{A},x}\left\{\Omega \setminus K^{-1}B(x,\delta)\right\}.$$

Then there exists $\bar{\mathcal{A}} \subseteq \mathcal{M}$ with $\bar{\mathcal{A}} < \mathcal{A}$ such that

$$(69) \quad B_{\bar{\mathcal{A}}}(\delta) \subseteq 2K^2\pi_{\bar{\mathcal{A}},x}\{\Omega \cap B(x,\delta)\}.$$

Proof. As before, we will translate and rescale, according to (9), (10) and (11). Thus we may assume that $x = 0, \delta = 1$. Let $P \in \Omega \setminus K^{-1}B(0,1)$ be such that $\pi_{\mathcal{A},0}(P) = 0$. The existence of such a polynomial P is guaranteed by (68). Then,

(70) $\partial^\beta P(0) = 0$ for $\beta \in \mathcal{A}$.

Let $\hat{\alpha} \in \mathcal{M}$ be chosen such that

(71) $|\partial^{\hat{\alpha}} P(0)| = \max_{\beta \in \mathcal{M}} |\partial^\beta P(0)|$.

Since $P \not\in K^{-1}B(0,1)$, necessarily $|\partial^{\hat{\alpha}} P(0)| > K^{-1} > 0$. Also, $\hat{\alpha} \not\in \mathcal{A}$ because of (70). Set

(72) $P_{\hat{\alpha}} = \frac{1}{\partial^{\hat{\alpha}} P(0)} P \in K\Omega$,

where $P_{\hat{\alpha}} \in K\Omega$ because $P \in \Omega$ and $|\frac{1}{\partial^{\hat{\alpha}} P(0)}| < K$. Denote $\bar{\mathcal{A}} = \mathcal{A} \cup \{\hat{\alpha}\}$. Then $\mathcal{A} \subset \bar{\mathcal{A}}$ and $\mathcal{A} \neq \bar{\mathcal{A}}$, hence $\bar{\mathcal{A}} < \mathcal{A}$. By (70), (72)

(73) $\partial^\beta P_{\hat{\alpha}}(0) = \delta_{\beta,\hat{\alpha}}$ for $\beta \in \bar{\mathcal{A}}$.

In addition, since $\hat{\alpha}$ was chosen to maximize in (71), we deduce from (72) that $|\partial^\beta P_{\hat{\alpha}}(0)| \leq 1$ for all $\beta \in \mathcal{M}$. Therefore,

(74) $P_{\hat{\alpha}} \in B(0,1)$.

Next, by (67) there exist polynomials

(75) $P'_\alpha \in K\,[\Omega \cap B(0,1)]$ for $\alpha \in \mathcal{A}$

with

(76) $\partial^\beta P'_\alpha(0) = \delta_{\alpha,\beta}$ for $\alpha, \beta \in \mathcal{A}$.

Since $P'_\alpha \in KB(0,1)$ by (75), then for any $\alpha \in \mathcal{A}$,

(77) $\forall \beta \in \mathcal{M},\ |\partial^\beta P'_\alpha(0)| \leq K,$ and in particular $|\partial^{\hat{\alpha}} P'_\alpha(0)| \leq K$.

Denote, for $\alpha \in \mathcal{A}$,

(78) $P_\alpha = P'_\alpha - \partial^{\hat{\alpha}} P'_\alpha(0) \cdot P_{\hat{\alpha}}$.

By (73), (76) and (78),

(79) $\partial^\beta P_\alpha(0) = \delta_{\alpha,\beta}$ for $\alpha, \beta \in \bar{\mathcal{A}}$.

According to (78), (72), (75) and (77) we know that

(80) $P_\alpha \in (K + K \cdot K)\Omega \subset 2K^2\Omega$ for $\alpha \in \bar{\mathcal{A}}$.

Additionally, for any $\alpha \in \mathcal{A}, \beta \in \mathcal{M}$, by (74), (77) and (78)

$$|\partial^\beta P_\alpha(0)| \leq |\partial^\beta P'_\alpha(0)| + |\partial^{\hat{\alpha}} P'_\alpha(0)| \cdot |\partial^\beta P_{\hat{\alpha}}(0)| \leq K + K \cdot 1 = 2K.$$

Thus $P_\alpha \in 2KB(0,1)$ for any $\alpha \in \mathcal{A}$. Together with (80) and (74), this gives

$$(81) \qquad P_\alpha \in 2K^2 \left[\Omega \cap B(0,1)\right] \quad \text{for } \alpha \in \bar{\mathcal{A}}.$$

Note that (79) implies that $B_{\bar{\mathcal{A}}}(1) = \mathrm{conv}\{\pm\pi_{\bar{\mathcal{A}},0}(P_\alpha)\}_{\alpha \in \bar{\mathcal{A}}}$. By convexity and central-symmetry, (81) gives

$$B_{\bar{\mathcal{A}}}(1) \subseteq 2K^2 \pi_{\bar{\mathcal{A}},0}\{\Omega \cap B(0,1)\},$$

which is exactly the desired inclusion (69). This finishes the proof. ■

In the proof of the next lemma we will make use of the following simple fact. Suppose K, T are centrally-symmetric, bounded convex sets in a finite dimensional vector space V. Then,

$$(82) \qquad K \subseteq T + \tfrac{1}{3}K \quad \Rightarrow \quad \tfrac{1}{2}K \subseteq T.$$

This is easily seen: The left hand side of (82) implies that for any functional $f \in V^*$ we have $\frac{2}{3}\sup_{x \in K} f(x) \leq \sup_{x \in T} f(x)$. Hence $K \subseteq \frac{3}{2}\bar{T} \subseteq 2T$, where $\bar{T}$ is the closure of T.

Lemma 6. *There exists a constant $C_0 > 1$ depending only on m and n for which the following holds: Let $\Omega \subset \mathcal{P}$ be a centrally-symmetric convex set, $x \in \mathbb{R}^n, \mathcal{A} \subseteq \mathcal{M}$, $\delta > 0, K \geq 1$. Assume that $\Omega \subseteq B(x,\delta)$, and that*

$$(83) \qquad B_{\mathcal{A}}(\delta) \subseteq K\pi_{\mathcal{A},x}\{\Omega\}.$$

Let $y \in \mathbb{R}^n$ be such that $|x-y| < \frac{\delta}{C_0 K}$. Then,

$$(84) \qquad B_{\mathcal{A}}(\delta) \subseteq 2K\pi_{\mathcal{A},y}\{\Omega\}.$$

Proof. Pick $P \in \Omega$. Then $P \in B(x,\delta)$ and hence $|\partial^\beta P(x)| \leq \delta^{m-|\beta|}$ for all $\beta \in \mathcal{M}$. By Taylor's theorem, for any $\alpha \in \mathcal{A}$,

$$(85) \qquad |\partial^\alpha P(y) - \partial^\alpha P(x)| = \left| \sum_{1 \leq |\beta| \leq m-1-|\alpha|} \frac{\partial^{\alpha+\beta}P(x)}{\beta!}(y-x)^\beta \right|$$

$$\leq C' \sum_{1 \leq |\beta| \leq m-1-|\alpha|} \delta^{m-(|\alpha|+|\beta|)}|x-y|^{|\beta|} \leq C'' \frac{|x-y|}{\delta}\delta^{m-|\alpha|},$$

since $|x-y| < \frac{\delta}{C_0 K} < \delta$. The inequality (85) implies that, for any $P \in \Omega$,

$$(86) \qquad \pi_{\mathcal{A},x}(P) - \pi_{\mathcal{A},y}(P) \in C\frac{|x-y|}{\delta}B_{\mathcal{A}}(\delta) \subseteq \frac{C}{C_0 K}B_{\mathcal{A}}(\delta).$$

We set $C_0 = 3C > 1$, where C is the constant from (86). By combining (83) with (86) we get that

$$(87) \quad \tfrac{1}{K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},x}\{\Omega\} \subseteq \pi_{\mathcal{A},y}\{\Omega\} + \tfrac{1}{3K}B_{\mathcal{A}}(\delta).$$

Since all the sets in (87) are bounded, convex and centrally-symmetric, then (82) entails

$$\frac{1}{2K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\{\Omega\},$$

and the lemma is proven. ■

31. Preparation for the Proof: Properties of the Basic Lengthscales

Recall the definition of $\sigma(x,\ell), \Gamma(x,\ell,M)$ from Section 10. Recall also Properties 0,...,4 of these blobs, from Section 13, and the definition of the constant ℓ_* from Section 14.

Properties 0,...,4 from Section 13 are the only properties of the Γ's and σ's that are relevant to the proof of the *Main Lemma*. In particular, one may replace the blobs Γ and the sets σ with any other family of blobs and sets, as long as these five properties still hold. The *Main Algorithm* and the proof of the *Main Lemma* would remain valid, even with this new family of blobs (see [16] for a different family of blobs that satisfy these crucial five properties). We will make use of $\sigma(x,\ell), \Gamma(x,\ell,M)$ for $x \in E, M > 0$ and $0 \leq \ell \leq \ell_*$. Since ℓ_* is a constant depending only on m and n, and we use ℓ only in the range $0 \leq \ell \leq \ell_*$, then we may view the constants c_ℓ, C_ℓ in Properties 0, 1, 2, 3, 4 from Section 13, as constants depending only on m and n.

Lemma 1. *There exist constants $C, C_0 > 1$ depending only on m and n for which the following holds: Let $\mathcal{A} \subseteq \mathcal{M}$, $x,y \in E$, $K \geq 1$, $1 \leq \ell \leq \ell_*$. Assume that $\delta > C_0K|x-y|$. Suppose that*

$$(1) \quad B_{\mathcal{A}}(\delta) \subseteq K\pi_{\mathcal{A},x}\{\sigma(x,\ell) \cap B(x,\delta)\}.$$

Then,

$$(2) \quad B_{\mathcal{A}}(\delta) \subseteq CK\pi_{\mathcal{A},y}\{\sigma(y,\ell-1) \cap B(y,\delta)\}.$$

Proof. We choose $C_0 > 1$ to be larger than the constant C_0 from Lemma 6 from the preceding section. Set $\Omega = \sigma(x,\ell) \cap B(x,\delta)$. The fact that $|x-y| < \frac{\delta}{C_0K}$ and (1) are the assumptions of Lemma 6 from the preceding section. By the conclusion of that lemma,

$$(3) \quad B_{\mathcal{A}}(\delta) \subseteq 2K\pi_{\mathcal{A},y}\{\Omega\} = 2K\pi_{\mathcal{A},y}\{\sigma(x,\ell) \cap B(x,\delta)\}.$$

Since $|x-y| < \frac{\delta}{C_0K} < \delta$, then the sets $B(x,\delta)$ and $B(y,\delta)$ are C-equivalent, by (3) from Section 12. Therefore (3) translates to

$$(4) \qquad \tfrac{1}{CK}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\{\sigma(x,\ell)\cap B(y,\delta)\}.$$

Next, we use Property 2 from Section 13 and the fact that $|x-y| < \frac{\delta}{C_0K}$, to obtain that

$$(5) \qquad \sigma(x,\ell) \subseteq C\left[\sigma(y,\ell-1)+B(x,y)\right] \subseteq C'\left[\sigma(y,\ell-1)+B\left(y,\tfrac{\delta}{C_0K}\right)\right].$$

Since $C_0K > 1$, the relation (2) from Section 12, gives

$$(6) \qquad B\left(y,\tfrac{\delta}{C_0K}\right) \subseteq \tfrac{1}{C_0K}B(y,\delta).$$

Thus (4), (5) and (6) imply that

$$(7) \qquad \tfrac{1}{C'K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\left\{\left[\sigma(y,\ell-1)+\tfrac{1}{C_0K}B(y,\delta)\right]\cap B(y,\delta)\right\}.$$

Recall (11) from Section 12, and note that $C_0K \geq 1$. According to (11) from Section 12 and (7),

$$(8) \qquad \tfrac{1}{C'K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\left\{\left[\sigma(y,\ell-1)\cap 2B(y,\delta)\right]+\tfrac{1}{C_0K}B(y,\delta)\right\}.$$

The sets $B_{\mathcal{A}}(\delta)$ and $\pi_{\mathcal{A},y}\{B(y,\delta)\}$ are C-equivalent, by (3) from Section 30. Then (8) translates into

$$(9) \qquad \tfrac{1}{C'K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\{[\sigma(y,\ell-1)\cap 2B(y,\delta)]\}+\tfrac{C''}{C_0K}B_{\mathcal{A}}(\delta).$$

We further stipulate that $C_0 > 3C'C''$, for C', C'' the constants from (9). Thus, (9) implies that

$$(10) \qquad \tfrac{1}{C'K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\{\sigma(y,\ell-1)\cap 2B(y,\delta)\}+\tfrac{1}{3C'K}B_{\mathcal{A}}(\delta).$$

All the involved sets are bounded, convex and centrally-symmetric. Recall the elementary fact (82) from Section 30. Therefore from (10) we deduce that

$$\tfrac{1}{2C'K}B_{\mathcal{A}}(\delta) \subseteq \pi_{\mathcal{A},y}\{\sigma(y,\ell-1)\cap 2B(y,\delta)\} \subseteq 2\pi_{\mathcal{A},y}\{\sigma(y,\ell-1)\cap B(y,\delta)\}$$

and the lemma follows, with C_0 a large enough constant depending solely on m and n. ■

Lemma 2. *Let* $\mathcal{A} \subseteq \mathcal{M}, x \in E, \delta > 0$ *and* $K_1, K_2 > 0$. *Suppose that* $0 \leq \ell \leq \ell_*$ *satisfies*

$$(11) \quad B_{\mathcal{A}}(\delta) \subseteq K_1\pi_{\mathcal{A},x}\{\sigma(x,\ell)\cap B(x,\delta)\}.$$

Let $M > 0, P \in \mathcal{P}$ *be such that*

$$(12) \quad P \in \Gamma(x,\ell,M)+K_2MB(x,\delta).$$

Then there exists $\tilde{P} \in \Gamma(x, \ell, r_1 M)$ *such that*

(13) $\pi_{\mathcal{A},x}(P - \tilde{P}) = 0, \quad P - \tilde{P} \in r_2 M B(x, \delta),$

where $r_1 = C(K_1 K_2 + 1)$ *and* $r_2 = C(K_1 + 1)K_2$*. Here* C *is a constant depending only on* m *and* n*.*

Proof. According to (12), there exists

(14) $P' \in \Gamma(x, \ell, M)$ such that $P - P' \in K_2 M B(x, \delta)$.

Since $\pi_{\mathcal{A},x}\{B(x, \delta)\}$ is C-equivalent to $B_{\mathcal{A}}(\delta)$ by (3) from Section 30, we conclude from (14) that

(15) $\pi_{\mathcal{A},x}(P - P') \in C K_2 M B_{\mathcal{A}}(\delta).$

Combining (11) and (15), we see that

(16) $\pi_{\mathcal{A},x}(P - P') \in C K_1 K_2 M \pi_{\mathcal{A},x}\{\sigma(x, \ell) \cap B(x, \delta)\}.$

In view of (16) there exists

(17) $P'' \in C K_1 K_2 M[\sigma(x, \ell) \cap B(x, \delta)]$

such that $\pi_{\mathcal{A},x}(P - P') = \pi_{\mathcal{A},x}(P'')$. Set $\tilde{P} = P' + P''$. Then,

(18) $\pi_{\mathcal{A},x}(P - \tilde{P}) = 0.$

Furthermore, by (14) and (17),

(19) $\tilde{P} = P' + P'' \in \Gamma(x, \ell, M) + C K_1 K_2 M \sigma(x, \ell) \subseteq \Gamma(x, \ell, r_1 M)$

for $r_1 = C(1 + K_1 K_2)$, according to Property 1 from Section 13. Also, again by (14) and (17),

(20) $P - \tilde{P} = (P - P') - P'' \in K_2 M B(x, \delta) + C K_1 K_2 M B(x, \delta) \subseteq r_2 M B(x, \delta),$

for $r_2 = C(1 + K_1)K_2$. The statements (19), (18) and (20) are exactly the conclusions of the lemma. The lemma is thus proven. ■

Recall the definition (OK1) and (OK2) of the basic lengthscales $\delta(x, \mathcal{A}) \in [0, \infty]$ ($\mathcal{A} \subseteq \mathcal{M}, x \in E$) from Section 18. Recall also Lemma 1 from Section 30. Let $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$ and $x \in E$. By Lemma 1 from Section 30, the basic property of $\delta(x, \mathcal{A})$ is equivalent to the following: If $0 < \delta < \delta(x, \mathcal{A})$ then

(21) $B_{\mathcal{A}}(\delta) \subseteq C A_1(\mathcal{A}) \pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A})) \cap \mathcal{R}_{\mathcal{A}}(x, \delta)\},$

and if $\delta > \delta(x, \mathcal{A})$, then

(22) $B_{\mathcal{A}}(\delta) \not\subseteq c A_1(\mathcal{A}) \pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A})) \cap \mathcal{R}_{\mathcal{A}}(x, \delta)\}.$

According to the remark following Lemma 1 in Section 18, inclusion (21) holds also for $\delta = \delta(x, \mathcal{A})$, provided $0 < \delta(x, \mathcal{A}) < \infty$. By the definition of the constant A_0 in Section 17, we may assume that

(23) $A_0 > \max\left\{C, c^{-1}\right\}$ where C, c are as in (21), (22), respectively.

Therefore (21) and (22) imply the following. Fix $x \in E$, $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, $0 < \delta < \infty$. If $0 < \delta \leq \delta(x, \mathcal{A})$, then

(24) $B_{\mathcal{A}}(\delta) \subseteq A_0 A_1(\mathcal{A}) \pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A})) \cap \mathcal{R}_{\mathcal{A}}(x, \delta)\}$,

and if $\delta > \delta(x, \mathcal{A})$, then

(25) $B_{\mathcal{A}}(\delta) \not\subseteq A_0^{-1} A_1(\mathcal{A}) \pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A})) \cap \mathcal{R}_{\mathcal{A}}(x, \delta)\}$.

Recall also that for a dyadic cube Q with $\delta_Q \leq A_2^{-1}$ and a subset $\mathcal{A} \subseteq \mathcal{M}$, we say that Q is $\mathsf{OK}(\mathcal{A})$ if for all $x \in E \cap Q^*$,

(26) $A_2 \delta_Q \leq \delta(x, \mathcal{A})$.

If $\mathcal{A} = \emptyset$, then $\delta(x, \emptyset) = +\infty$ for all $x \in E$, and thus Q is always $\mathsf{OK}(\emptyset)$. A cube Q is almost $\mathsf{OK}(\mathcal{A})$ if

(27) $\sharp(E \cap Q^*) \leq 1$ or Q is $\mathsf{OK}(\bar{\mathcal{A}})$ for some $\bar{\mathcal{A}} \leq \mathcal{A}$.

In order to show that a dyadic cube Q with $\delta_Q \leq A_2^{-1}$ is $\mathsf{OK}(\mathcal{A})$, for $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, it is sufficient to prove that for all $x \in Q^* \cap E$,

(28) $B_{\mathcal{A}}(A_2\delta_Q) \subseteq A_0^{-1} A_1(\mathcal{A}) \pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A})) \cap \mathcal{R}_{\mathcal{A}}(x, A_2\delta_Q)\}$,

as follows from (25) and (26).

Lemma 3. *There exists a constant $0 < c_1 < 1$, depending only on m and n, for which the following holds: Let $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, Q a dyadic cube with $\delta_Q \leq A_2^{-1}$, $x \in E \cap Q^{**}$. Suppose that*

(29) $B_{\mathcal{A}}(A_2\delta_Q) \subseteq c_1 A_0^{-1} A_1(\mathcal{A})\ \pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A}) + 1) \cap B(x, A_2\delta_Q)\}$.

Then the cube Q is $\mathsf{OK}(\mathcal{A})$.

Proof. According to (28) and to (6) from Section 30, it is sufficient to show that for any $y \in E \cap Q^*$,

(30) $B_{\mathcal{A}}(A_2\delta_Q) \subseteq A_0^{-1} A_1(\mathcal{A})\ \pi_{\mathcal{A},y}\{\sigma(y, \ell(\mathcal{A})) \cap B(y, A_2\delta_Q)\}$.

Let $y \in E \cap Q^*$. We will show that y satisfies (30). Note that $x, y \in Q^{**}$, and hence $|x - y| \leq \sqrt{n}\delta_{Q^{**}} = 25\sqrt{n}\delta_Q$. According to the definition of $p_\sharp$ and A_0 from Section 17, we may suppose that

(31) $p_\sharp \geq 2$ and $A_0 > 25\sqrt{n}\max\{C, C_0\}$, for C, C_0 from Lemma 1.

From (2) of Section 17 and from (31) we know that $A_2 \geq A_0^2 A_1(\mathcal{M}) > 25\sqrt{n} C_0 A_0^{-1} A_1(\mathcal{A})$. Consequently ,

$$(32) \quad |x-y| \leq \sqrt{n}\delta_{Q^{**}} = 25\sqrt{n}\delta_Q < \frac{1}{C_0 A_0^{-1} A_1(\mathcal{A})} A_2 \delta_Q < \frac{1}{C_0 (c_1 A_0^{-1} A_1(\mathcal{A}))} A_2 \delta_Q,$$

since $c_1 < 1$.

We now select the constant c_1 such that $Cc_1 = 1$, where $C > 1$ is the constant from Lemma 1. Then by (31) and by (1) from Section 17,

$$(33) \qquad c_1 A_0^{-1} A_1(\mathcal{A}) \geq c_1 A_0^{-1} A_1(\emptyset) \geq c_1 A_0^{-1} \cdot A_0^2 \geq 1.$$

In view of (29), (32) and (33), we may apply Lemma 1 (for $\delta = A_2\delta_Q$, $K = c_1 A_0^{-1} A_1(\mathcal{A}) \geq 1$ and $\ell = \ell(\mathcal{A}) + 1$; note that $\ell \leq \ell_*$). We conclude that

$$\begin{aligned} (34) \qquad B_{\mathcal{A}}(A_2\delta_Q) &\subseteq Cc_1 A_0^{-1} A_1(\mathcal{A}) \pi_{\mathcal{A},y}\{\sigma(y, \ell(\mathcal{A})) \cap B(y, A_2\delta_Q)\} \\ &= A_0^{-1} A_1(\mathcal{A}) \pi_{\mathcal{A},y}(\sigma(y, \ell(\mathcal{A})) \cap B(y, A_2\delta_Q)). \end{aligned}$$

Now (30) follows from (34). The lemma is proven. ■

Lemma 4. *Let* $\emptyset \neq \mathcal{A} \subseteq \mathcal{M}$, Q *a dyadic cube with* $\delta_Q \leq A_2^{-1}$, $x \in E \cap Q^{**}$. *Let also* $1 < K \leq A_0^2 A_1(\mathcal{A})$. *Suppose that* $\nu \in \{0, 1\}$ *satisfies*

$$(35) \quad B_{\mathcal{A}}(A_2\delta_Q) \subseteq K\pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A}) - \nu) \cap \mathcal{R}_{\mathcal{A}}(x, A_2\delta_Q)\}\ ; \quad \textit{and}$$

$$(36) \quad B_{\mathcal{A}}(A_2\delta_Q) \not\subseteq K\pi_{\mathcal{A},x}\{\sigma(x, \ell(\mathcal{A}) - \nu) \cap B(x, A_2\delta_Q)\}.$$

Then there exists $\bar{\mathcal{A}} < \mathcal{A}$ *such that the cube* Q *is* $\mathrm{OK}(\bar{\mathcal{A}})$.

Proof. Our assumptions (35) and (36) are precisely the requirements of Lemma 2 from the preceding section. By the conclusion of that lemma, there exists $\bar{\mathcal{A}} < \mathcal{A}$, such that

$$(37) \qquad B_{\bar{\mathcal{A}}}(A_2\delta_Q) \subseteq CK^p \cdot \pi_{\bar{\mathcal{A}},x}\{\sigma(x, \ell(\mathcal{A}) - \nu) \cap B(x, A_2\delta_Q)\}.$$

We have $K \leq A_0^2 A_1(\mathcal{A})$ by assumption. Hence (37) implies that

$$(38) \qquad B_{\bar{\mathcal{A}}}(A_2\delta_Q) \subseteq C\left(A_0^2 A_1(\mathcal{A})\right)^p \cdot \pi_{\bar{\mathcal{A}},x}\{\sigma(x, \ell(\mathcal{A}) - \nu) \cap B(x, A_2\delta_Q)\}.$$

Since $\bar{\mathcal{A}} < \mathcal{A}$, then $\ell(\bar{\mathcal{A}}) + 1 \leq \ell(\mathcal{A}) - 1 \leq \ell(\mathcal{A}) - \nu$. By Property 4 from Section 13,

$$(39) \qquad \sigma(x, \ell(\mathcal{A}) - \nu) \subseteq C'\sigma(x, \ell(\bar{\mathcal{A}}) + 1).$$

According to (1) from Section 17,

$$(40) \qquad A_1(\bar{\mathcal{A}}) \geq \left(A_0^2 A_1(\mathcal{A})\right)^{p_\#}.$$

From the definition of A_0 and $p_\#$ in Section 17, we may assume that

$$(41) \qquad A_0 \text{ is larger than } \frac{CC'}{c_1} \quad \text{and} \quad p_\# > p + 1$$

where C, p are the constants from (38), C' is the constant from (39) and c_1 is the constant from Lemma 3.

Then (42) and (41) imply that

$$(42) \qquad CC'\left(A_0^2 A_1(\mathcal{A})\right)^p \leq c_1 A_0^{-1} A_1(\bar{\mathcal{A}}).$$

From (38), (39) and (40) we conclude that

$$(43) \qquad B_{\bar{\mathcal{A}}}(A_2\delta_Q) \subseteq c_1 A_0^{-1} A_1(\bar{\mathcal{A}})\ \pi_{\bar{\mathcal{A}},x}\left\{\sigma(x, \ell(\bar{\mathcal{A}})+1) \cap B(x, A_2\delta_Q)\right\}.$$

The estimate (43) and the fact that $x \in E \cap Q^{**}$ are the assumptions of Lemma 3. (Note also that $\bar{\mathcal{A}} \neq \emptyset$ since $\bar{\mathcal{A}} < \mathcal{A}$.) By that lemma, we conclude that Q is $\mathsf{OK}(\bar{\mathcal{A}})$. Since $\bar{\mathcal{A}} < \mathcal{A}$, the lemma is proven. ■

32. Preparation for the Proof: Analysis of Find-Neighbor

Recall from Section 20 the definition of the Calderón-Zygmund decomposition $CZ(\mathcal{A})$, associated with any subset $\mathcal{A} \subseteq \mathcal{M}$. Throughout this section, assume that we are given a subset $\mathcal{A}_0 \subset \mathcal{M}$ with $\mathcal{A}_0 \neq \mathcal{M}$, a dyadic cube Q_0 with $\delta_{Q_0} \leq A_2^{-1}$, a polynomial $P_0 \in \mathcal{P}$, $M_0 > 0$ and $x_0 \in \mathbb{R}^n$ that satisfy:

(FN1) $x_0 \in E \cap Q_0^{**}$. If $E \cap Q_0^* \neq \emptyset$, then $x_0 \in E \cap Q_0^*$.

(FN2) $Q_0 \in CZ(\mathcal{A}_0)$, and $Q_0 \notin CZ(\mathcal{A})$ for any $\mathcal{A} < \mathcal{A}_0$.

(FN3) $P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0)$.

Recall the procedure Find-Neighbor from Section 15. In the current section we analyze the outcome of the procedure Find-Neighbor, assuming that (FN1), (FN2) and (FN3) hold.

Lemma 1. *We have*

(FN1′) $x_0 \in E \cap Q_0^*$.

(FN2′) *The cube* Q_0 *is* $\mathsf{OK}(\mathcal{A}_0)$.
For any $\mathcal{A} < \mathcal{A}_0$, *the cube* Q_0 *is not almost* $\mathsf{OK}(\mathcal{A})$.

Proof. Assume on the contrary that Q_0 is almost $\mathsf{OK}(\mathcal{A})$ for some $\mathcal{A} < \mathcal{A}_0$. Then Q_0 is contained in a maximal almost $\mathsf{OK}(\mathcal{A})$-cube Q. Hence $Q \in CZ(\mathcal{A})$. According to Lemma 3 from Section 21, we know that $CZ(\mathcal{A})$ is a refinement of $CZ(\mathcal{A}_0)$. Since $Q_0 \in CZ(\mathcal{A}_0)$ by (FN2), it is impossible for Q strictly to contain Q_0. Hence necessarily $Q_0 = Q \in CZ(\mathcal{A})$, contradicting the assumption (FN2). Therefore

$$(1) \qquad Q_0 \text{ is not almost } \mathsf{OK}(\mathcal{A}) \text{ for any } \mathcal{A} < \mathcal{A}_0.$$

This establishes the second part of (FN2′). Recall that $\mathcal{A}_0 \neq \mathcal{M}$ and that $\mathcal{M}$ is the minimal element in our order relation $<$. Therefore $\mathcal{M} < \mathcal{A}_0$ and (1) states that in particular Q_0 is in not almost $\mathsf{OK}(\mathcal{M})$. By the definition of almost $\mathsf{OK}(\mathcal{M})$ from Section 20,

$$(2) \qquad \#(E \cap Q_0^*) > 1\,,$$

and by (FN1) we conclude (FN1′). Next, according to (FN2) we know that $Q_0 \in CZ(\mathcal{A}_0)$, and hence Q_0 is almost $\mathsf{OK}(\mathcal{A}_0)$. By (1) and (2), necessarily Q_0 is $\mathsf{OK}(\mathcal{A}_0)$. This finishes the proof. ■

Lemma 2. *For all* $x \in E \cap Q_0^*$,

$$(3) \qquad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq A_0\, A_1(\mathcal{A}_0)\, \pi_{\mathcal{A}_0,x}\big\{\sigma(x, \ell(\mathcal{A}_0)) \cap B(x, A_2\delta_{Q_0})\big\}\,.$$

Proof. If $\mathcal{A}_0 = \emptyset$ then (3) trivially holds. Assume $\mathcal{A}_0 \neq \emptyset$. By (FN2′), the cube Q is $\mathsf{OK}(\mathcal{A}_0)$. Since $x \in E \cap Q_0^*$, then (26) from the preceding section yields,

$$A_2\delta_{Q_0} \leq \delta(x, \mathcal{A}_0).$$

Consequently, (24) from the preceding section implies that

$$(4) \qquad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq A_0A_1(\mathcal{A}_0)\, \pi_{\mathcal{A}_0,x}\big\{\sigma(x, \ell(\mathcal{A}_0)) \cap \mathcal{R}_{\mathcal{A}_0}(x, A_2\delta_{Q_0})\big\}\,.$$

Assume on the contrary that (3) does not hold. That is,

$$(5) \qquad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \not\subseteq A_0A_1(\mathcal{A}_0)\, \pi_{\mathcal{A}_0,x}\big\{\sigma(x, \ell(\mathcal{A}_0)) \cap B(x, A_2\delta_{Q_0})\big\}\,.$$

The relations (4) and (5) are precisely the assumptions of Lemma 4 of the preceding section, for $\nu = 0, \mathcal{A} = \mathcal{A}_0$ and $K = A_0A_1(\mathcal{A}_0) \leq A_0^2A_1(\mathcal{A}_0)$. The conclusion of that lemma implies that Q_0 is $\mathsf{OK}(\mathcal{A})$, for some $\mathcal{A} < \mathcal{A}_0$. This contradicts (FN2′). Therefore, our assumption (5) was false. Consequently, (3) holds and the lemma is proven. ■

Lemma 3. *For all* $x \in E \cap Q_0^{***}$,

$$(6) \qquad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq CA_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x}\big\{\sigma(x, \ell(\mathcal{A}_0) - 1) \cap B(x, A_2\delta_{Q_0})\big\}\,.$$

where $C > 0$ *is a constant depending only on* m *and* n.

Proof. By (FN1′) we know that $x_0 \in E \cap Q_0^*$. According to Lemma 2,

$$(7) \qquad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq A_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x_0}\big\{\sigma(x_0, \ell(\mathcal{A}_0)) \cap B(x_0, A_2\delta_{Q_0})\big\}\,.$$

Let $x \in E \cap Q_0^{***}$. Then since $x, x_0 \in Q_0^{***}$,

$$(8) \qquad |x - x_0| \leq \sqrt{n}\,\delta_{Q_0^{***}} \leq C\delta_{Q_0} = \tfrac{CA_0A_1(\mathcal{A}_0)}{A_2} \cdot \tfrac{A_2\delta_{Q_0}}{A_0A_1(\mathcal{A}_0)}\,.$$

Assume, as we may (see Section 17), that,

(9) $A_0 > CC_0$ and $p_\# \geq 2$

where C, C_0 are the constants from (8) and from Lemma 1 of Section 31, respectively. Recall from (2) of Section 17 that $A_2 \geq (A_0 A_1(\mathcal{A}_0))^2$. Therefore (8) and (9) entail that

(10) $|x - x_0| < \frac{1}{C_0} \cdot \frac{A_2 \delta_{Q_0}}{A_0 A_1(\mathcal{A}_0)}$.

The statements (7) and (10) are the assumptions of Lemma 1 from Section 31 (for $K = A_0 A_1(\mathcal{A}_0) \geq 1$, $\ell = \ell(\mathcal{A}_0)$ and $\delta = A_2 \delta_{Q_0}$). By the conclusion of that lemma,

$$B_{\mathcal{A}_0}(A_2 \delta_{Q_0}) \subseteq C A_0 A_1(\mathcal{A}_0) \pi_{\mathcal{A}_0, x} \{\sigma(x, \ell(\mathcal{A}_0) - 1) \cap B(x, A_2 \delta_{Q_0})\},$$

and the lemma is proven. ■

Lemma 4. *The set* $\mathcal{A}_0$ *is monotonic.*

Proof. According to (FN1′), we know that $x_0 \in E \cap Q_0^*$. By Lemma 2,

(11) $B_{\mathcal{A}_0}(A_2 \delta_{Q_0}) \subseteq A_0 A_1(\mathcal{A}_0) \pi_{\mathcal{A}_0, x_0} \{\sigma(x_0, \ell(\mathcal{A}_0)) \cap B(x_0, A_2 \delta_{Q_0})\}$.

Assume on the contrary that the set $\mathcal{A}_0$ is not monotonic. In particular, $\mathcal{A}_0 \neq \emptyset$. According to Property 3 from Section 13 the set $\sigma(x_0, \ell(\mathcal{A}_0))$ is Whitney t-convex at x_0 with Whitney constant $\tilde{C} > 1$. Thus, based on (11), we may apply Lemma 4 from Section 30 for $\Omega = \sigma(x_0, \ell(\mathcal{A}_0))$, $\delta = A_2 \delta_{Q_0}$, $K = A_0 A_1(\mathcal{A}_0) \geq 1$). By the conclusion of that lemma, there exists a monotonic set $\mathcal{A} \leq \mathcal{A}_0$ such that

(12) $B_{\mathcal{A}}(A_2 \delta_{Q_0}) \subseteq C\tilde{C}(A_0 A_1(\mathcal{A}_0))^p \cdot \pi_{\mathcal{A}, x_0} \{\sigma(x_0, \ell(\mathcal{A}_0)) \cap B(x_0, A_2 \delta_{Q_0})\}$.

Since $\mathcal{A}$ is monotonic and $\mathcal{A}_0$ is not monotonic, evidently $\mathcal{A} < \mathcal{A}_0$. Hence $\ell(\mathcal{A}) + 1 \leq \ell(\mathcal{A}_0)$, and by Property 4 from Section 13,

(13) $\sigma(x_0, \ell(\mathcal{A}_0)) \subseteq C' \sigma(x_0, \ell(\mathcal{A}) + 1)$.

Recall from (1), Section 17 that

(14) $A_1(\mathcal{A}) \geq \left(A_0^2 A_1(\mathcal{A}_0)\right)^{p_\#}$.

By the definition of $A_0, p_\#$ from Section 17, we may assume that

(15) A_0 is larger than $\frac{C\tilde{C}C'}{c_1}$ and $p_\# > p + 1$

where $C, \tilde{C}, p$ are the constants from (12), C' is the constant from (13) and c_1 is the constant from Lemma 3 of Section 31. Then (12), (13) and (15) imply that

(16) $B_{\mathcal{A}}(A_2 \delta_{Q_0}) \subseteq c_1 A_0^{-1} A_1(\mathcal{A}) \cdot \pi_{\mathcal{A}, x_0} \{\sigma(x_0, \ell(\mathcal{A}) + 1) \cap B(x_0, A_2 \delta_{Q_0})\}$.

Since $x_0 \in E \cap Q_0^*$, we may invoke Lemma 3 from Section 31, based on (16). That Lemma implies that Q_0 is $\mathsf{OK}(\mathcal{A})$. Since $\mathcal{A} < \mathcal{A}_0$, this contradicts (FN2′). Hence our assumption, that $\mathcal{A}_0$ is not monotonic, is absurd. The lemma is proven. ■

Recall that $\mathcal{A}_0^-$ is the predecessor of $\mathcal{A}_0$ in our order relation on subsets of multi-indices. The decomposition $\mathsf{CZ}(\mathcal{A}_0^-)$ is a refinement of $\mathsf{CZ}(\mathcal{A}_0)$, by Lemma 3 from Section 21. Cubes in $\mathsf{CZ}(\mathcal{A}_0^-)$ may be much smaller than their containers in $\mathsf{CZ}(\mathcal{A}_0)$. Nevertheless, based on Lemma 2 from Section 30, we will show that these smaller cubes satisfy the same conditions as their containers in $\mathsf{CZ}(\mathcal{A}_0)$.

Lemma 5. *Let* $\widehat{Q} \in \mathsf{CZ}(\mathcal{A}_0^-)$ *be such that* $(1+c_G)\widehat{Q} \cap (1+c_G)Q_0 \neq \emptyset$. *Let* $x \in E \cap \widehat{Q}^{**}$. *Then,*

$$(17) \quad B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}}) \subseteq CA_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x}\left\{\sigma(x,\ell(\mathcal{A}_0)-1) \cap B(x, A_2\delta_{\widehat{Q}})\right\},$$

where C *is a constant depending only on* m *and* n.

Proof. First, note that $\mathcal{A}_0^-$ makes sense, as $\mathcal{A}_0 \neq \mathcal{M}$; and that we may suppose $\mathcal{A}_0 \neq \emptyset$, since (17) holds trivially for $\mathcal{A}_0 = \emptyset$. Second, by Lemma 7 from Section 21, we know that $\widehat{Q} \subseteq Q_0^*$. Therefore, $x \in E \cap Q_0^{***}$, and by Lemma 3,

$$(18) \quad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq CA_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x}\{\sigma(x,\ell(\mathcal{A}_0)-1) \cap B(x, A_2\delta_{Q_0})\}.$$

According to Lemma 6 from Section 21, we have

$$\delta_{\widehat{Q}} \leq C\delta_{Q_0}.$$

Suppose first that $\delta_{\widehat{Q}} \geq \frac{\delta_{Q_0}}{2}$. Then,

$$(19) \quad \frac{\delta_{Q_0}}{2} \leq \delta_{\widehat{Q}} \leq C\delta_{Q_0}.$$

Consequently, the sets $B_{\mathcal{A}_0}(A_2\delta_{Q_0}), B(x, A_2\delta_{Q_0})$ are C'-equivalent to the sets $B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}}), B(x, A_2\delta_{\widehat{Q}})$, respectively, because of (19). From (18) we conclude (17). This completes the proof, for the case $\delta_{\widehat{Q}} \geq \frac{\delta_{Q_0}}{2}$.

We may thus restrict our attention to the case where

$$(20) \quad \delta_{\widehat{Q}} < \frac{\delta_{Q_0}}{2}.$$

From (6) of Section 30 we have $B(x,\delta) \subseteq \mathcal{R}_{\mathcal{A}_0}(x,\delta)$. Hence the inclusion (18) implies that

$$(21) \quad B_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq r\pi_{\mathcal{A}_0,x}\{\sigma(x,\ell(\mathcal{A}_0)-1) \cap \mathcal{R}_{\mathcal{A}_0}(x, A_2\delta_{Q_0})\},$$

for $r = C_1A_0A_1(\mathcal{A}_0)$, where C_1 is a constant depending only on m and n.

By (20) we know that $\delta_{\widehat{Q}^+} = 2\delta_{\widehat{Q}} < \delta_{Q_0}$. According to (21) and to (8) from Section 30, we get that

$$(22) \quad B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}^+}) \subseteq r\pi_{\mathcal{A}_0,x}\left\{\sigma(x,\ell(\mathcal{A}_0)-1)\cap \mathcal{R}_{\mathcal{A}_0}(x,A_2\delta_{\widehat{Q}^+})\right\},$$

for $r = C_1A_0A_1(\mathcal{A}_0)$. The sets $B(x,A_2\delta_{\widehat{Q}}), B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}})$ are C_2-equivalent to $B(x,A_2\delta_{\widehat{Q}^+})$, $B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}^+})$, respectively, for some constant C_2 depending only on m and n. Assume on the contrary that (17) does not hold, with constant $C = C_1C_2^2$. That is, assume on the contrary that

$$(23) \quad B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}}) \not\subseteq C_1C_2^2A_0A_1(\mathcal{A}_0)\,\pi_{\mathcal{A}_0,x}\left\{\sigma(x,\ell(\mathcal{A}_0)-1)\cap B(x,A_2\delta_{\widehat{Q}})\right\}.$$

The definition of the constant C_2 implies that

$$(24) \quad B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}^+}) \not\subseteq r\pi_{\mathcal{A}_0,x}\left\{\sigma(x,\ell(\mathcal{A}_0)-1)\cap B(x,A_2\delta_{\widehat{Q}^+})\right\}.$$

for $r = C_1A_0A_1(\mathcal{A}_0)$, the same r as in (22). We assume, as we may, that

$$(25) \quad A_0 > C_1 \quad \text{where } C_1 \text{ is the constant from (22) and (24)}.$$

Consequently, $K := C_1A_0A_1(\mathcal{A}_0)$ satisfies $K < A_0^2A_1(\mathcal{A}_0)$. Note also that $x \in E \cap \widehat{Q}^{**} \subseteq E \cap (\widehat{Q}^+)^{**}$. Using (22) and (24), we may apply Lemma 4 of Section 31 with $\nu = 1, Q = \widehat{Q}^+$ and K as just defined. According to that lemma, we obtain $\mathcal{A} < \mathcal{A}_0$ such that $\widehat{Q}^+$ is $OK(\mathcal{A})$. The fact that $\mathcal{A} < \mathcal{A}_0$ implies that $\mathcal{A} \le \mathcal{A}_0^-$, and by the definition of almost OK from Section 20, the cube $\widehat{Q}^+$ is almost $OK(\mathcal{A}_0^-)$. On the other hand, $\widehat{Q} \in CZ(\mathcal{A}_0^-)$, $\delta_{\widehat{Q}} < \frac{\delta_{Q_0}}{2} < A_2^{-1}$ and thus $\widehat{Q}^+$ cannot be almost $OK(\mathcal{A}_0^-)$. Thus we arrive at a contradiction, and (23) is false. This proves the lemma. ■

For $x \in \mathbb{R}^n$, $\mathcal{A} \subseteq \mathcal{M}$ and a point $a \in \mathbb{R}^{\sharp(\mathcal{A})}$ we put

$$\pi_{\mathcal{A},x}^{-1}(a) = \{P \in \mathcal{P} : \pi_{\mathcal{A},x}(P) = a\}.$$

The set $\pi_{\mathcal{A},x}^{-1}(a)$ is an affine subspace in $\mathcal{P}$.

Lemma 6. *Let* $\widehat{Q}$ *and* x *be as in Lemma* 5. *Then,*

$$(26) \quad \pi_{\mathcal{A}_0,x}^{-1}(0)\cap\sigma(x,\ell(\mathcal{A}_0)-2) \subseteq CB(x,A_2\delta_{\widehat{Q}}),$$

where C *is a constant depending only on* m *and* n.

Proof. We consider first the case where $\delta_{\widehat{Q}} < A_2^{-1}$. In this case, we prove the stronger statement,

$$(27) \quad \pi_{\mathcal{A}_0,x}^{-1}(0)\cap\sigma(x,\ell(\mathcal{A}_0)-3) \subseteq CB(x,A_2\delta_{\widehat{Q}}).$$

Indeed (27) implies (26), since $\sigma(x, \ell(\mathcal{A}_0) - 2) \subseteq \tilde{C}\sigma(x, \ell(\mathcal{A}_0) - 3)$ by Property 4 from Section 13. We focus on proving (27). According to Lemma 5,

$$B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}}) \subseteq r\pi_{\mathcal{A}_0,x}\left\{\sigma(x, \ell(\mathcal{A}_0) - 1) \cap B(x, A_2\delta_{\widehat{Q}})\right\}$$

for $r = C'A_0A_1(\mathcal{A}_0)$. Since $\sigma(x, \ell(\mathcal{A}_0) - 1) \subseteq \tilde{C}\sigma(x, \ell(\mathcal{A}_0) - 3)$ by Property 4 from Section 13, we deduce that

$$(28) \quad B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}}) \subseteq r\pi_{\mathcal{A}_0,x}\left\{\sigma(x, \ell(\mathcal{A}_0) - 3) \cap B(x, A_2\delta_{\widehat{Q}})\right\} .$$

for $r = CA_0A_1(\mathcal{A}_0)$. Let us assume by contradiction that,

$$(29) \quad 0 \in \pi_{\mathcal{A}_0,x}\left\{\sigma(x, \ell(\mathcal{A}_0) - 3) \setminus r^{-1}B(x, A_2\delta_{\widehat{Q}})\right\}$$

for the same $r = CA_0A_1(\mathcal{A}_0)$, as in (28). We will show that (29) cannot hold. The statements (28) and (29) are precisely the assumptions of Lemma 5 from Section 30 (for $\Omega = \sigma(x, \ell(\mathcal{A}_0) - 3), \delta = A_2\delta_{\widehat{Q}}, K = r = CA_0A_1(\mathcal{A}_0) \geq 1)$. That lemma implies that for some $\mathcal{A} < \mathcal{A}_0$,

$$(30) \quad B_{\mathcal{A}}(A_2\delta_{\widehat{Q}}) \subseteq r\pi_{\mathcal{A},x}\left\{\sigma(x, \ell(\mathcal{A}_0) - 3) \cap B(x, A_2\delta_{\widehat{Q}})\right\},$$

for $r = 2(CA_0A_1(\mathcal{A}_0))^2$. Note that $\ell(\mathcal{A}) + 1 \leq \ell(\mathcal{A}_0) - 3$ as $\mathcal{A} < \mathcal{A}_0$. We conclude from Property 4 of Section 13 that $\sigma(x, \ell(\mathcal{A}_0){-}3) \subseteq \widehat{C}\sigma(x, \ell(\mathcal{A}){+}1)$. Consequently, (30) implies that

$$(31) \quad B_{\mathcal{A}}(A_2\delta_{\widehat{Q}}) \subseteq r\pi_{\mathcal{A},x}\left\{\sigma(x, \ell(\mathcal{A}) + 1) \cap B(x, A_2\delta_{\widehat{Q}})\right\} ,$$

for $r = C'(A_0A_1(\mathcal{A}_0))^2$. The sets $B_{\mathcal{A}}(A_2\delta_{\widehat{Q}})$, $B(x, A_2\delta_{\widehat{Q}})$ are C-equivalent to the sets $B_{\mathcal{A}}(A_2\delta_{\widehat{Q}^+})$, $B(x, A_2\delta_{\widehat{Q}^+})$, respectively. Therefore, by (31),

$$(32) \quad B_{\mathcal{A}}(A_2\delta_{\widehat{Q}^+}) \subseteq r\pi_{\mathcal{A},x}\left\{\sigma(x, \ell(\mathcal{A}) + 1) \cap B(x, A_2\delta_{\widehat{Q}^+})\right\} ,$$

for $r = \tilde{C}(A_0A_1(\mathcal{A}_0))^2$. Recall that $\mathcal{A} < \mathcal{A}_0$, and that by (1) from Section 17,

$$(33) \quad A_1(\mathcal{A}) \geq (A_0^2A_1(\mathcal{A}_0))^{p_\#} .$$

We stipulate, as we may, that

$$(34) \quad A_0 > \tfrac{\tilde{C}}{c_1}, \quad \text{and} \quad p_\# \geq 3,$$

where $\tilde{C}$ is the constant from (32) and c_1 is the constant from Lemma 3 of Section 31. Then, (33) and (34) give

$$(35) \quad \tilde{C}(A_0A_1(\mathcal{A}_0))^2 < c_1A_0^{-1}A_1(\mathcal{A}).$$

From (32) and (35) we get,

$$(36) \quad B_{\mathcal{A}}(A_2\delta_{\widehat{Q}^+}) \subseteq c_1A_0^{-1}A_1(\mathcal{A})\ \pi_{\mathcal{A},x}\left\{\sigma(x, \ell(\mathcal{A}) + 1) \cap B(x, A_2\delta_{\widehat{Q}^+})\right\}.$$

Recall that $\delta_{\widehat{Q}} < A_2^{-1}$, and thus $\delta_{\widehat{Q}^+} \leq A_2^{-1}$. Also, note that $\mathcal{A} \neq \emptyset$, since $\mathcal{A} < \mathcal{A}_0$. Since $x \in E \cap \widehat{Q}^{**} \subseteq E \cap (\widehat{Q}^+)^{**}$, by (36) the requirements of Lemma 3 from Section 31 are fulfilled, for $Q = \widehat{Q}^+$. The conclusion of that lemma asserts that the cube $\widehat{Q}^+$ is $\mathrm{OK}(\mathcal{A})$. Since $\mathcal{A} < \mathcal{A}_0$, the cube $\widehat{Q}^+$ is almost $\mathrm{OK}(\mathcal{A}_0^-)$ by the definition of almost $\mathrm{OK}(\mathcal{A}_0^-)$. This contradicts the fact that $\widehat{Q}$ is in $\mathrm{CZ}(\mathcal{A}_0^-)$. Thus, our assumption (29) is false. That is,

$$\pi^{-1}_{\mathcal{A}_0,x}(0) \cap \sigma(x, \ell(\mathcal{A}_0) - 3) \subseteq r^{-1}B(x, A_2\delta_{\widehat{Q}}),$$

for $r = CA_0A_1(\mathcal{A})$. Since $r \geq 1$, we conclude (27). In particular, the lemma is proven for the case where $\delta_{\widehat{Q}} < A_2^{-1}$.

Suppose now that $\delta_{\widehat{Q}} = A_2^{-1}$. According to Lemma 3 from Section 21, the Calderón-Zygmund decomposition $\mathrm{CZ}(\mathcal{A}_0^-)$ is a refinement of $\mathrm{CZ}(\mathcal{A}_0)$. Since $Q_0 \in \mathrm{CZ}(\mathcal{A}_0)$ and $(1 + c_G)Q_0 \cap (1 + c_G)\widehat{Q} \neq \emptyset$, by Lemma 8 from Section 21 we may pick a cube $\widetilde{Q} \in \mathrm{CZ}(\mathcal{A}_0^-)$ such that

$$\widetilde{Q} \subseteq Q_0 \quad \text{and} \quad (1 + c_G)\widetilde{Q} \cap (1 + c_G)\widehat{Q} \neq \emptyset.$$

By (FN2), we know that $\widetilde{Q} \neq Q_0$. Since $\delta_{Q_0} \leq A_2^{-1}$, we conclude that $\delta_{\widetilde{Q}} \leq \frac{A_2^{-1}}{2}$. Since $\delta_{\widehat{Q}} = A_2^{-1}$ and $(1 + c_G)\widetilde{Q} \cap (1 + c_G)\widehat{Q} \neq \emptyset$, Lemma 2 from Section 21 implies that

(37) $\quad \delta_{\widetilde{Q}} = \frac{1}{2}A_2^{-1}$.

Lemma 4 from Section 21, based on (37), yields that $\widetilde{Q}^{**} \cap E \neq \emptyset$. Pick $\tilde{x} \in \widetilde{Q}^{**} \cap E$. We know that $\delta_{\widetilde{Q}} < A_2^{-1}, \widetilde{Q} \in \mathrm{CZ}(\mathcal{A}_0^-)$, $\tilde{x} \in \widetilde{Q}^{**} \cap E$ and $(1 + c_G)\widetilde{Q} \cap (1 + c_G)Q_0 \neq \emptyset$. Thus, we are in the case already treated, and hence by (27),

(38) $\quad \pi^{-1}_{\mathcal{A}_0,\tilde{x}}(0) \cap \sigma(\tilde{x}, \ell(\mathcal{A}_0) - 3) \subseteq CB(\tilde{x}, A_2\delta_{\widetilde{Q}}) \subseteq CB(\tilde{x}, 1)\,,$

where the last inclusion follows since $A_2\delta_{\widetilde{Q}} \leq 1$.

Recall that $(1 + c_G)\widetilde{Q} \cap (1 + c_G)\widehat{Q} \neq \emptyset$ with $\delta_{\widehat{Q}} = A_2^{-1}, \delta_{\widetilde{Q}} \leq A_2^{-1}$ and $\tilde{x} \in \widetilde{Q}^{**}, x \in \widehat{Q}^{**}$. Consequently,

(39) $\quad |x - \tilde{x}| < CA_2^{-1}\,.$

Let us pick any

(40) $\quad P \in \pi^{-1}_{\mathcal{A}_0,x}(0) \cap \sigma(x, \ell(\mathcal{A}_0) - 2).$

To obtain (26), it is sufficient to show that

(41) $\quad P \in CB(x, A_2\delta_{\widehat{Q}}) = CB(x, 1)\,.$

From (40) and Property 2 of Section 13, there exists $\tilde{P} \in \mathcal{P}$ such that

$$(42) \quad \tilde{P} \in C\sigma(\tilde{x}, \ell(\mathcal{A}_0) - 3), \quad \text{and} \quad \tilde{P} - P \in CB(x, \tilde{x}) \subseteq C'B(\tilde{x}, A_2^{-1}),$$

where the last inclusion follows from (39). According to (40), we know that $\pi_{\mathcal{A}_0,x}(P) = 0$. Lemma 4 tells us that $\mathcal{A}_0$ is monotonic. By (46) from Section 30, also $\pi_{\mathcal{A}_0,\tilde{x}}(P) = 0$. Projecting the right hand side of (42), we get that

$$(43) \quad \begin{aligned} \pi_{\mathcal{A}_0,\tilde{x}}(\tilde{P}) = \pi_{\mathcal{A}_0,\tilde{x}}(\tilde{P} - P) \in \pi_{\mathcal{A}_0,\tilde{x}}\{C'B(\tilde{x}, A_2^{-1})\} &\subseteq \pi_{\mathcal{A}_0,\tilde{x}}\{\tilde{C}A_2^{-1}B(\tilde{x}, 1)\} \\ &\subseteq \bar{C}A_2^{-1}B_{\mathcal{A}_0}(1), \end{aligned}$$

where the last two inclusions follow from (2) of Section 12 and (3) from Section 30, respectively. Since $\tilde{x} \in E \cap \widetilde{Q}^{**}$, $\widetilde{Q} \in CZ(\mathcal{A}_0^-)$ and $(1 + c_G)\widetilde{Q} \cap (1 + c_G)Q_0 \neq \emptyset$, we may apply Lemma 5. By the conclusion of that lemma,

$$B_{\mathcal{A}_0}(A_2\delta_{\widetilde{Q}}) \subseteq CA_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,\tilde{x}}\big\{\sigma(\tilde{x}, \ell(\mathcal{A}_0) - 1) \cap B(\tilde{x}, A_2\delta_{\widetilde{Q}})\big\},$$

and hence,

$$(44) \quad \tfrac{1}{\tilde{C}A_0A_1(\mathcal{A}_0)}B_{\mathcal{A}_0}(1) \subseteq \pi_{\mathcal{A}_0,\tilde{x}}\big\{\sigma(\tilde{x}, \ell(\mathcal{A}_0) - 3) \cap B(\tilde{x}, 1)\big\},$$

where we have used (37), as well as Property 4 from Section 13. Recall that $A_2 > A_0A_1(\mathcal{A}_0)$ according to Section 17. Using (43) and (44) we deduce that there exists

$$(45) \quad P' \in C[\sigma(\tilde{x}, \ell(\mathcal{A}_0) - 3) \cap B(\tilde{x}, 1)]$$

such that

$$(46) \quad \pi_{\mathcal{A}_0,\tilde{x}}(P') = \pi_{\mathcal{A}_0,\tilde{x}}(\tilde{P}).$$

According to (42), (45) and (46), we have that

$$P' - \tilde{P} \in \pi^{-1}_{\mathcal{A}_0,\tilde{x}}(0) \cap C\sigma(\tilde{x}, \ell(\mathcal{A}_0) - 3).$$

With the help of (38), we conclude that

$$(47) \quad P' - \tilde{P} \in CB(\tilde{x}, 1).$$

Combining (42) with (47) and (45), we see that

$$P = (P - \tilde{P}) + (\tilde{P} - P') + P' \in C'B(\tilde{x}, A_2^{-1}) + CB(\tilde{x}, 1) + CB(\tilde{x}, 1) \subseteq \tilde{C}B(\tilde{x}, 1).$$

From (39), we get that $P \in C''B(x, 1)$ and thus (41) is proven. Therefore, we have proved that (40) implies (41) under the assumption that $\delta_{\widehat{Q}} = A_2^{-1}$. Equivalently,

$$\pi^{-1}_{\mathcal{A}_0,x}(0) \cap \sigma(x, \ell(\mathcal{A}_0) - 2) \subseteq CB(x, A_2\delta_{\widehat{Q}})$$

also in the case $\delta_{\widehat{Q}} = A_2^{-1}$. The lemma is thus proven in all cases. ■

We set, for any $x \in E, \tilde{P}_0 \in \mathcal{P}$ and $M > 0$,

$$(48) \quad \Gamma^{\sharp}_{\mathcal{A}_0}(x, \tilde{P}_0, M) = \Gamma(x, \ell(\mathcal{A}_0) - 1, M) \cap \pi^{-1}_{\mathcal{A}_0,x}(\pi_{\mathcal{A}_0,x}(\tilde{P}_0)).$$

Lemma 7. *Let* $x \in E$ *be such that* $x \in \widehat{Q}^{**}$ *for some cube* $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ *with* $(1+c_G)\widehat{Q} \cap (1+c_G)Q_0 \neq \emptyset$. *Then,*

(49) $\Gamma^\sharp_{\mathcal{A}_0}(x, P_0, CM_0) \neq \emptyset.$

Moreover, for any $A > 1$ *and* $P \in \Gamma^\sharp_{\mathcal{A}_0}(x, P_0, AM_0)$,

(50) $P - P_0 \in CAM_0B(x, A_2\delta_{Q_0}).$

Here C *is a constant depending only on* m *and* n.

Proof. By (FN3), we know that

(51) $P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0) \subseteq \Gamma(x, \ell(\mathcal{A}_0) - 1, CM_0) + CM_0B(x_0, x)$

where the inclusion follows from Property 2 of Section 13. By Lemma 7 from Section 21, we have $x, x_0 \in Q_0^{***}$, and hence $|x_0 - x| \leq \sqrt{n}\delta_{Q_0^{***}} \leq C\delta_{Q_0}$. Therefore (51), with the help of (3) from Section 12, implies that

(52) $P_0 \in \Gamma(x, \ell(\mathcal{A}_0) - 1, CM_0) + CM_0B(x, \delta_{Q_0}).$

Recall that $A_2 > 1$ by (2) from Section 17. Thus (2) from Section 12 entails that $B(x, \delta_{Q_0}) \subseteq \frac{1}{A_2}B(x, A_2\delta_{Q_0})$. Together with (52), this gives

(53) $P_0 \in \Gamma(x, \ell(\mathcal{A}_0) - 1, CM_0) + \frac{C}{A_2}M_0B(x, A_2\delta_{Q_0}).$

Since $x \in E \cap Q_0^{***}$, Lemma 3 implies that

(54) $\mathcal{B}_{\mathcal{A}_0}(A_2\delta_{Q_0}) \subseteq CA_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x}\{\sigma(x, \ell(\mathcal{A}_0) - 1) \cap B(x, A_2\delta_{Q_0})\}.$

The inclusions (54) and (53) are the assumptions of Lemma 2 from Section 31 (with $K_1 = CA_0A_1(\mathcal{A}_0), K_2 = \frac{C}{A_2}$, $\delta = A_2\delta_{Q_0}, \ell = \ell(\mathcal{A}_0) - 1$). By the conclusion of that lemma, with the help of the definition (48), there exists $\tilde{P} \in \mathcal{P}$ such that

(55) $\tilde{P} \in \Gamma^\#_{\mathcal{A}_0}(x, P_0, r_1M_0) \cap [P_0 + r_2M_0B(x, A_2\delta_{Q_0})]$

for $r_1 = \tilde{C}\left(1 + \frac{CA_0A_1(\mathcal{A}_0)}{A_2}\right), r_2 = \frac{\tilde{C}}{A_2}(1 + CA_0A_1(\mathcal{A}_0))$. Note that $r_1 < C'$ and $r_2 < C'$ since $A_2 \geq A_0A_1(\mathcal{A}_0)$ by (2) from Section 17. The statement (55) implies, in particular, that,

(56) $\Gamma^\sharp_{\mathcal{A}_0}(x, P_0, C'M_0) \neq \emptyset.$

We thus conclude (49), the first part of the lemma. We move to the second part of the lemma. According to (55),

(57) $\tilde{P} - P_0 \in CM_0B(x, A_2\delta_{Q_0}).$

Pick

$$(58)\quad P \in \Gamma^{\sharp}_{\mathcal{A}_0}(x, P_0, AM_0) \subseteq \Gamma(x, \ell(\mathcal{A}_0) - 1, AM_0),$$

where the inclusion is justified by (48). Then by (55), (58), and the definition (48) of $\Gamma^{\sharp}$,

$$(59)\quad P - \tilde{P} \in \Gamma(x, \ell(\mathcal{A}_0) - 1, AM_0) - \Gamma(x, \ell(\mathcal{A}_0) - 1, CM_0)$$
$$\subseteq \tilde{C}AM_0\sigma(x, \ell(\mathcal{A}_0) - 1) \subseteq C'AM_0\sigma(x, \ell(\mathcal{A}_0) - 2),$$

where we used Property 1 and Property 4 from Section 13. Note also that by (55) and (58) we have

$$(60)\quad \pi_{\mathcal{A}_0,x}(P - \tilde{P}) = 0.$$

According to the assumptions of the present lemma, there exists $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ with $x \in E \cap \widehat{Q}^{**}$ and $(1 + c_G)\widehat{Q} \cap (1 + c_G)Q_0 \neq \emptyset$. We may thus apply Lemma 6, based on (59) and (60). We get that

$$(61)\quad P - \tilde{P} \in CAM_0B(x, A_2\delta_{\widehat{Q}}).$$

According to Lemma 6 from Section 21, $\delta_{\widehat{Q}} \leq C\delta_{Q_0}$. Consequently, (61) implies that

$$(62)\quad P - \tilde{P} \in C'AM_0B(x, A_2\delta_{Q_0}).$$

Combining (57) and (62), we obtain the desired estimate (50). The lemma is thus proven. ■

Lemma 8. *Let* $\widehat{Q}, \widetilde{Q} \in CZ(\mathcal{A}_0^-)$ *be two cubes, such that* $(1+c_G)Q_0$ *intersects both* $(1+c_G)\widehat{Q}$ *and* $(1+c_G)\widetilde{Q}$*. Assume also that* $(1+c_G)\widehat{Q}\cap(1+c_G)\widetilde{Q} \neq \emptyset$*. Let* $x_1 \in E\cap\widehat{Q}^{**}, x_2 \in E\cap\widetilde{Q}^{**}$*. Let also* $P_1, P_2 \in \mathcal{P}$ *and* $A > 1$*. Assume that*

$$(63)\quad P_1 \in \Gamma^{\sharp}_{\mathcal{A}_0}(x_1, P_0, AM_0), \quad P_2 \in \Gamma^{\sharp}_{\mathcal{A}_0}(x_2, P_0, AM_0)\,.$$

Then,

$$(64)\quad P_1 - P_2 \in CAM_0B(x_1, A_2\delta_{\widehat{Q}}),$$

where $C > 0$ *is a constant depending only on* m *and* n*.*

Proof. By (63), $P_2 \in \Gamma^{\sharp}_{\mathcal{A}_0}(x_2, P_0, AM_0)$. According to the definition (48) of $\Gamma^{\#}$,

$$(65)\quad \pi_{\mathcal{A}_0,x_2}(P_2 - P_0) = 0\,.$$

Lemma 4 entails that $\mathcal{A}_0$ is monotonic. We will now use the basic property of monotonic sets; according to (46) from Section 30, also

$$(66)\quad \pi_{\mathcal{A}_0,x_1}(P_2 - P_0) = 0\,.$$

Next, (48) and (63), followed by Property 2 from Section 13, imply that,

(67) $P_2 \in \Gamma(x_2, \ell(\mathcal{A}_0)-1, AM_0) \subseteq \Gamma(x_1, \ell(\mathcal{A}_0)-2, CAM_0)+CAM_0B(x_2, x_1)$.

By our assumptions, $(1+c_G)\widehat{Q} \cap (1+c_G)\widetilde{Q} \neq \emptyset$, and $\widehat{Q}, \widetilde{Q} \in CZ(\mathcal{A}_0^-)$. By Lemma 2 from Section 21 we know that $\delta_{\widehat{Q}}$ and $\delta_{\widetilde{Q}}$ are comparable. Recall that $x_1 \in \widehat{Q}^{**}, x_2 \in \widetilde{Q}^{**}$. Therefore,

(68) $|x_1 - x_2| < C\delta_{\widehat{Q}}$.

Consequently $B(x_2, x_1) \subseteq CB(x_1, \delta_{\widehat{Q}})$ and by (67) we can assert that

(69) $P_2 \in \Gamma(x_1, \ell(\mathcal{A}_0)-2, CAM_0) + CAM_0B(x_1, \delta_{\widehat{Q}})$.

Recall that $A_2 > 1$ by (2) of Section 17. Thus (2) from Section 12 implies that $B(x_1, \delta_{\widehat{Q}}) \subseteq \frac{1}{A_2}B(x_1, A_2\delta_{\widehat{Q}})$. Using (69) we deduce that

(70) $P_2 \in \Gamma(x_1, \ell(\mathcal{A}_0)-2, CAM_0) + \frac{C}{A_2}AM_0B(x_1, A_2\delta_{\widehat{Q}})$.

Note that x_1 and $\widehat{Q}$ satisfy the requirements of Lemma 5; indeed, $x_1 \in \widehat{Q}^{**} \cap E$ by our assumptions, and $(1+c_G)\widehat{Q} \cap (1+c_G)Q_0 \neq \emptyset$. By the conclusion of Lemma 5,

(71)
$$\begin{aligned} B_{\mathcal{A}_0}(A_2\delta_{\widehat{Q}}) &\subseteq CA_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x_1}\left\{\sigma(x_1, \ell(\mathcal{A}_0)-1) \cap B(x_1, A_2\delta_{\widehat{Q}})\right\} \\ &\subseteq C'A_0A_1(\mathcal{A}_0)\pi_{\mathcal{A}_0,x_1}\left\{\sigma(x_1, \ell(\mathcal{A}_0)-2) \cap B(x_1, A_2\delta_{\widehat{Q}})\right\}, \end{aligned}$$

where the second inclusion is correct since $\sigma(x_1, \ell(\mathcal{A}_0)-1) \subseteq C'\sigma(x, \ell(\mathcal{A}_0)-2)$ by Property 4 from Section 13. The inclusions (71) and (70) are precisely the assumptions of Lemma 2 from Section 31 ($K_1 = C'A_0A_1(\mathcal{A}_0), K_2 = \frac{C}{A_2}$, $\delta = A_2\delta_{\widehat{Q}}, \ell = \ell(\mathcal{A}_0)-2, M = CAM_0$). By the conclusion of that lemma, there exists

(72) $\hat{P} \in \Gamma\left(x_1, \ell(\mathcal{A}_0)-2, \tilde{C}AM_0\left(1+\frac{C'A_0A_1(\mathcal{A}_0)}{A_2}\right)\right)$

such that

(73) $P_2 - \hat{P} \in \frac{CAM_0}{A_2}(1 + A_0A_1(\mathcal{A}_0)) \cdot B(x_1, A_2\delta_{\widehat{Q}})$

and

(74) $\pi_{\mathcal{A}_0,x_1}(\hat{P} - P_2) = 0$.

Recall that $P_1 \in \Gamma^\sharp_{\mathcal{A}_0}(x_1, P_0, AM_0)$ by (63). The definition (48) of $\Gamma^\sharp_{\mathcal{A}_0}$, together with (66) and (74) imply that

(75) $\pi_{\mathcal{A}_0,x_1}(\hat{P} - P_1) = 0$.

Furthermore, since $P_1 \in \Gamma^{\sharp}_{\mathcal{A}_0}(x_1, P_0, AM_0)$ by (63), the definition (48) entails that

$$(76)\qquad P_1 \in \Gamma(x_1, \ell(\mathcal{A}_0) - 1, AM_0) \subseteq \Gamma(x_1, \ell(\mathcal{A}_0) - 2, CAM_0)$$

where the last inclusion follows from Property 4 from Section 13. Using Property 1 from that section, together with (72), (76) we get that

$$(77)\qquad \begin{aligned} \hat{P} - P_1 &\in \tilde{C}AM_0 \left(1 + \tfrac{C' A_0 A_1(\mathcal{A}_0)}{A_2}\right) \sigma(x_1, \ell(\mathcal{A}_0) - 2) \\ &\subseteq CAM_0\sigma(x_1, \ell(\mathcal{A}_0) - 2), \end{aligned}$$

since $A_2 > A_0A_1(\mathcal{A}_0)$ by (2) from Section 17. Given (75) and (77), Lemma 6 tells us that

$$(78)\qquad \hat{P} - P_1 \in CAM_0B(x_1, A_2\delta_{\widehat{Q}})\,.$$

(We may invoke Lemma 6 since $x_1 \in E \cap \widehat{Q}^{**}$, $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ and $(1 + c_G)\widehat{Q} \cap (1 + c_G)Q_0 \neq \emptyset$.) Now, (73), (78) and the fact that $A_2 > A_0A_1(\mathcal{A}_0)$, imply that

$$(79)\qquad P_1 - P_2 \in CAM_0B(x_1, A_2\delta_{\widehat{Q}})\,.$$

The statement (79) is the conclusion of the lemma. ■

Recall the definition of the procedure "Find-Neighbor" from Section 15. Let $\tilde{P}_0 \in \mathcal{P}, \mathcal{A} \subseteq \mathcal{M}, x \in E$. Then,

$$P = \text{Find-Neighbor}(\tilde{P}_0, \mathcal{A}, x)$$

is a polynomial in $\mathcal{P}$ that satisfies the following: For any $M > 0$,

$$(80)\qquad \Gamma^{\sharp}_{\mathcal{A}}(x, \tilde{P}_0, M) \neq \emptyset \quad \Rightarrow \quad P \in \Gamma^{\sharp}_{\mathcal{A}}(x, \tilde{P}_0, CM),$$

where $C > 0$ is a constant depending only on m and n. We will conclude this section with the following lemma, which is a reformulation of Lemma 8.

Lemma 9. *Let $\widehat{Q}, \widetilde{Q} \in CZ(\mathcal{A}_0^-)$ be two cubes such that*

$$(1 + c_G)\widehat{Q} \cap (1 + c_G)\widetilde{Q} \neq \emptyset.$$

Assume also that both $(1+c_G)\widehat{Q}$ and $(1+c_G)\widetilde{Q}$ intersect $(1+c_G)Q_0$. Suppose that $x_1 \in E \cap \widehat{Q}^{**}, x_2 \in E \cap \widetilde{Q}^{**}$, and that for $\nu = 1, 2$,

$$(81)\qquad \textit{Either } x_\nu = x_0 \textit{ and } P_\nu = P_0, \textit{ or else } P_\nu = \text{Find-Neighbor}(P_0, \mathcal{A}_0, x_\nu)\,.$$

Then,

$$(82)\qquad P_1 - P_2 \in CM_0B(x_1, A_2\delta_{\widehat{Q}})$$

where C is a constant depending only on m and n.

Proof. Fix $\nu \in \{1,2\}$. We will show that

$$(83) \quad P_\nu \in \Gamma^\sharp_{\mathcal{A}_0}(x_\nu, P_0, CM_0)\,.$$

We split the proof of (83) to two cases, corresponding to the two cases in our assumption (81). Suppose first that $x_\nu = x_0$ and $P_\nu = P_0$. Then $P_\nu = P_0 \in \Gamma(x_\nu, \ell(\mathcal{A}_0), M_0)$ by (FN3). Using Property 1 from Section 13, we deduce that

$$(84) \quad P_\nu = P_0 \in \Gamma(x_\nu, \ell(\mathcal{A}_0) - 1, CM_0)\,.$$

Then (84) and the definition (48) of $\Gamma^\#$ imply that

$$P_\nu = P_0 \in \Gamma^\sharp_{\mathcal{A}_0}(x_\nu, P_0, CM_0).$$

Therefore (83) is proven in the case where $x_\nu = x_0$ and $P_\nu = P_0$.

It remains to handle the second case of our assumption (81); that is, when $P_\nu = \mathsf{Find\text{-}Neighbor}(P_0, \mathcal{A}_0, x_\nu)$. According to Lemma 7, the sets

$$\Gamma^\sharp_{\mathcal{A}_0}(x_1, P_0, CM_0) \quad \text{and} \quad \Gamma^\sharp_{\mathcal{A}_0}(x_2, P_0, CM_0)$$

are non-empty. Therefore, the basic property (80) of Find-Neighbor implies that

$$P_\nu \in \Gamma^\sharp_{\mathcal{A}_0}(x_\nu, P_0, C'M_0).$$

Hence (83) is proven also in the case where $P_\nu = \mathsf{Find\text{-}Neighbor}(P_0, \mathcal{A}_0, x_\nu)$.

Thus (83) holds in all cases. We may thus apply Lemma 8, based on (83) for $\nu = 1, 2$. Lemma 8 implies (82) and the lemma is proven. ■

Remark. Suppose that $x \in \widehat{Q}^{**} \cap E$ for some cube $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ such that

$$(1 + c_G)\widehat{Q} \cap (1 + c_G)Q_0 \neq \emptyset.$$

Suppose also that either $x = x_0, P = P_0$ or else $P = \mathsf{Find\text{-}Neighbor}(P_0, \mathcal{A}_0, x)$. Then we actually proved in Lemma 9 (see (83)) that

$$(85) \quad P \in \Gamma^\sharp_{\mathcal{A}_0}(x, P_0, CM_0)\,.$$

We will use the last remark, as well as Lemma 7 and Lemma 9, in the next section. Recall that (FN1), (FN2) and (FN3) were the fundamental assumptions in the present section. Thus, when we apply Lemma 7, Lemma 9 or other results from the current section, we have to make sure that (FN1), (FN2) and (FN3) hold.

33. The proof of the *Main Lemma*

Recall that $\mathcal{P}^+$ stands for the space of all polynomials of degree m in $\mathbb{R}^n$. The space $\mathcal{P}$, of all polynomials of degree $m-1$ in $\mathbb{R}^n$, embeds naturally in $\mathcal{P}^+$. Recall also that we denote, for $x \in \mathbb{R}^n, \delta > 0$,

$$B^+(x,\delta) = \{P \in \mathcal{P}^+ : |\partial^\beta P(x)| \le \delta^{m-|\beta|} \text{ for all } |\beta| \le m\}.$$

Clearly, for $P \in \mathcal{P}$, we have $P \in B^+(x,\delta)$ if and only if $P \in B(x,\delta)$. Recall that $J_x^+(F)$ stands for the m-jet at x of a function $F : \mathbb{R}^n \to \mathbb{R}$.

We will prove the *Main Lemma for* $\mathcal{A}_0$ by induction on the set $\mathcal{A}_0$ with respect to the order relation $<$. The minimal set in the order is $\mathcal{M}$. Next, we establish the *Main Lemma for* $\mathcal{M}$, the base of our induction.

33.1. The case $\mathcal{A}_0 = \mathcal{M}$

Recall the assumptions of the *Main Lemma for* $\mathcal{M}$ from Section 29. Thus, assume that we are given a dyadic cube Q_0 with $\delta_{Q_0} \le A_2^{-1}$, a polynomial $P_0 \in \mathcal{P}$, $M_0 > 0$ and $x_0 \in \mathbb{R}^n$ that satisfy:

(AM1) $x_0 \in E \cap Q_0^{**}$. If $E \cap Q_0^* \neq \emptyset$, then $x_0 \in E \cap Q_0^*$.

(AM2) $Q_0 \in CZ(\mathcal{M})$.

(AM3) $P_0 \in \Gamma(x_0, \ell(\mathcal{M}), M_0) = \Gamma(x_0, 1, M_0)$.

To establish the lemma, we need to exhibit a function $F \in C^m((1+c_G)Q_0)$ such that:

(1) $J_x^+(F) = f_x(\mathcal{M}, Q_0, x_0, P_0)$ for all $x \in (1+c_G)Q_0$,

where $f_x(\mathcal{M}, Q_0, x_0, P_0)$ is defined in the *Main Algorithm* from Section 29,

(2) $J_x^+(F - P_0) \in A_3(\mathcal{M})M_0 \cdot B^+(x, \delta_{Q_0})$ for all $x \in (1+c_G)Q_0$,

(3) $J_x(F) \in \Gamma(x, 0, A_3(\mathcal{M})M_0)$ for all $x \in E \cap (1+c_G)Q_0$,

(4) If $x_0 \in (1+c_G)Q_0$ then $J_{x_0}(F) = P_0$.

To that end, we set

(5) $F(x) = P_0(x)$ for all $x \in (1+c_G)Q_0$,

a polynomial on $(1+c_G)Q_0$. Thus,

(6) $J_x^+(F) = P_0$ for all $x \in (1+c_G)Q_0$,

and therefore (2) and (4) trivially hold.

According to `Line 1` of the *Main Algorithm*, we know that

$$P_0 = f_x(\mathcal{M}, Q_0, x_0, P_0), \quad \text{for all } x \in (1 + c_G)Q_0,$$

and consequently (1) holds.

It remains to establish (3). According to (6), we need to prove that

$$(7) \qquad P_0 \in \Gamma(x, 0, A_3(\mathcal{M})M_0) \quad \text{for all } x \in (1 + c_G)Q_0 \cap E\,.$$

By (AM2) we know that $Q_0 \in CZ(\mathcal{M})$, and hence the cube Q_0 is almost OK($\mathcal{M}$). Recall the definition of "almost OK($\mathcal{M}$)" from Section 20. Since Q_0 is almost OK($\mathcal{M}$), then either

Case 1: $\#(E \cap Q_0^*) \leq 1$,

or

Case 2: Q_0 is OK($\mathcal{A}$) for some $\mathcal{A} \leq \mathcal{M}$ and $\#(E \cap Q_0^*) > 1$.

Suppose we are in Case 1. If $E \cap Q_0^* = \emptyset$ then (7) holds vacuously as $E \cap (1 + c_G)Q_0 \subseteq E \cap Q_0^* = \emptyset$. Otherwise, $\#(E \cap Q_0^*) = 1$. According to (AM1), the point x_0 is the unique element in $E \cap Q_0^*$. According to (AM3), we know that

$$(8) \qquad P_0 \in \Gamma(x_0, 1, M_0) \subseteq \Gamma(x_0, 0, CM_0)$$

where the inclusion follows by Property 4 from Section 13. The definitions of $A_3(\mathcal{M})$ and A_0 in Section 17 imply that

$$(9) \qquad A_3(\mathcal{M}) = A_0^2 A_1(\mathcal{M}) \geq A_0 \geq C\,,$$

where C is the constant from (8). Since $(1 + c_G)Q_0 \subseteq Q_0^*$, and x_0 is the unique point in $E \cap Q_0^*$, then (8) and (9) imply (7). This finishes Case 1.

We move our attention to Case 2. Then $\#(E \cap Q_0^*) > 1$ and Q_0 is OK($\mathcal{A}$), for some $\mathcal{A} \leq \mathcal{M}$. As $\mathcal{M}$ is minimal, Q_0 is OK($\mathcal{M}$). Let us pick

$$(10) \qquad x \in E \cap (1 + c_G)Q_0 \subseteq E \cap Q_0^*\,.$$

Recall the definition of OK($\mathcal{M}$), that is (1) from Section 20. Since Q is OK($\mathcal{M}$) and $x \in E \cap Q_0^*$, then according to (1) from Section 20,

$$A_2 \delta_{Q_0} \leq \delta(x, \mathcal{M}),$$

and from the definition of $\delta(x, \mathcal{M})$ (e.g., (24) of Section 31) we know that

$$(11) \qquad B_{\mathcal{M}}(A_2 \delta_{Q_0}) \subseteq A_0 A_1(\mathcal{M}) \pi_{\mathcal{M},x}\{\sigma(x, 1)\}\,.$$

Recall that $\pi_{\mathcal{M},x}$, as defined in (1) from Section 30, is an isomorphism of $\mathcal{P}$ and $\mathbb{R}^{\sharp(\mathcal{M})}$. Thus, applying $\pi^{-1}_{\mathcal{M},x}$ to both sides in (11) we get that

$$(12) \quad \pi^{-1}_{\mathcal{M},x}\{B_{\mathcal{M}}(A_2\delta_{Q_0})\} \subseteq A_0A_1(\mathcal{M})\sigma(x,1)\,.$$

According to (3) from Section 30, the left hand side of (12) is C-equivalent to $B(x, A_2\delta_{Q_0})$. Therefore,

$$(13) \quad B(x, A_2\delta_{Q_0}) \subseteq CA_0A_1(\mathcal{M})\sigma(x,1)\,.$$

Both x and x_0 belong to Q_0^{**}, by (AM1) and (10). Thus

$$(14) \quad |x - x_0| \leq \sqrt{n}\delta_{Q_0^{**}} \leq C\delta_{Q_0} \leq A_0\delta_{Q_0} \leq A_2\delta_{Q_0},$$

according to the definition of A_0, A_2 from Section 17. By (13) and (14),

$$(15) \quad B(x, x_0) \subseteq B(x, A_2\delta_{Q_0}) \subseteq CA_0A_1(\mathcal{M})\sigma(x,1)\,.$$

Next, according to (AM3) followed by Property 2 from Section 13,

$$(16) \quad P_0 \in \Gamma(x_0, 1, M_0) \subseteq \Gamma(x, 0, CM_0) + CM_0B(x, x_0)\,.$$

Combining (15) with (16), we see that

$$(17) \quad P_0 \in \Gamma(x, 0, CM_0) + CA_0A_1(\mathcal{M})M_0\sigma(x,1)\,.$$

By applying Property 4 and Property 1 from Section 13 to (17), we conclude that

$$(18) \quad P_0 \in \Gamma\left(x, 0, CA_0A_1(\mathcal{M})M_0\right)\,.$$

Assume, as we may, that

$$(19) \quad A_0 > C \text{ where } C \text{ is the constant from (18)}\,.$$

According to (3) from Section 17, we have that $A_3(\mathcal{M}) = A_0^2A_1(\mathcal{M}) > CA_0A_1(\mathcal{M})$, where C is the constant from (18). From (18) we thus conclude that

$$P_0 \in \Gamma\left(x, 0, A_3(\mathcal{M})M_0\right).$$

The point $x \in (1 + c_G)Q_0 \cap E$ is arbitrary, and hence (7) is proven. This completes the proof of the *Main Lemma for* $\mathcal{M}$.

33.2. The *Main Lemma* in an easy case

We have established the base of the induction. Let $\mathcal{A}_0 \subset \mathcal{M}$ be such that $\mathcal{A}_0 \neq \mathcal{M}$. Assume that the Main Lemma was proven for all $\mathcal{A} < \mathcal{A}_0$. Let us now prove the *Main Lemma for* $\mathcal{A}_0$.

Thus, suppose we are given a dyadic cube Q_0 with $\delta_{Q_0} \leq A_2^{-1}$, a polynomial $P_0 \in \mathcal{P}$, $M_0 > 0$ and $x_0 \in \mathbb{R}^n$ that satisfy:

(ML1) $x_0 \in E \cap Q_0^{**}$. If $E \cap Q_0^* \neq \emptyset$, then $x_0 \in E \cap Q_0^*$.

(ML2) $Q_0 \in CZ(\mathcal{A}_0)$.

(ML3) $P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0)$.

To establish the *Main Lemma for* $\mathcal{A}_0$, we need to exhibit a function $F \in C^m((1 + c_G)Q_0)$ such that:

(MLC1) $J_x^+(F) = f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ for all $x \in (1 + c_G)Q_0$,

where $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ is defined in the *Main Algorithm* from Section 29,

(MLC2) $J_x^+(F - P_0) \in A_3(\mathcal{A}_0)M_0 \cdot B^+(x, \delta_{Q_0})$ for all $x \in (1 + c_G)Q_0$,

(MLC3) $J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A}_0)M_0)$ for all $x \in E \cap (1 + c_G)Q_0$,

(MLC4) If $x_0 \in (1 + c_G)Q_0$ then $J_{x_0}(F) = P_0$.

We split the proof into two cases, according to whether there exists $\mathcal{A} < \mathcal{A}_0$ such that $Q_0 \in CZ(\mathcal{A})$, or whether there is no such $\mathcal{A}$. We will next treat the first, easy, case.

Thus, suppose there exists $\mathcal{A} < \mathcal{A}_0$ such that $Q_0 \in CZ(\mathcal{A})$. We may assume that $\mathcal{A} < \mathcal{A}_0$ is the minimal subset of $\mathcal{M}$, with respect to our order relation, such that

(ML2′) $Q_0 \in CZ(\mathcal{A})$.

In particular $\ell(\mathcal{A}) < \ell(\mathcal{A}_0)$ and hence (ML3) and Property 4 from Section 13 imply that

(ML3′) $P_0 \in \Gamma(x_0, \ell(\mathcal{A}), CM_0) = \Gamma(x_0, \ell(\mathcal{A}), M_0')$

where

(20) $M_0' = CM_0\,.$

Note that (ML1), (ML2′) and (ML3′) are precisely the assumptions of the *Main Lemma for* $\mathcal{A}$, with M_0' in place of M_0. Since $\mathcal{A} < \mathcal{A}_0$, we may apply the induction hypothesis, to get a function $F \in C^m((1 + c_G)Q_0)$ such that:

(21) $\quad J^+_x(F) = f_x(\mathcal{A}, Q_0, x_0, P_0)$ for all $x \in (1 + c_G)Q_0\,,$

(22) $\quad J^+_x(F - P_0) \in A_3(\mathcal{A})M'_0 \cdot B^+(x, \delta_{Q_0})$ for all $x \in (1 + c_G)Q_0\,,$

(23) $\quad J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A})M'_0)$ for all $x \in E \cap (1 + c_G)Q_0\,,$

(24) $\quad$ If $x_0 \in (1 + c_G)Q_0$ then $J_{x_0}(F) = P_0\,.$

We will show that the function F satisfies the conclusions of the *Main Lemma for* $\mathcal{A}_0$. That is, we will establish (MLC1), (MLC2), (MLC3) and (MLC4).

First, (MLC4) holds, because of (24). Since $\mathcal{A} < \mathcal{A}_0$, then by the definitions of $A_3(\mathcal{A}_0)$ and A_0 from Section 17, we have that

(25) $\quad A_3(\mathcal{A}_0) \geq A_0 A_3(\mathcal{A}) > CA_3(\mathcal{A})$ where C is the constant from (20) .

According to (20) and (25), we immediately conclude that (22) implies (MLC2), and that (23) implies (MLC3). It remains to prove (MLC1).

Recall the *Main Algorithm* from Section 29. According to `Lines 2-3` in the *Main Algorithm*, since $\mathcal{A}$ is the minimal subset of $\mathcal{M}$ such that $Q_0 \in CZ(\mathcal{A})$, then

(26) $\quad f_x(\mathcal{A}_0, Q_0, x_0, P_0) = f_x(\mathcal{A}, Q_0, x_0, P_0)\,.$

Now (21) and (26) imply (MLC1). We have thus proven that the function F satisfies the conclusions (MLC1), (MLC2), (MLC3) and (MLC4). Therefore the *Main Lemma for* $\mathcal{A}_0$ is proven in the case where there exists $\mathcal{A} < \mathcal{A}_0$ such that $Q_0 \in CZ(\mathcal{A})$.

33.3. The *Main Lemma* in the non-trivial case

In this section we prove the *Main Lemma for* $\mathcal{A}_0$ in the remaining case, where there is no $\mathcal{A} < \mathcal{A}_0$ with $Q_0 \in CZ(\mathcal{A})$. Therefore, we assume here, in addition to (ML1), (ML2) and (ML3), that

(ML4) $\quad \mathcal{A}_0 \neq \mathcal{M}$, and for all $\mathcal{A} < \mathcal{A}_0$, we have that $Q_0 \notin CZ(\mathcal{A})$.

Our goal is to prove the conclusions of the *Main Lemma for* $\mathcal{A}_0$, i.e., the existence of a function $F \in C^m((1 + c_G)Q_0)$ that satisfies (MLC1), (MLC2), (MLC3) and (MLC4) from Section 33.2.

Let $Q_1, ..., Q_{k_{max}}$ be an enumeration of all cubes $Q \in CZ(\mathcal{A}_0^-)$ such that $(1 + c_G)Q \cap (1 + c_G)Q_0 \neq \emptyset$. For each cube Q_k, we will define a point $x_k \in Q_k^{**}$ and a polynomial $P_k \in \mathcal{P}$. Later on, we will apply the induction hypothesis for the cubes Q_k, the points x_k and the polynomials P_k. Fix $1 \leq k \leq k_{max}$. To define x_k and P_k in the case where $Q_k^{**} \cap E = \emptyset$, we simply set

(27) $\quad x_k =$ center of $Q_k \quad$ and $\quad P_k = P_0.$

Clearly $x_k \in Q_k^{**}$ in this case. We still need to define x_k and P_k when $Q_k^{**} \cap E \neq \emptyset$. Thus, suppose that $Q_k^{**} \cap E \neq \emptyset$. If $x_0 \in Q_k^*$, then we set

(28) $\quad x_k = x_0 \quad$ and $\quad P_k = P_0.$

(Obviously $x_k = x_0 \in Q_k^* \cap E \subseteq Q_k^{**}$ here.) In the case where $x_0 \notin Q_k^*$, we define

(29) $\quad x_k = \mathsf{Find\text{-}Representative}(Q_k) \quad$ and $\quad P_k = \mathsf{Find\text{-}Neighbor}(P_0, \mathcal{A}_0, x_k),$

where Algorithm Find-Representative is described in Section 27, and Algorithm Find-Neighbor is presented in Section 15. The defining property of Find-Representative from Section 27 shows that $x_k \in Q_k^{**}$. This completes the definition of x_k and P_k in all cases.

Thus, for each $1 \leq k \leq k_{max}$ we have defined a representative $x_k \in Q_k^{**}$ and a polynomial $P_k \in \mathcal{P}$. The representative x_k satisfies that $x_k \in E$ whenever $Q_k^{**} \cap E \neq \emptyset$; this follows at once by (28), (29) and the defining property of Find-Representative from Section 27.

In the next two lemmas, we will make use of Lemma 7, Lemma 9 and property (85) from Section 32. Note that the basic assumptions (FN1), (FN2) and (FN3) from Section 32 hold, in view of (ML1), (ML2), (ML3) and (ML4). Therefore we may safely use results from Section 32 (see also the last paragraph in Section 32).

Lemma 1. *Let* $1 \leq k \leq k_{max}$. *Then,*

(30) $\quad P_k - P_0 \in CM_0 B(x_k, A_2 \delta_{Q_0})\,.$

Furthermore, if $Q_k^{**} \cap E \neq \emptyset$, *then*

(31) $\quad P_k \in \Gamma(x_k, \ell(\mathcal{A}_0^-), CM_0).$

Here, C *is a constant depending only on* m *and* n.

Proof. Suppose first that $E \cap Q_k^{**} = \emptyset$. Then $P_k = P_0$ according to (27), and hence (30) trivially holds. Therefore the lemma is proven for the case where $E \cap Q_k^{**} = \emptyset$, and we may thus confine our attention to the case where $E \cap Q_k^{**} \neq \emptyset$. Consequently, $x_k \in Q_k^{**} \cap E$, and the cube $Q_k \in CZ(\mathcal{A}_0^-)$ satisfies $(1 + c_G)Q_k \cap (1 + c_G)Q_0 \neq \emptyset$. According to (28), (29) we either have that $x_k = x_0, P_k = P_0$, or else $P_k = \mathsf{Find\text{-}Neighbor}(P_0, \mathcal{A}_0, x_k)$. We may thus invoke (85) from Section 32, and conclude that

(32) $\quad P_k \in \Gamma^\sharp_{\mathcal{A}_0}(x_k, P_0, CM_0)\,.$

Since $Q_k \in CZ(\mathcal{A}_0^-), x_k \in Q_k^{**} \cap E$ and $(1 + c_G)Q_k \cap (1 + c_G)Q_0 \neq \emptyset$, then the requirements of Lemma 7 from Section 32 are satisfied. From (32) and from the conclusion of that lemma (the “Moreover” part), we deduce that

(33) $\quad P_k - P_0 \in C' M_0 B(x_k, A_2 \delta_{Q_0})\,.$

Therefore (30) is proven.

We move to the proof of (31). Recall the definition of $\Gamma^{\#}$, that appears in (48) from Section 32. According to (32),

$$(34) \quad P_k \in \Gamma(x_k, \ell(\mathcal{A}_0) - 1, CM_0)\,.$$

Since $\ell(\mathcal{A}_0^-) \leq \ell(\mathcal{A}_0) - 1$, then by combining (34) with Property 4 from Section 13, we get that

$$(35) \quad P_k \in \Gamma(x_k, \ell(\mathcal{A}_0^-), C'M_0)\,.$$

This completes the proof of (31). The lemma is thus proven. ■

Lemma 2. *Let* $1 \leq \mu, \nu \leq k_{max}$ *be such that* $(1+c_G)Q_\mu \cap (1+c_G)Q_\nu \neq \emptyset$. *Then,*

$$(36) \quad P_\mu - P_\nu \in CM_0 B(x_\mu, A_2\delta_{Q_\mu})\,,$$

where C *is a constant depending only on* m *and* n.

Proof. Suppose first that $E \cap Q_\mu^{**} = \emptyset$ and $E \cap Q_\nu^{**} = \emptyset$. In this case, by (27),

$$P_\mu = P_0, \qquad P_\nu = P_0$$

and (36) trivially holds.

Next, suppose that $E \cap Q_\mu^{**} = \emptyset$ but $E \cap Q_\nu^{**} \neq \emptyset$. Since $E \cap Q_\mu^{**} = \emptyset$, then $P_\mu = P_0$ by (27). Additionally, by Lemma 4 from Section 21, we know that

$$(37) \quad \delta_{Q_\mu} = A_2^{-1}\,.$$

According to Lemma 1,

$$(38) \quad P_\nu - P_0 \in CM_0 B(x_\nu, A_2\delta_{Q_0}) \subseteq CM_0 B(x_\nu, A_2\delta_{Q_\mu})\,,$$

since $\delta_{Q_\mu} = A_2^{-1} \geq \delta_{Q_0}$. Recall that $x_\nu \in Q_\nu^{**}, x_\mu \in Q_\mu^{**}$ and that the dyadic cubes Q_μ, Q_ν satisfy $(1+c_G)Q_\mu \cap (1+c_G)Q_\nu \neq \emptyset$. Since $\delta_{Q_\mu} = A_2^{-1} \geq \delta_{Q_\nu}$ then $|x_\nu - x_\mu| < C\delta_{Q_\mu}$. Consequently, (38) translates, with the help of (3) from Section 12, to

$$(39) \quad P_\nu - P_\mu = P_\nu - P_0 \in CM_0 B(x_\mu, A_2\delta_{Q_\mu})\,,$$

and (36) is established, in the case where $E \cap Q_\mu^{**} = \emptyset, E \cap Q_\nu^{**} \neq \emptyset$.

Note that since $(1+c_G)Q_\mu \cap (1+c_G)Q_\nu \neq \emptyset$, and $Q_\mu, Q_\nu \in CZ(\mathcal{A}_0^-)$, then Lemma 2 of Section 21 implies that the sidelengths δ_{Q_ν} and δ_{Q_μ} have the same order of magnitude. Furthermore, since $x_\mu \in Q_\mu^{**}, x_\nu \in Q_\nu^{**}$, then

$$(40) \quad |x_\mu - x_\nu| < C\delta_{Q_\nu}\,.$$

We conclude that $B(x_\nu, A_2\delta_{Q_\nu})$ and $B(x_\mu, A_2\delta_{Q_\mu})$ are C-equivalent. Therefore, (36) is actually symmetric in μ and ν; reversing their rôles merely changes the constant C in (36). By this symmetry, the lemma is also proven for the case where $E \cap Q_\mu^{**} \neq \emptyset, E \cap Q_\nu^{**} = \emptyset$.

It remains to consider the case where both $E \cap Q_\mu^{**}$ and $E \cap Q_\nu^{**}$ are nonempty. Let us verify the requirements of Lemma 9 from Section 32, with $Q_\mu, Q_\nu, x_\mu, x_\nu, P_\mu, P_\nu$ in place of $\widehat{Q}, \widetilde{Q}, x_1, x_2, P_1, P_2$. By their definition, Q_μ and Q_ν belong to $CZ(\mathcal{A}_0^-)$. One of the assumptions of the present lemma was that $(1+c_G)Q_\mu \cap (1+c_G)Q_\nu \neq \emptyset$. Note also that the list $Q_1, \ldots, Q_{k_{max}}$ is defined to consist of all cubes $Q \in CZ(\mathcal{A}_0^-)$ such that $(1+c_G)Q \cap (1+c_G)Q_0 \neq \emptyset$. Consequently, $(1+c_G)Q_0$ intersects both $(1+c_G)Q_\mu$ and $(1+c_G)Q_\nu$. In addition, we know that $x_\mu \in E \cap Q_\mu^{**}$ and $x_\nu \in E \cap Q_\nu^{**}$. Hence, Q_μ, Q_ν, x_μ and x_ν satisfy the requirements of Lemma 9 from Section 32. Thanks to (28), (29), also the polynomials P_μ and P_ν satisfy the assumptions of Lemma 9 from Section 32. Therefore, we may apply that lemma, and conclude that,

$$P_\mu - P_\nu \in CM_0 B(x_\mu, A_2\delta_{Q_\mu}).$$

Thus (36) is established, and the lemma is proven. ■

We have defined the cubes $Q_1, \ldots, Q_{k_{max}} \in CZ(\mathcal{A}_0^-)$, and to each cube we have associated a point $x_k \in Q_k^{**}$ and a polynomial $P_k \in \mathcal{P}$. Next, we will construct certain functions $F_k \in C^m((1+c_G)Q_k)$. Fix $1 \leq k \leq k_{max}$. Suppose first that $Q_k^{**} \cap E = \emptyset$. In this case we simply set

(41) $F_k = P_0$.

We obviously have that $F_k \in C^m((1+c_G)Q_k)$. In order to define F_k in the case where $Q_k^{**} \cap E \neq \emptyset$, we will invoke the induction hypothesis, the *Main Lemma for* $\mathcal{A}_0^-$. Thus, suppose that $Q_k^{**} \cap E \neq \emptyset$. We know that

(42) $Q_k \in CZ(\mathcal{A}_0^-)$.

Next, we claim that

(43) $x_k \in E \cap Q_k^{**}$, with $x_k \in E \cap Q_k^*$ whenever $E \cap Q_k^* \neq \emptyset$.

Indeed, if $x_0 \in Q_k^*$ then $x_k = x_0 \in Q_k^*$ by (28), and (43) obviously holds. If $x_0 \notin Q_k^*$, then (43) follows from (29) and the defining property of Find-Representative from Section 27. Thus, we have proved (43) in all cases.

Let us set

(44) $M_0' = CM_0 > 0$

where C is the constant from (31).

According to (42), (43), (44) and (31), the cube Q_k, the point x_k, the polynomial P_k and the positive number M_0' satisfy the requirements of the *Main Lemma for* $\mathcal{A}_0^-$. Since $\mathcal{A}_0^- < \mathcal{A}_0$, by the induction hypothesis we may apply the *Main Lemma for* $\mathcal{A}_0^-$. According to the conclusion of the *Main Lemma for* $\mathcal{A}_0^-$, there exists a function $F_k \in C^m((1+c_G)Q_k)$ with the following properties:

(45) $J_x^+(F_k - P_k) \in A_3(\mathcal{A}_0^-)M_0' \, B^+(x, \delta_{Q_k})$ for all $x \in (1+c_G)Q_k$,

(46) $J_x(F_k) \in \Gamma(x, 0, A_3(\mathcal{A}_0^-)M_0')$ for all $x \in E \cap (1+c_G)Q_k$,

(47) $J_x^+(F_k) = f_x(\mathcal{A}_0^-, Q_k, x_k, P_k)$ for all $x \in (1+c_G)Q_k$,

where $f_x(\mathcal{A}_0^-, Q_k, x_k, P_k)$ is defined by the *Main Algorithm*, and

(48) If $x_k \in (1+c_G)Q_k$ then $J_{x_k}(F_k) = P_k$.

This completes the definition of the function $F_k \in C^m((1+c_G)Q_k)$ in the case where $Q_k^{**} \cap E \neq \emptyset$. Therefore, $F_k \in C^m((1+c_G)Q_k)$ is defined for all $1 \leq k \leq k_{max}$. We summarize the properties of the functions F_k in the following lemma.

Lemma 3. *Let* $1 \leq k \leq k_{max}$. *Then,*

(FK1) $J_x^+(F_k - P_k) \in CA_3(\mathcal{A}_0^-)M_0 B^+(x, \delta_{Q_k})$ *for all* $x \in (1+c_G)Q_k$,

(FK2) $J_x(F_k) \in \Gamma(x, 0, CA_3(\mathcal{A}_0^-)M_0)$ *for all* $x \in E \cap (1+c_G)Q_k$,

(FK3) *If* $x_0 \in (1+c_G)Q_k$ *then* $J_{x_0}(F_k) = P_0$.

Furthermore, if $Q_k^{**} \cap E \neq \emptyset$, *then,*

(FK4) $J_x^+(F_k) = f_x(\mathcal{A}_0^-, Q_k, x_k, P_k)$ for all $x \in (1+c_G)Q_k$,

where $f_x(\mathcal{A}_0^-, Q_k, x_k, P_k)$ *is defined by the Main Algorithm. Here* C *is a constant depending only on* m *and* n.

Proof. Suppose first that $Q_k^{**} \cap E \neq \emptyset$. Recall that $M_0' = CM_0$, according to (44). Then (FK1), (FK2) and (FK4) follow from (45), (46) and (47), respectively. It remains to prove (FK3). Suppose $x_0 \in (1+c_G)Q_k \subseteq Q_k^*$. According to the definition (28), we have that $x_k = x_0$ and $P_k = P_0$. Therefore $x_k \in (1+c_G)Q_k$, and by (48) necessarily

$$J_{x_0}(F_k) = J_{x_k}(F_k) = P_k = P_0,$$

and (FK3) follows. The lemma is thus proven in the case where $Q_k^{**} \cap E \neq \emptyset$. Suppose now that $Q_k^{**} \cap E = \emptyset$. Then (FK2) vacuously holds. Additionally, $F_k = P_k = P_0$ by (41) and (27), and hence (FK1), (FK3) trivially hold. This completes the proof. ■

Recall the functions $\theta^{\mathcal{A}_0^-}_{Q_k}$ $(k = 1, ..., k_{max})$ from Section 28. Fix $1 \leq k \leq k_{max}$. According to the discussion in Section 28, the function $\theta^{\mathcal{A}_0^-}_{Q_k}$ is a C^m-function whose support is contained in $(1 + c_G/2)Q_k$. Since $F_k \in C^m((1 + c_G)Q_k)$, then $\theta^{\mathcal{A}_0^-}_{Q_k}(x)F_k(x)$ is a C^m-function on the entire $\mathbb{R}^n$. Next, for any $x \in (1 + 2c_G)Q_0$, we set

$$(49) \quad F(x) = \sum_{k=1}^{k_{max}} \theta^{\mathcal{A}_0^-}_{Q_k}(x)F_k(x).$$

Then F is a C^m-function on $(1 + c_G)Q_0$, since it is a finite sum of C^m-functions. We will see that F satisfies (MLC1),..., (MLC4) from Section 33.2. Note that it follows from (49) that for any $x \in (1 + c_G)Q_0$,

$$(50) \quad J^+_x(F) = \sum_{k=1}^{k_{max}} J^+_x\left(\theta^{\mathcal{A}_0^-}_{Q_k}\right) \odot^+_x J^+_x(F_k).$$

Lemma 4. *Let* $x \in (1 + c_G)Q_0$. *Then,*

(MLC1′) $J^+_x(F) = f_x(\mathcal{A}_0, Q_0, x_0, P_0)$, *where* $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ *is defined by the Main Algorithm in Section* 29.

(MLC2′) $J^+_x(F - P_0) \in A_3(\mathcal{A}_0)M_0 \, B^+(x, \delta_{Q_0})$.

(MLC3′) If $x \in E$ then $J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A}_0)M_0)$.

(MLC4′) If $x = x_0$ then $J_x(F) = P_0$.

Proof. We start with establishing (MLC1′). Recall the *Main Algorithm*. Recall also our assumption (ML4). Since $\mathcal{A}_0 \neq \mathcal{M}$, according to (ML4), then in the computation of $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$, the *Main Algorithm* reaches the execution of `Line 2`. According to (ML4), for all $\mathcal{A} < \mathcal{A}_0$ we have $Q_0 \notin CZ(\mathcal{A})$. Thus, the *Main Algorithm* reaches the execution of `Line 5`.

Denote by L the list of the cubes that are being produced in `Lines 5-6` of the *Main Algorithm* in the course of computing $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$. According to `Lines 5-6` of the *Main Algorithm*, we know that

$$(51) \quad L = \{Q \in CZ(\mathcal{A}_0^-) \ : \ x \in (1 + c_G)Q\}.$$

Since $x \in (1 + c_G)Q_0$, we conclude from (51) that $(1 + c_G)Q_0$ intersects $(1 + c_G)Q$ for all $Q \in L$. Consequently, all cubes in L appear in the list $Q_1, ..., Q_{k_{max}}$. (Recall the definition of $Q_1, ..., Q_{k_{max}}$ from the beginning of Section 33.3.) Furthermore, we claim that for any $1 \leq k \leq k_{max}$,

$$(52) \quad \text{If } x \in \mathrm{Supp}\left(\theta^{\mathcal{A}_0^-}_{Q_k}\right) \text{ then } Q_k \in L,$$

where, as usual, for any function g, we write $\mathrm{Supp}(g)$ to denote the closure of the set $\{x \in \mathbb{R}^n : g(x) \neq 0\}$.

Indeed, if $x \in \mathtt{Supp}\big(\theta^{\mathcal{A}_0^-}_{Q_k}\big)$ then $x \in (1+c_G)Q_k$ according to the discussion in Section 28. Since $Q_k \in CZ(\mathcal{A}_0^-)$, then (52) follows from (51). The cubes in L are enumerated in `Lines 5-6` of the *Main Algorithm* in the course of computing $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$. Let $Q_{k_1}, \ldots, Q_{k_{i_{max}}}$ be an enumeration of the cubes in L, that corresponds to the enumeration in `Lines 5-6` of the *Main Algorithm.* We conclude from (50) and (52) that

$$(53) \qquad J_x^+(F) = \sum_{i=1}^{i_{max}} J_x^+\left(\theta^{\mathcal{A}_0^-}_{Q_{k_i}}\right) \odot_x^+ J_x^+(F_{k_i}).$$

Indeed, if $Q_k \notin L$ then $\theta^{\mathcal{A}_0^-}_{Q_k}$ vanishes in a neighborhood of x, and hence (50) reduces to (53). Note that `Lines 8-15` of the *Main Algorithm* are being executed exactly once for each cube Q_{k_i} $(i = 1, \ldots, i_{max})$, during the computation of $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$. Fix $1 \le i \le i_{max}$. Consider the i^{th} execution of the loop in `Lines 8-15` of the *Main Algorithm.* In this execution of the loop, the *Main Algorithm* deals with the cube Q_{k_i}, and computes a a certain polynomial $f_{k_i} \in \mathcal{P}^+$. (The indexing in `Lines 8-15` is slightly different from here. The polynomial f_{k_i} is referred to as f_k in the *Main Algorithm.*) We claim that, for $1 \le i \le i_{max}$,

(54) If $Q_{k_i}^{**} \cap E = \emptyset$ then $f_{k_i} = P_0$, and otherwise $f_{k_i} = f_x(\mathcal{A}_0^-, Q_{k_i}, x_{k_i}, P_{k_i})$.

Indeed, consider first the case where $Q_{k_i}^{**} \cap E = \emptyset$. Then (54) holds according to `Line 8` of the *Main Algorithm.* Suppose now that $Q_{k_i}^{**} \cap E \neq \emptyset$. Observe that the definition of x_{k_i}, P_{k_i} in (28), (29) agrees with the computation of x_k, P_k in `Lines 9-11` of the *Main Algorithm.* In view of `Line 13` of the *Main Algorithm*, (54) holds also in the case where $Q_{k_i}^{**} \cap E \neq \emptyset$. Therefore (54) holds in all cases.

Recall the definition of the functions F_k, for $k = 1, \ldots, k_{max}$. Fix $1 \le i \le i_{max}$. If $Q_{k_i}^{**} \cap E = \emptyset$, then $F_{k_i} = P_0$ according to (41), and consequently $J_x^+(F_{k_i}) = P_0 = f_{k_i}$ by (54). If $Q_{k_i}^{**} \cap E \neq \emptyset$, then (FK4) and (54) show that $J_x^+(F_{k_i}) = f_{k_i}$. We conclude that

$$(55) \qquad J_x^+(F_{k_i}) = f_{k_i} \quad \text{for any} \quad 1 \le i \le i_{max}.$$

Therefore, by (53) and (55),

$$(56) \qquad J_x^+(F) = \sum_{i=1}^{i_{max}} J_x^+\left(\theta^{\mathcal{A}_0^-}_{Q_{k_i}}\right) \odot_x^+ f_{k_i}.$$

Inspection of `Line 16` of the *Main Algorithm* shows that the right hand side of (56) equals $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$. We conclude from (56) that

$$J_x^+(F) = f_x(\mathcal{A}_0, Q_0, x_0, P_0).$$

Thus (MLC1′) is proven.

Next, we will prove (MLC2′). Recall that $L = \left\{Q_{k_1}, ..., Q_{k_{i_{max}}}\right\}$ satisfies (51). By the Corollary to Lemma 2 from Section 21 we deduce that

(57) $i_{max} \leq C$.

Also, by (51), clearly $i_{max} = \#(L) \geq 1$, since $CZ(\mathcal{A}_0^-)$ is a tiling of $\mathbb{R}^n$. Set $\delta = \delta_{Q_{k_1}}$. Since $x \in (1 + c_G)Q_{k_i} \cap (1 + c_G)Q_{k_1}$ for $i = 1, ..., i_{max}$, then Lemma 2 from Section 21 implies that

(58) $\frac{1}{2}\delta \leq \delta_{Q_{k_i}} \leq 2\delta$ for any $i = 1, ..., i_{max}$.

According to the assumption of the present lemma, $x \in (1 + c_G)Q_0 \subseteq Q_0^{***}$. Since $x \in (1 + c_G)Q_{k_1}$ and $Q_{k_1} \in CZ(\mathcal{A}_0^-)$ then Lemma 6 from Section 21 implies that

(59) $\delta = \delta_{Q_{k_1}} \leq C\delta_{Q_0}$.

Next, we use property (10) from Section 28, pertaining to the functions $\theta_{Q_k}^{\mathcal{A}_0^-}$. According to that property, for any $i = 1, ..., i_{max}$,

(60) $J_x^+\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \in C\delta_{Q_{k_i}}^{-m} B^+(x, \delta_{Q_{k_i}}) \subseteq C'\delta^{-m}B^+(x, \delta)$,

where the last inclusion follows from (58). Since the θ's are a partition of unity, their sum is one, and (53) implies that

(61) $J_x^+(F) = P_{k_1} + \sum_{i=1}^{i_{max}} J_x^+\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \odot_x^+ [J_x^+(F_{k_i}) - P_{k_1}]$.

Recall that $x \in (1 + c_G)Q_{k_i}$ for each $i = 1, ..., i_{max}$. According to (FK1), for any $i = 1, ..., i_{max}$,

(62) $J_x^+(F_{k_i} - P_{k_i}) \in CA_3(\mathcal{A}_0^-)M_0\ B^+(x, \delta_{Q_{k_i}}) \subseteq C'A_3(\mathcal{A}_0^-)M_0B^+(x, \delta)$,

where the last inclusion follows from (58). Furthermore, for all $i = 1, ..., i_{max}$ we have that $(1 + c_G)Q_{k_i} \cap (1 + c_G)Q_{k_1} \neq \emptyset$, since both cubes contain x. Lemma 2 implies that for $i = 1, ..., i_{max}$,

(63) $P_{k_i} - P_{k_1} \in CM_0\ B(x_{k_1}, A_2\delta_{Q_{k_1}}) = CM_0\ B(x_{k_1}, A_2\delta)$.

Since $x_{k_1}, x \in Q_{k_1}^{**}$, we know that $|x - x_{k_1}| < C\delta$. Consequently, (63) together with (2) ,(3) from Section 12 entail that for any $i = 1, ..., i_{max}$,

(64) $P_{k_i} - P_{k_1} \in C'M_0\ B(x, A_2\delta) \subseteq \tilde{C}A_2^mM_0B(x, \delta) \subseteq CA_2^mM_0B^+(x, \delta)$.

Using (62) and (64), we may state that, for any $i = 1, ..., i_{max}$,

(65) $J_x^+(F_{k_i}) - P_{k_1} \in \tilde{C}A_2^mA_3(\mathcal{A}_0^-)M_0B^+(x, \delta)$.

From (60) and (65), with the help of (9) from Section 12, we get that for any $i = 1, \dots, i_{max}$,

$$(66) \quad J_x^+\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \odot_x^+ [J_x^+(F_{k_i}) - P_{k_1}] \in CA_2^m A_3(\mathcal{A}_0^-) M_0 B^+(x, \delta)\,.$$

Combining (57), (61) and (66), we obtain

$$(67) \quad J_x^+(F) \in P_{k_1} + CA_2^m A_3(\mathcal{A}_0^-) M_0 B^+(x, \delta)\,.$$

Next, according to Lemma 1, we know that,

$$(68) \quad P_{k_1} - P_0 \in CM_0 B(x_{k_1}, A_2 \delta_{Q_0}) \subseteq CA_2^m M_0 B(x_{k_1}, \delta_{Q_0})\,,$$

where we used (2) from Section 12. Since $x, x_{k_1} \in Q_{k_1}^{**}$, then $|x_{k_1} - x| \leq C\delta \leq C'\delta_{Q_0}$ by (59). Consequently, (68), together with (3) from Section 12, imply that

$$(69) \quad P_{k_1} - P_0 \in C'A_2^m M_0 B(x, \delta_{Q_0}) \subseteq C'A_2^m M_0 B^+(x, \delta_{Q_0})\,.$$

By combining (59), (67) and (69) we conclude that,

$$(70) \quad J_x^+(F) - P_0 \in CA_2^m A_3(\mathcal{A}_0^-) M_0 B^+(x, \delta_{Q_0})\,.$$

Assume, as we may from the discussion in Section 17, that the constant A_0 satisfies

(71) $A_0 > C$ where C is the constant from (70).

According to (3) of Section 17, we know that $A_3(\mathcal{A}_0) = A_0 A_2^m A_3(\mathcal{A}_0^-)$. Thus (70) and (71) imply that

$$(72) \quad J_x^+(F - P_0) \in A_3(\mathcal{A}_0) M_0 \; B^+(x, \delta_{Q_0})\,,$$

since $J_x^+(P_0) = P_0$, and (MLC2′) is proven.

Next, we will establish (MLC3′). Suppose that $x \in E$. According to (FK2), for any $i = 1, \dots, i_{max}$,

$$(73) \quad J_x(F_{k_i}) \in \Gamma(x, 0, CA_3(\mathcal{A}_0^-) M_0)\,,$$

since $x \in (1 + c_G) Q_{k_i} \cap E$. By (73) and Property 1 from Section 13, for all $i = 1, \dots, i_{max}$,

$$(74) \quad J_x(F_{k_i}) - J_x(F_{k_1}) \in C'A_3(\mathcal{A}_0^-) M_0 \cdot \sigma(x, 0).$$

We apply (65) twice, and deduce that for $i = 1, \dots, i_{max}$,

$$(75) \quad J_x(F_{k_i} - F_{k_1}) = J_x(F_{k_i} - P_{k_1}) + J_x(P_{k_1} - F_{k_1}) \in 2\tilde{C} A_2^m A_3(\mathcal{A}_0^-) M_0 B(x, \delta)\,.$$

According to (74) and (75), for any $i = 1, ..., i_{max}$,

(76) $\quad J_x(F_{k_i} - F_{k_1}) \in C' A_2^m A_3(\mathcal{A}_0^-) M_0 \, [\sigma(x,0) \cap B(x,\delta)]\,.$

Recall that $\sigma(x,0)$ is Whitney t-convex at x with Whitney constant C, by Property 3 from Section 13. The definition of Whitney t-convexity (10) from Section 12, together with (60) and (76), implies that for any $i = 1, ..., i_{max}$,

(77) $\quad J_x\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \odot_x [J_x(F_{k_i} - F_{k_1})] \in \tilde{C} A_2^m A_3(\mathcal{A}_0^-) M_0 \cdot \sigma(x,0).$

Next, we rewrite (53), discarding some information, as

(78) $$J_x(F) = J_x(F_{k_1}) + \sum_{i=1}^{i_{max}} J_x\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \odot_x [J_x(F_{k_i} - F_{k_1})]\,.$$

Recall that $i_{max} \leq C$, by (57). Therefore (77) and (78) lead to

(79) $\quad J_x(F) \in J_x(F_{k_1}) + C A_2^m A_3(\mathcal{A}_0^-) M_0 \cdot \sigma(x,0).$

Next we apply (73) for $i = 1$, together with (79) and Property 1 from Section 13. We conclude that

(80) $\quad J_x(F) \in \Gamma(x, 0, C A_2^m A_3(\mathcal{A}_0^-) M_0).$

Assume, as we may, that

(81) $\quad A_0 > C$ where C is the constant from (80).

According to (3) from Section 17 we know that $A_3(\mathcal{A}_0) = A_0 A_2^m A_3(\mathcal{A}_0^-)$, and hence (80), (81) imply

$$J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A}_0) M_0).$$

Therefore (MLC3′) is proven.

It remains to prove (MLC4′). Next, suppose that $x = x_0$. By restricting (53) to $(m-1)$-jets, we get

(82) $$J_x(F) = J_{x_0}(F) = \sum_{i=1}^{i_{max}} J_{x_0}\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \odot_{x_0} J_{x_0}(F_{k_i}).$$

For each $i = 1, ..., i_{max}$ we have that $x = x_0 \in (1 + c_G) Q_{k_i}$, and hence, according to (FK3), we know that $J_{x_0}(F_{k_i}) = P_0$. Thus (82) entails that

$$J_{x_0}(F) = \sum_{i=1}^{i_{max}} J_{x_0}\left(\theta_{Q_{k_i}}^{\mathcal{A}_0^-}\right) \odot_{x_0} P_0 = P_0,$$

since the θ's are partition of unity and their sum is one. Therefore (MLC4′) is established. This completes the proof of the lemma. ■

According to Lemma 4, each $x \in (1+c_G)Q_0$ satisfies (MLC1′),...,(MLC4′). By comparing (MLC1′),...,(MLC4′) with (MLC1),...,(MLC4) from Section 33.2, we see that the conclusions of the *Main Lemma for* $\mathcal{A}_0$ hold true. Thus, we have proven the conclusions of the *Main Lemma for* $\mathcal{A}_0$ under the assumptions (ML1),...,(ML4). Consequently, the *Main Lemma for* $\mathcal{A}_0$ is proven also in the non-trivial case. This finishes the proof of the *Main Lemma for* $\mathcal{A}_0$ in all cases.

The *Main Lemma for* $\mathcal{A}_0$ is therefore established, for all $\mathcal{A}_0 \subseteq \mathcal{M}$.

Remark. Note that we actually used only Properties 1, 2, 3, 4 from Section 13 in order to prove the *Main Lemma.* Property 0 will be used only when applying the *Main Lemma* in the next section.

34. Applications of the Main Lemma

In this section, we will apply the *Main Lemma for* $\emptyset$, whose proof was completed in the previous section. Recall from Section 29 the formulation of the *Main Lemma for* $\emptyset$. Recall from Section 14 and Section 17, that $\ell_* = \ell(\emptyset) + 1, A_2, A_3(\emptyset)$ are constants depending only on m and n. According to Lemma 5 of Section 21, the tiling $CZ(\emptyset)$ consists of all dyadic cubes of sidelength A_2^{-1}. In the particular case where $x_0 \in E \cap Q_0^*$, the *Main Lemma for* $\emptyset$ from Section 29 reads as follows:

Lemma 1. *Suppose that* $Q_0 \subset \mathbb{R}^n$ *is a dyadic cube of sidelength* A_2^{-1}, $x_0 \in E \cap Q_0^*$ *and* $M_0 > 0$. *Let* $P_0 \in \mathcal{P}$ *be such that*

$$P_0 \in \Gamma(x_0, \ell_* - 1, M_0).$$

Then, there exists

$$F \in C^m((1+c_G)Q_0)$$

with the following properties:

$$(1) \qquad |\partial^\beta(F - P_0)(x)| \le CM_0 \qquad \textit{for all} \quad |\beta| \le m, x \in (1+c_G)Q_0.$$

$$(2) \qquad J_x(F) \in \Gamma(x, 0, CM_0) \qquad \textit{for all} \quad x \in E \cap (1+c_G)Q_0.$$

$$(3) \qquad J_x^+(F) = f_x(\emptyset, Q_0, x_0, P_0) \qquad \textit{for all} \quad x \in (1+c_G)Q_0.$$

$$(4) \qquad \textit{If} \;\; x_0 \in (1+c_G)Q_0, \quad \textit{then also} \;\; J_{x_0}(F) = P_0.$$

Here, $C > 0$ *is a constant depending only on* m *and* n.

The polynomial $f_x(\emptyset, Q_0, x_0, P_0)$ in (3) was computed by the *Main Algorithm* in Section 29. As a first application of Lemma 1, we will prove the following theorem.

Theorem 3. *Suppose we are given the following data:*

- *A finite set* $E \subset \mathbb{R}^n$ *of size* N.
- *For each* $x \in E$*, two real numbers* $f(x) \in \mathbb{R}$ *and* $\sigma(x) \geq 0$.
- *A point* $x_0 \in E$ *and a polynomial* $P_0 \in \mathcal{P}$.

Then, there exists $F \in C^m(\mathbb{R}^n)$ *with* $J_{x_0}(F) = P_0$ *such that the following hold:*

(I) *Suppose* $M > 0$ *satisfies that* $P_0 \in \Gamma(x_0, \ell_*, M)$*. Then,*

$$\| F \|_{C^m(\mathbb{R}^n)} \leq CM \quad \text{and} \quad |F(x) - f(x)| \leq CM\sigma(x) \quad \text{for all } x \in E.$$

(II) *The algorithm to be described below, receives the given data, performs one-time work, and then responds to queries.*

A query consists of a point $x \in \mathbb{R}^n$*, and the response to the query is the jet* $J^+_x(F)$.

The one-time work takes $CN \log N$ *operations, and* CN *storage. The time to answer a query is* $C \log N$.

Here, C *is a constant depending only on* m *and* n.

We start with describing the relevant algorithm.

The algorithm promised in Theorem 3

As in (2) from Section 10, we will consider the blobs,

(5) $\Gamma(x, 0, M) = \{P \in \mathcal{P} : |P(x) - f(x)| \leq M\sigma(x), |\partial^\beta P(x)| \leq M \text{ for } |\beta| \leq m-1\}$.

In the one-time work, as is described in Section 10, we will construct from $(\Gamma(x, 0, M))_{x \in E}$ the ALPs that give rise to blobs that are C-equivalent to $\Gamma(x, \ell, M)$ for $x \in E, 0 \leq \ell \leq \ell_*$. We will also perform the one-time work that is described in Section 9, in Section 23 and in Sections 24,...,26.

Next, we subdivide $\mathbb{R}^n$ into dyadic cubes of sidelength A_2^{-1}. Let Ω_0 be the set all dyadic cubes Q of sidelength A_2^{-1}, such that $E \cap Q^* \neq \emptyset$. For each $x \in E$, there are at most 5^n cubes $Q \in \Omega_0$ such that $x \in Q^*$. These cubes may be calculated in a straightforward manner, using C operations (for a fixed $x \in E$). By inspecting all points $x \in E$, we may find all the cubes of Ω_0 using CN computer operations. Note that $\#(\Omega_0) < CN$.

Let us fix a linear order $\prec$ on the cubes of Ω_0, say, lexicographic order on the centers of the cubes (lexicographic order with respect to the standard coordinates). Within $CN \log N$ computer operations, we may sort Ω_0 according to the order $\prec$. To summarize,

(6) At the one-time work, we compute and store the ordered list Ω_0, consuming $CN \log N$ computer operations and CN storage.

For each $Q \in \Omega_0$, we compute a representative $x_Q \in E \cap Q^*$, as follows.

(7) If $x_0 \in E \cap Q^*$ then $x_Q := x_0$, and otherwise $x_Q :=$ Find-Representative(Q).

The computation of those representatives requires $CN \log N$ operations, and may be performed during the process of computing the list Ω_0. (As a matter of fact, this task may be carried out using only CN operations, in the course of the computation of Ω_0. We will not use this fact.)

Next, with each $Q \in \Omega_0$ we will associate a polynomial P_Q. Fix $Q \in \Omega_0$. If $x_Q = x_0$ we will simply set $P_Q := P_0$. Otherwise, we compute a polynomial P_Q such that

(8) P_Q is a C-original vector of the blob $\Gamma(x_Q, \ell_* - 1)$,

where C-original vectors are defined in Section 2. The computation of P_Q as in (8) is done using Algorithm ALP3 from Section 5. For a fixed $Q \in \Omega_0$ the computation of P_Q requires C operations, once we have precomputed the Γ's. During the one-time work, we also compute and store x_Q, P_Q $(Q \in \Omega_0)$. The total time required for the computation of the points x_Q and the polynomials P_Q does not exceed $CN \log N$, and the amount of storage needed is no more than CN.

This completes the description of the one-time work of our algorithm. The resources being spent for the one-time work are bounded by $CN \log N$ computer operations and CN storage, for C depending only on m and n.

We move to the implementation of the query-algorithm. Thus, suppose we are given a point $x \in \mathbb{R}^n$. We define $\Omega_0(x) = \{Q \in \Omega_0 : x \in (1 + c_G)Q\}$. Note that $\#(\Omega_0(x)) < C$.

It is straightforward to compute, say, the centers of all dyadic cubes Q of sidelength A_2^{-1} such that $x \in (1 + c_G)Q$. This computation requires C computer operations, and produces the centers of all the cubes in $\Omega_0(x)$. We still need to locate these cubes in Ω_0 (i.e., to identify their indices in the list Ω_0); this is done using C binary searches in the ordered list Ω_0, each consuming $C' \log N$ work. Therefore $\Omega_0(x)$ is recovered within $C \log N$ operations.

Once $\Omega_0(x)$ is obtained, the algorithm computes and returns the polynomial

$$\bar{P}_x := \sum_{Q \in \Omega_0(x)} J_x^+ \left(\theta_Q^{\emptyset}\right) \odot_x^+ f_x(\emptyset, Q, x_Q, P_Q). \tag{9}$$

Thanks to Algorithm PU2 from Section 28, we may compute all the jets $J_x^+ \left(\theta_Q^{\emptyset}\right)$ within $C \log N$ operations. (As a matter of fact, C operations suffice here; see the last paragraph in Section 28. We will not make use of this improvement here.)

The computation of the polynomials $f_x(\emptyset, Q, x_Q, P_Q)$ is described in the *Main Algorithm*, Section 29, and requires no more than $C \log N$ computer operations, given our one-time work.

This completes the description of the query-algorithm. The query-algorithm clearly terminates within $C \log N$ operations.

Next we will prove that the polynomials $\bar{P}_x$, as defined in (9), are the m-jets of a function F that satisfies (I) from Theorem 3.

Lemma 2. *Let* E, f, σ, x_0, P_0 *be as in Theorem* 3. *Then, there exists* $F \in C^m(\mathbb{R}^n)$ *for which the following holds: Suppose* $M > 0$ *satisfies that*

(10) $\quad P_0 \in \Gamma(x_0, \ell_*, M).$

Then,

(11) $\quad |F(x) - f(x)| \leq CM\sigma(x)$ *for all* $x \in E$,

(12) $\quad \| F \|_{C^m(\mathbb{R}^n)} \leq CM,$

(13) $\quad J_x^+(F) = \bar{P}_x$ *for all* $x \in \mathbb{R}^n$, *where* $\bar{P}_x$ *is defined in* (9)*; and*

(14) $\quad J_{x_0}(F) = P_0.$

Here, $C > 0$ *denotes a constant depending only on* m *and* n.

Proof. We begin with analyzing the definition (8) of the polynomials P_Q. Let $Q \in \Omega_0$. If $x_Q \neq x_0$, then according to the defining property of a "C-original vector" from Section 2, the polynomial P_Q satisfies the following:

(15) $\quad$ Let $M' > 0$ and assume that $\Gamma(x_Q, \ell_* - 1, M') \neq \emptyset$. Then $P_Q \in \Gamma(x_Q, \ell_* - 1, CM')$.

In the case where $x_Q = x_0$, we know from (10) that

(16) $\quad P_Q = P_0 \in \Gamma(x_Q, \ell_*, M).$

Fix a cube $Q \in \Omega_0$. According to Property 2 from Section 13,

(17) $\quad \Gamma(x_0, \ell_*, M) \subseteq \Gamma(x_Q, \ell_* - 1, CM) + CMB(x_0, x_Q).$

In particular, from (10) and (17) we see that

(18) $\quad \Gamma(x_Q, \ell_* - 1, CM) \neq \emptyset.$

Thus, if $x_Q \neq x_0$, then (15), (18) imply that

(19) $\quad P_Q \in \Gamma(x_Q, \ell_* - 1, CM) \subseteq C'MB(x_Q, 1),$

where the last inclusion follows from (5) and Property 4 of Section 13. (Note that in (19) we use a trivial property of the Γ's, the fact that $\Gamma(x, 0, M) \subseteq MB(x, 1)$, which is not included in Properties 0,...,4 from Section 13.)

If $x_Q = x_0$, then (19) follows from (16). Thus, (19) holds in all cases. Now (19), together with (3) of Section 12, implies that for all $x \in (1+c_G)Q$,

(20) $|\partial^\beta P_Q(x)| \leq CM$ for all $|\beta| \leq m$.

(Recall that $|x - x_Q| \leq C\delta_Q \leq C$ for all $x \in (1+c_G)Q$.) We will now invoke Lemma 1, for the cube Q, the point $x_Q \in E \cap Q^*$, the polynomial P_Q and $M_0 = CM$, based on (19). By the conclusion of the lemma, there exists $F_Q \in C^m((1+c_G)Q)$, with the following properties:

(21) $|\partial^\beta(F_Q - P_Q)(x)| \leq CM$ for all $|\beta| \leq m, x \in (1+c_G)Q$.

(22) $J_x(F_Q) \in \Gamma(x, 0, CM)$ for all $x \in E \cap (1+c_G)Q$,

(23) $J_x^+(F_Q) = f_x(\emptyset, Q, x_Q, P_Q)$ for all $x \in (1+c_G)Q$,

where $f_x(\emptyset, Q, x_Q, P_Q)$ is defined by the *Main Algorithm* from Section 29, and

(24) If $x_Q \in (1+c_G)Q$, then $J_{x_Q}(F_Q) = P_Q$.

The cube $Q \in \Omega_0$ is arbitrary, hence a function $F_Q \in C^m((1+c_G)Q)$ that satisfies (21),...,(24) exists for all $Q \in \Omega_0$. We define a function $F : \mathbb{R}^n \to \mathbb{R}$ by setting

(25) $$F(x) = \sum_{Q \in \Omega_0} \theta_Q^\emptyset(x) F_Q(x).$$

Since $\mathrm{Supp}(\theta_Q^\emptyset) \subseteq (1 + c_G/2)Q$ and $F_Q \in C^m((1+c_G)Q)$, then each summand in the right-hand side of (25) is a well-defined $C^m(\mathbb{R}^n)$-function. The sum in (25) is finite, since $\#(\Omega_0) < \infty$. Hence F is a $C^m(\mathbb{R}^n)$-function. For any $x \in \mathbb{R}^n$, we have that $x \in \mathrm{Supp}(\theta_Q^\emptyset)$ only for $Q \in \Omega_0(x)$. Therefore (25) implies that

(26) $$J_x^+(F) = \sum_{Q \in \Omega_0(x)} J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ J_x^+(F_Q).$$

For any $Q \in \Omega_0$ we have $\delta_Q = A_2^{-1}$, and by (10) of Section 28,

(27) $|\partial^\beta(\theta_Q^\emptyset)(x)| \leq C$ for all $|\beta| \leq m$ and $x \in \mathbb{R}^n$.

Our estimates (20),(21) and (27) imply that for any $Q \in \Omega_0$,

(28) $|\partial^\beta(\theta_Q^\emptyset F_Q)(x)| < CM$ for all $|\beta| \leq m$ and $x \in \mathbb{R}^n$,

as $\mathrm{Supp}(\theta_Q^\emptyset) \subseteq (1 + c_G/2)\,Q$. In view of the fact that $\#(\Omega_0(x)) < C$ for any $x \in \mathbb{R}^n$, we deduce from (26) and (28) that

(29) $|\partial^\beta F(x)| < CM$ for all $|\beta| \leq m, x \in \mathbb{R}^n$.

Thus, (12) is proven. Furthermore, from (23) and (26) we conclude that,

$$(30) \quad J^+_x(F) = \sum_{Q \in \Omega_0(x)} J^+_x\left(\theta^\emptyset_Q\right) \odot^+_x f_x(\emptyset, Q, x_Q, P_Q) \text{ for all } x \in \mathbb{R}^n.$$

By comparing (30) with (9), we arrive at (13). Next, we focus on proving (11). Fix $x \in E$. We need to show that

$$(31) \quad J_x(F) \in \Gamma(x, 0, CM).$$

Let us also fix a cube $Q \in \Omega_0(x)$. Then, since the θ's are a partition of unity,

$$(32) \quad J_x(F) = J_x(F_Q) + M \sum_{Q_\nu \in \Omega_0(x)} J_x\left(\theta^\emptyset_{Q_\nu} \cdot \frac{F_{Q_\nu} - F_Q}{M}\right).$$

According to (20), (21) we know that for any $Q_\nu \in \Omega_0(x)$,

$$(33) \quad J_x(F_{Q_\nu} - F_Q) \in CMB(x, 1),$$

since $x \in (1 + c_G)Q \cap (1 + c_G)Q_\nu$. By applying (22) twice, we conclude that for any $Q_\nu \in \Omega_0(x)$,

$$(34) \quad J_x(F_{Q_\nu} - F_Q) \in \Gamma(x, 0, CM) - \Gamma(x, 0, CM) \subseteq C'M\sigma(x, 0),$$

where the inclusion follows from Property 1 of Section 13. Next, we invoke the Whitney t-Convexity of $\sigma(x, 0)$, according to Property 3 from Section 13. The Whitney t-Convexity, based on (33), (34) and (27), entails that for any $Q_\nu \in \Omega_0(x)$,

$$(35) \quad J_x\left(\theta^\emptyset_{Q_\nu} \cdot \tfrac{F_{Q_\nu} - F_Q}{M}\right) = J_x\left(\theta^\emptyset_{Q_\nu}\right) \odot_x J_x\left(\tfrac{F_{Q_\nu} - F_Q}{M}\right) \in C\sigma(x, 0).$$

Recall that $\#(\Omega_0(x)) < C$. From (32), (35) and (22), we obtain that

$$(36) \quad J_x(F) \in \Gamma(x, 0, CM) + C'M\sigma(x, 0) \subseteq \Gamma(x, 0, \tilde{C}M),$$

where the last inclusion follows from Property 1 of Section 13. The inclusion (36) is precisely the desired estimate (31). Hence (11) is proven. It remains to prove (14). By the definition of x_Q, P_Q, for any cube $Q \in \Omega_0(x_0)$ we have $x_Q = x_0, P_Q = P_0$. According to (24) and (26), we have

$$J_{x_0}(F) = \sum_{Q_\nu \in \Omega_0(x_0)} J_{x_0}\left(\theta^\emptyset_{Q_\nu}\right) \odot_{x_0} P_0 = P_0,$$

since the θ's constitute a partition of unity. Thus (14) follows and the lemma is proven. ∎

Lemma 2 implies that the output of our query-algorithm, that is, the polynomials $\bar{P}_x$ defined in (9), are the m-jets of a function that satisfies (I) from Theorem 3. This completes the proof of Theorem 3. ■

Remark. In Theorem 3, we prescribe the $(m-1)$-jet of F at a given point x_0. We would like to mention without proof, that it is equally easy to prescribe the m-jet of F at the given point x_0. Denote by $\pi_{x_0} : \mathcal{P}^+ \to \mathcal{P}$ the linear map that satisfies $\partial^\beta(\pi_{x_0}P)(x_0) = \partial^\beta P(x_0)$ for all $|\beta| \le m-1$ and $P \in \mathcal{P}^+$. Suppose we alter the formulation of Theorem 3 as follows:

- Replace "$P_0 \in \mathcal{P}$" with "$P_0^+ \in \mathcal{P}^+$".
- Replace "$J_{x_0}(F) = P_0$" with "$J_{x_0}^+(F) = P_0^+$".
- Replace "$P_0 \in \Gamma(x_0, \ell_*, M)$" with "$P_0^+ \in MB^+(x_0, 1)$ and $\pi_{x_0}(P_0^+) \in \Gamma(x_0, \ell_*, M)$".

Then the modified theorem holds true. We invite the reader to fill in the proof.

We are now in a position to prove Theorem 2 from Section 1. Theorem 2 follows from the following theorem.

Theorem 4. *Suppose we are given the following data:*

- *A finite set $E \subset \mathbb{R}^n$.*
- *For each $x \in E$, two real numbers $f(x) \in \mathbb{R}$ and $\sigma(x) \ge 0$.*

Assume that $\#(E) = N$. Then, there exists $F \in C^m(\mathbb{R}^n)$ with the following properties:

(A) *If $\tilde{F} \in C^m(\mathbb{R}^n)$ and $M > 0$ satisfy*

(37) $\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \le M$ *and* $|\tilde{F}(x) - f(x)| \le M\sigma(x)$ *for* $x \in E$,

then

(38) $\| F \|_{C^m(\mathbb{R}^n)} \le CM$ *and* $|F(x) - f(x)| \le CM\sigma(x)$ *for* $x \in E$.

(B) *There is an algorithm, that receives the given data, performs one-time work, and then responds to queries.*

◇ *A query consists of a point $x \in \mathbb{R}^n$, and the response to the query is the jet $J_x^+(F)$.*

◇ *The one-time work takes $CN \log N$ operations, and CN storage. The work to answer a query is $C \log N$.*

Here, C is a constant depending only on m and n.

Proof. Let us pick $x_0 \in E$, and let $P_0 \in \mathcal{P}$ be such that

(39) $\quad P_0$ is a C-original vector of the blob $\Gamma(x_0, \ell_*)$.

We will compute the ALPs that give rise to blobs that are C-equivalent to the Γ's in the one-time work, using $CN \log N$ operations and CN storage. Having already constructed those ALPs, we may compute the polynomial P_0 using Algorithm ALP3 from Section 5, using no more than C operations.

We will now invoke Theorem 3, for E, f, σ and x_0, P_0. By the conclusion of that theorem, we obtain a certain function F. We will show that F satisfies (A) and (B). Note that (B) follows from (II) of Theorem 3. We still need to prove (A). Suppose that $\tilde{F} \in C^m(\mathbb{R}^n)$ and $M > 0$ are such that

(40) $\quad \| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq M$ and $|\tilde{F}(x) - f(x)| \leq M\sigma(x)$ for $x \in E$.

We will show that

(41) $\quad \| F \|_{C^m(\mathbb{R}^n)} \leq CM$ and $|F(x) - f(x)| \leq CM\sigma(x)$ for $x \in E$.

To that end, note that Property 0 from Section 13 and (40) imply that

(42) $\quad J_{x_0}(\tilde{F}) \in \Gamma(x_0, \ell_*, CM)$.

In particular $\Gamma(x_0, \ell_*, CM) \neq \emptyset$. By (39), and by the defining property of "C-original vectors" from Section 2,

(43) $\quad P_0 \in \Gamma(x_0, \ell_*, C'M)$.

From (43), and according to (I) of Theorem 3, we conclude (41). Thus, given (40) we deduce (41). This is exactly the content of (A). The proof is thus complete. ■

Remark. An alternative implementation for Theorem 4 moves work from the query algorithm into the one-time work. The idea is as follows. Let $F \in C^m(\mathbb{R}^n)$ be as in Theorem 4. For $x \in E$, let $P^x = J_x^+(F)$.

The algorithm given above for Theorem 4 allows us to compute (and store) all the jets P^x $(x \in E)$, with work $CN \log N$ and storage CN. We view this as part of the one-time work.

The proof of the classical Whitney extension theorem produces a function $\tilde{F} \in C^m(\mathbb{R}^n)$, with $J_x^+(\tilde{F}) = P^x$ for every $x \in E$, and with $\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq C\| F \|_{C^m(\mathbb{R}^n)}$. (Here, C depends only on m and n.) Thus, $\tilde{F}$ serves as well as F in Theorem 4. The methods of this paper allow us easily to answer queries as follows: Given a query point $\underline{x} \in \mathbb{R}^n$, we produce the jet $J_{\underline{x}}^+(\tilde{F})$. We omit the details.

For $x \in \mathbb{R}^n$ and $M > 0$ we define the blob $\Sigma(x) = (\Sigma(x, M))_{M>0}$ by setting

$$\Sigma(x, M) = \left\{ J_x(F) : \| F \|_{C^m(\mathbb{R}^n)} \leq M, \quad \text{and} \quad \forall x \in E, \ |F(x) - f(x)| \leq M\sigma(x) \right\}.$$

The set $\Sigma(x, M)$ is convex and increasing with M, hence $\Sigma(x)$ is a blob.

Lemma 3. *Let $x_0 \in E$. Then the blobs $\Sigma(x_0)$ and $\Gamma(x_0, \ell_*)$ are C-equivalent, for a constant C depending only on m and n.*

Proof. Fix $M > 0$, and let $P_0 \in \Sigma(x_0, M)$. According to Property 0 from Section 13, we have that

$$P_0 \in \Gamma(x_0, \ell_*, CM).$$

Since $P_0 \in \Sigma(x_0, M)$ is arbitrary, we conclude that $\Sigma(x_0, M) \subseteq \Gamma(x_0, \ell_*, CM)$. Next, suppose $P_0 \in \Gamma(x_0, \ell_*, M)$. We will apply Theorem 3, for E, f, σ and for x_0, P_0. According to (I) from Theorem 3, there exists a function $F \in C^m(\mathbb{R}^n)$ with $P_0 = J_{x_0}(F)$ such that

$$\| F \|_{C^m(\mathbb{R}^n)} \leq CM, \quad \text{and} \quad \forall x \in E, \ |F(x) - f(x)| \leq CM\sigma(x).$$

Therefore, $P_0 \in \Sigma(x_0, CM)$. Hence $\Gamma(x_0, \ell_*, M) \subseteq \Sigma(x_0, CM)$. This completes the proof. ■

Lemma 4. *Let $x \in \mathbb{R}^n$ be such that $x \notin E$, and let $x_0 \in E$ be such that*

$$|x - x_0| \leq 2 \min_{y \in E} |x - y| = 2\mathsf{dist}(x, E).$$

Then, $\Sigma(x)$ is C-equivalent to the blob

$$\left[\Gamma(x_0, \ell_*) + \mathcal{B}(x, |x - x_0|) \right] \cap \mathcal{B}(x, 1).$$

Here, C is a constant depending only on m and n.

Proof. Fix $M > 0$, and let $P \in \Sigma(x, M)$. By the definition of $\Sigma(x, M)$, there exists a function $F \in C^m(\mathbb{R}^n)$ such that

(44) $J_x(F) = P$,

(45) $\| F \|_{C^m(\mathbb{R}^n)} \leq M$, and

(46) $|F(y) - f(y)| \leq M\sigma(y)$ for all $y \in E$.

According to (44) and (45),

(47) $P \in MB(x, 1)$.

Furthermore, by (45), (46) and Property 0 from Section 13,

(48) $J_{x_0}(F) \in \Gamma(x_0, \ell_*, CM)$.

The function F satisfies (45). By Taylor's theorem, $J_w(F) - J_z(F) \in CMB(z, w)$ for any $z, w \in \mathbb{R}^n$ (see also (3) from Section 1). From (44) we thus conclude that,

(49) $\quad P - J_{x_0}(F) = J_x(F) - J_{x_0}(F) \in CMB(x, x_0).$

According to (48) and (49),

(50) $\quad P \in \Gamma(x_0, \ell_*, CM) + CMB(x, x_0).$

Since P is an arbitrary polynomial in $\Sigma(x, M)$, then (50) and (47) tell us that for any $M > 0$,

(51) $\quad \Sigma(x, M) \subseteq \left[\Gamma(x_0, \ell_*, CM) + CMB(x, x_0)\right] \cap CMB(x, 1).$

This proves half of the conclusion of the lemma. We now focus on proving the second half. Let $M > 0$ and let

(52) $\quad P \in \left[\Gamma(x_0, \ell_*, M) + MB(x, x_0)\right] \cap MB(x, 1).$

Then there exists $P_0 \in \Gamma(x_0, \ell_*, M)$ such that

(53) $\quad P - P_0 \in MB(x, x_0) \subseteq CMB(x, \mathrm{dist}(x, E)),$

since, by our assumptions, $|x - x_0| \leq 2\mathrm{dist}(x, E)$. Since $P_0 \in \Gamma(x_0, \ell_*, M)$, then according to Lemma 3, we have that $P_0 \in \Sigma(x_0, CM)$. By the definition of $\Sigma(x_0, CM)$, there exists $F' \in C^m(\mathbb{R})$ with $J_{x_0}(F') = P_0$ such that

(54) $\quad \| F' \|_{C^m(\mathbb{R}^n)} \leq C'M, \quad \text{and} \quad \forall x' \in E,\ |F'(x') - f(x')| \leq CM\sigma(x').$

Next, fix $y \in E$. Then, by (54), (53) and the definition of F',

(55) $$\begin{aligned} J_y(F') - P &= (J_y(F') - J_{x_0}(F')) + (P_0 - P) \\ &\in CMB(x_0, y) + CMB(x, \mathrm{dist}(x, E)). \end{aligned}$$

However, according to (3) of Section 12,

(56) $\quad B(x_0, y) = B(x_0, |y - x_0|) \subseteq CB(x, |y - x_0| + |x - x_0|).$

Furthermore, since $|x - x_0| \leq 2\mathrm{dist}(x, E) \leq 2|x - y|$ then,

(57) $\quad |y - x_0| + |x - x_0| \leq (|y - x| + |x - x_0|) + |x - x_0| \leq 5|y - x|.$

From (56) and (57) we deduce that $B(x_0, y) \subseteq C'B(x, 5|y - x|) \subseteq CB(x, y)$. By combining the last inclusion with (55), we conclude that for any $y \in E$,

(58) $\quad J_y(F') - P \in C'MB(x, y) + CMB(x, \mathrm{dist}(x, E)) \subseteq \tilde{C}MB(x, y),$

as $\mathrm{dist}(x, E) \leq |x - y|$. Denote $\tilde{E} = E \cup \{x\}$. To any $y \in \tilde{E}$ we associate a polynomial $\tilde{P}_y \in \mathcal{P}$ as follows: $\tilde{P}_y = J_y(F')$ if $y \in E$, and $\tilde{P}_y = P$ if $y = x$. Note that $x \notin E$, and hence the $\tilde{P}_y$'s are well-defined.

We would like to apply Whitney's theorem, as described in Section 1 (see also [35] or [32, Section VI]). According to (54), we have that

$$(59)\quad \tilde{P}_y - \tilde{P}_z = J_y(F') - J_z(F') \in CMB(y,z) \text{ for any } y, z \in E.$$

We use (54) for the case $y \in E$, and we use (52) for the case $y = x$, to obtain that

$$(60)\quad \tilde{P}_y \in CMB(y,1) \text{ for all } y \in \tilde{E}.$$

Based on (58), (59) and (60) we may invoke Whitney's theorem, for the set $\tilde{E}$ and the polynomials $\{\tilde{P}_y\}_{y \in \tilde{E}}$. By the conclusion of Whitney's theorem, there exists $\tilde{F} : \mathbb{R}^n \to \mathbb{R}$ with $\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq \tilde{C}M$ such that

$$(61)\quad J_y(\tilde{F}) = \tilde{P}_y = J_y(F') \text{ for all } y \in E, \text{ and also } J_x(\tilde{F}) = \tilde{P}_x = P.$$

According to (61) and (54), the function $\tilde{F}$ witnesses that $P \in \Sigma(x, \hat{C}M)$. Since P is an arbitrary polynomial in $[\Gamma(x_0, \ell_*, M) + MB(x, x_0)] \cap MB(x, 1)$, we conclude that for any $M > 0$,

$$(62)\quad [\Gamma(x_0, \ell_*, M) + MB(x, x_0)] \cap MB(x, 1) \subseteq \Sigma(x, CM).$$

The lemma follows from (62) and (51). ■

Theorem 5. *Suppose we are given the following data:*

- *A finite set $E \subset \mathbb{R}^n$.*
- *For each $x \in E$, two real numbers $f(x) \in \mathbb{R}$ and $\sigma(x) \geq 0$.*

Assume that $\#(E) = N$. Then there is an algorithm, that gets the given data, performs one-time work, and then responds to queries. A query consists of a point $x \in \mathbb{R}^n$, and the response to the query is an ALP *$\mathcal{A}$ (of length at most* dim *$\mathcal{P}$) such that $\mathcal{K}(\mathcal{A})$ is C-equivalent to the blob $\Sigma(x) = (\Sigma(x, M))_{M>0}$ defined by,*

$$\Sigma(x, M) = \left\{ J_x(F) : \| F \|_{C^m(\mathbb{R}^n)} \leq M, \ \textit{and} \ \forall y \in E, \ |F(y) - f(y)| \leq M\sigma(y) \right\}.$$

The one-time work requires $CN \log N$ operations and CN storage. The time to answer a query is $C \log N$. Here, C is a constant depending only on m and n.

Proof. Let us describe the relevant algorithm. First, we perform all the one-time work of the algorithm from Theorem 3, and also all the one-time work related to Theorem BBD1 from Section 23. This one-time work requires $CN \log N$ operations, and CN storage, as stated in Theorem 3 and in Theorem BBD1.

We will now present the query-algorithm. Suppose we are given a point $x \in \mathbb{R}^n$. We need to produce an ALP $\mathcal{A}$, such that $\mathcal{K}(\mathcal{A})$ is C-equivalent to $\Sigma(x)$. To that end, note that according to Theorem BBD1 from Section 23, we may compute within $C \log N$ operations, a point $x_0 \in E$ such that

$$(63) \quad |x - x_0| \leq 2\mathrm{dist}(x, E).$$

Note also that the ALPs that give rise to blobs that are C-equivalent to the Γ's are computed in the one-time work of our algorithm. According to (63), we may easily detect whether $x \in E$ or $x \notin E$. In case where $x \in E$, our query-algorithm returns an ALP $\mathcal{A}$ of length $\dim \mathcal{P}$ such that $\mathcal{K}(\mathcal{A})$ is C-equivalent to $\Gamma(x, \ell_*)$. Such an ALP was already computed in the one-time work (see Section 10), and by Lemma 3 we have that $\mathcal{K}(\mathcal{A})$ is C-equivalent to $\Sigma(x)$.

In the case where $x \notin E$, we compute, within C computer operations, an ALP $\mathcal{A}$ such that the blob $\mathcal{K}(\mathcal{A})$ is C-equivalent to

$$(64) \quad [\Gamma(x_0, \ell_*) + \mathcal{B}(x, x_0)] \cap \mathcal{B}(x, 1).$$

Indeed, we simply need to apply Algorithm ALP6 and Algorithm ALP7 from Section 5. From the explanation in Section 5 we know that the ALP returned by Algorithm ALP6 has length at most $\dim \mathcal{P}$. Lemma 4 tells us that the blob in (64) is C-equivalent to $\Sigma(x)$. Our query algorithm clearly uses at most $C \log N$ computer operations. The proof is complete. ■

In the following two sections, we will discuss several variants of the theorems and algorithms presented in this section. These variants will be formulated precisely, in Theorem 6, Theorem 7 and Theorem 8, but we will not supply full details pertaining their proofs. Instead, we will indicate the necessary modifications of the above arguments, that lead to the proofs of Theorems 6, 7, 8. Filling in the missing details is quite routine, given the proofs of Theorem 3, Theorem 4 and Theorem 5 on which we have elaborated throughout this manuscript.

35. Linear Dependence on Input

In this section, we strengthen Theorem 4 from Section 34, by producing an extension function F that depends linearly on the given f. We write c, C, C', etc., to denote constants depending only on m and n.

Let us recall from Section 16 the concept of a linear map of depth k. Suppose $L : \mathbb{R}^N \longrightarrow \mathcal{P}^+$ is a depth k linear map, given by a $D^+ \times N$ matrix L'. (Here, $D^+ = \dim \mathcal{P}^+$.) Then at most Ck of the entries of L' are non-zero. In order to specify the depth k linear map L, it is sufficient to indicate which entries of L' are non-zero, and then specify the values of those non-zero entries. By using this representation, we may hold a depth k linear map using $C \cdot (k+1)$ storage (see Section 16). We call this representation the "compact representation" of L.

Theorem 6. *Suppose we are given the following data:*

- *A finite set* $E \subset \mathbb{R}^n$*.*
- *A number* $\sigma(x) \geq 0$ *for each* $x \in E$*.*

Assume that $\#(E) = N$*.*

Then, there exists a collection of linear maps $\{L_x : x \in \mathbb{R}^n\}$*, where* $L_x : \mathbb{R}^{\#(E)} \to \mathcal{P}^+$ *for each* $x \in \mathbb{R}^n$*, such that the following hold.*

(A) *For each* $x \in \mathbb{R}^n$*, the linear map* $L_x : \mathbb{R}^{\#(E)} \to \mathcal{P}^+$ *is of depth* C *(and thus depends only on at most* C' *among the* N *coordinates of its input).*

(B) *Suppose* $f := (f(y))_{y \in E}$*, where* $f(y) \in \mathbb{R}$ *for all* $y \in E$ *(i.e.,* $f \in \mathbb{R}^{\#(E)}$*). Then there exists a function* $F_f \in C^m(\mathbb{R}^n)$ *such that:*

(B1) $J_x^+(F_f) = L_x[f]$ *for all* $x \in \mathbb{R}^n$*.*

(B2) *If* $\tilde{F} \in C^m(\mathbb{R}^n)$ *and* $M > 0$ *satisfy*

$$\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq M \quad \textit{and} \quad |\tilde{F}(x) - f(x)| \leq M\sigma(x) \quad \textit{for } x \in E,$$

then

$$\| F_f \|_{C^m(\mathbb{R}^n)} \leq CM \quad \textit{and} \quad |F_f(x) - f(x)| \leq CM\sigma(x) \quad \textit{for } x \in E.$$

(C) *There is an algorithm, that takes the given data, performs one-time work, and then responds to queries.*

A query consists of a point $x \in \mathbb{R}^n$*, and the response to the query is the depth-*C *linear map* L_x*, given in its compact representation.*

The one-time work takes $CN \log N$ *operations, and* CN *storage. The time to answer a query is* $C \log N$*.*

Here, C *is a constant depending only on* m *and* n*.*

The proof of this theorem is identical to the proof given in the previous sections, with a few obvious modifications. The main modification, is that rather than computing $\Gamma(x, \ell, M)$ and $\sigma(x, \ell, M)$ using ALPs, we bring in the PALPs described in Section 6, Section 11 and Section 16. We supply details.

We define $\bar{N} = N$. We suppose that the index of an input point, an integer between 1 and N, may be stored in a single memory word. Thus assumption (2) from Section 6 is verified. We formally think of $(f(x))_{x \in E}$ as depending linearly on $\xi \in \mathbb{R}^{\sharp(E)} = \mathbb{R}^{\bar{N}}$, and we write $f = f_\xi$ to denote this formal linear dependence. For a fixed $x \in E$, the linear projection $\xi \mapsto f_\xi(x)$ from $\mathbb{R}^{\bar{N}}$ to $\mathbb{R}$ is (trivially) of depth 1. Therefore (1) from Section 11 holds, with $k = 1$. Recall from Section 11 that we are able to compute PALPs $\underline{\mathcal{A}}(x, \ell)$ with the following properties.

(1) For each $x \in E, 0 \le \ell \le \ell_*$, the PALP $\underline{\mathcal{A}}(x, \ell)$ and the ALP $\mathcal{A}(x, \ell)$ constructed from E, σ, f_ξ in Section 10 agree at ξ, for every $\xi \in \mathbb{R}^{\bar{N}}$.

(2) For each $x \in E, 0 \le \ell \le \ell_*$ the PALP $\underline{\mathcal{A}}(x, \ell)$ is of depth at most C'.

(3) The computation of the PALPs $\underline{\mathcal{A}}(x, \ell)$, for $x \in E, 0 \le \ell \le \ell_*$, requires no more than $CN \log N$ operations and CN storage.

Next, based on the construction of the PALPs $\underline{\mathcal{A}}(x, \ell)$, we were able to describe in Section 16 the procedure, Find-Parametrized-Neighbor$(\vec{P}_0, \mathcal{A}, x)$, defined for $\mathcal{A} \subseteq \mathcal{M}, x \in E$ and a depth-k parametrized polynomial $\vec{P}_0$. (Recall from Section 16 the concept of a depth-k parametrized polynomial.) The output of Find-Parametrized-Neighbor is a depth-Ck parametrized polynomial $\vec{P}$ with the following property:

(4) Fix $\xi \in \mathbb{R}^{\bar{N}}$. Set $P_0 = \vec{P}_0(\xi), P = \vec{P}(\xi)$ and $f = f_\xi$. Then P is the polynomial returned by Find-Neighbor$(P_0, \mathcal{A}, x)$ with initial data E, σ, f.

Assuming we have already computed the PALPs $\underline{\mathcal{A}}(x, \ell)$, the procedure Find-Parametrized-Neighbor terminates within C' computer operations.

Next, we describe the (trivial) modifications needed, to adapt the *Main Algorithm* to the new situation. The first change, is that all the polynomials in the *Main Algorithm* (except for the jets of the θ's), are now being replaced with parametrized polynomials. In particular, $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ is replaced by $\vec{f}_x(\mathcal{A}_0, Q_0, x_0, \vec{P}_0)$ where $\vec{P}_0$ is a parametrized polynomial. The second change is that `Line 11` is replaced by

```
Line 11′:        Define $\vec{P}_k :=$ Find-Parametrized-Neighbor$(\vec{P}_0, \mathcal{A}_0, x_k)$.
```

Suppose that $\vec{P}_0$ is a parametrized polynomial of depth C, for some constant C depending only on m and n, and denote $\vec{P} = \vec{f}_x(\mathcal{A}_0, Q_0, x_0, \vec{P}_0)$. By an easy induction on $\mathcal{A}_0 \subseteq \mathcal{M}$ we obtain that for any $Q_0 \in CZ(\mathcal{A}_0), x_0 \in E \cap Q_0^{**}, x \in (1 + c_G)Q_0$, the following hold:

(5) $\vec{P}$ is a parametrized polynomial of depth C'.

(6) Fix $\xi \in \mathbb{R}^{\bar{N}}$ and set $P_0 = \vec{P}_0(\xi), P = \vec{P}(\xi)$ and $f = f_\xi$. Then P is the polynomial returned by $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ with initial data E, σ, f.

(7) The computation of $\vec{f}_x(\mathcal{A}_0, Q_0, x_0, \vec{P}_0)$ requires $C \log N$ computer operations, given one-time work of at most $CN \log N$ operations and CN storage.

We may now move to the proof of Theorem 4 in Section 34. Again, only obvious modifications are needed. We just need to replace (8), (9) and (39) from Section 34 with

(8′) $\vec{P}_Q$ is a parametrized C-original vector for the PALP $\underline{\mathcal{A}}(x_Q, \ell_*)$.

(9′) $$\vec{P}_x := \sum_{Q_\nu \in \Omega_0(x)} J_x^+\left(\theta_{Q_\nu}^{\emptyset}\right) \odot_x^+ \vec{f}_x(\emptyset, Q_\nu, x_{Q_\nu}, \vec{P}_{Q_\nu})$$

(39′) $\vec{P}_0$ is a parametrized C-original vector for the PALP $\underline{\mathcal{A}}(x_0, \ell_*)$.

We compute $\vec{P}_0$ in (39′) and $\vec{P}_Q$ in (8′) with the help of Algorithm PALP3 from Section 6. Since the PALPs $\underline{\mathcal{A}}(x, \ell_*)$ $(x \in E)$ are of depth C, it follows from the defining properties of Algorithm PALP3 that $\vec{P}_0$ and $\vec{P}_Q$ are parametrized polynomials of depth C'. By using (5), (6) and (7), it is straightforward to obtain the following result: For any $x \in \mathbb{R}^n$,

(8) The polynomial $\vec{P}_x$ is a parametrized polynomial of depth C.

(9) The polynomial $\bar{P}_x$ computed in (9) of Section 34 with initial data $E, \sigma, f_\xi, x_0, \vec{P}_0(\xi)$ equals the polynomial $\vec{P}_x(\xi)$ from (9′) with initial data $E, \sigma, x_0, \vec{P}_0$.

(10) The computation of $\vec{P}_x$ requires $C \log N$ computer operations, given one-time work of at most $CN \log N$ operations and CN storage.

It is now easy to deduce the conclusions of Theorem 6. Indeed, conclusion (A) follows from (8), conclusion (B) follows from (9) and Theorem 4 from Section 34, and (C) follows from (10). The proof of Theorem 6 is complete.

36. Different Types of Input

So far in this manuscript, we were mainly concerned with an efficient computation of a function $F : \mathbb{R}^n \to \mathbb{R}$, having a nearly minimal C^m-norm under certain restrictions on the values that the function F may attain on a given set $E \subset \mathbb{R}^n$.

In this section we will consider more general types of constraints on the desired function F. Rather than restricting the values that F may attain on the set E, we will impose conditions on the full jets $J_x(F)$ $(x \in E)$. In particular, we will discuss algorithms for efficiently computing a function $F : \mathbb{R}^n \to \mathbb{R}$, having prescribed jets of various orders at the points of E, such that $\| F \|_{C^m(\mathbb{R}^n)}$ has the smallest possible order of magnitude.

Recall the definition of Whitney t-convex sets, from Section 12.

Theorem 7. *Suppose we are given the following data:*

- *A finite set* $E \subset \mathbb{R}^n$.
- *For each* $x \in E$, *an* $(m-1)$*-jet* $f(x) \in \mathcal{P}$, *and a centrally-symmetric convex set* $\sigma(x) \subset \mathcal{P}$, *defined by at most, say,* $2 \dim \mathcal{P}$ *linear inequalities.*

Assume that $\#(E) = N$, *and that for each* $x \in E$, *the set* $\sigma(x)$ *is Whitney* t*-convex at* x, *with Whitney constant* W_0.

Then, there exists $F \in C^m(\mathbb{R}^n)$ *with the following properties*:

(I) *If* $\tilde{F} \in C^m(\mathbb{R}^n)$ *and* $M > 0$ *satisfy*

$$\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq M \quad \textit{and} \quad J_x(\tilde{F}) \in f(x) + M\sigma(x) \quad \textit{for} \quad x \in E,$$

then

$$\| F \|_{C^m(\mathbb{R}^n)} \leq CM \quad \textit{and} \quad J_x(F) \in f(x) + CM\sigma(x) \quad \textit{for} \quad x \in E.$$

Here C *is a constant depending only on* m, n *and* W_0.

(II) *There is an algorithm, that receives the given data, performs one-time work, and then responds to queries.*

A query consists of a point $x \in \mathbb{R}^n$, *and the response to the query is the jet* $J_x^+(F)$.

The one-time work takes $C'N \log N$ *operations, and* $C'N$ *storage. The work to answer a query is* $C' \log N$.

Here C' *is a constant depending only on* m *and* n.

Note that Theorem 4 is a particular case of Theorem 7, in which all the sets $\sigma(x) \subset \mathcal{P}$ take the form

(1) $\quad \sigma(x) = \{P \in \mathcal{P} \,:\, |P(x)| \leq \sigma'(x)\}$

for all $x \in E$, where $\sigma' : E \to [0, \infty)$ is some function. The centrally-symmetric convex set $\sigma(x)$ in (1) is Whitney t-convex at x, with Whitney constant 1. An additional interesting example of a Whitney t-convex set, is

(2) $\quad \sigma(x) = \left\{P \in \mathcal{P} \,:\, \partial^\beta P(x) = 0 \text{ for all } |\beta| \leq \sigma'(x)\right\}$

for all $x \in E$, where $\sigma' : E \to \{0, ..., m-1\}$. Note that $\sigma(x)$ as in (2) is Whitney t-convex at x with Whitney constant 1.

The proof of Theorem 7 is almost identical to the proof of Theorem 4. Next we will describe the differences between the two arguments. Thus, let E, f, σ be as in Theorem 7. From now on, in this section $C, C', \tilde{C}$, etc. denote constants depending only on m, n and W_0.

Recall the definition of the Γ's and the σ's from Section 10. In order to prove Theorem 7, we need to replace (2) of Section 10 with

$$(2') \quad \Gamma(x,0,M) = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M \text{ for } |\alpha| \leq m-1 , \\ \text{and } P \in f(x) + M\sigma(x)\}$$

for all $x \in E$. Having replaced (2) from Section 10 with (2′), we inductively define the sets $\Gamma(x,\ell,M)$ and $\sigma(x,\ell)$ for all $x \in E, M > 0$ and $\ell \geq 0$ exactly as in Section 10. Since $\sigma(x)$ is given by at most $2 \dim \mathcal{P}$ linear inequalities, then the blob $\Gamma(x,0)$ defined in (2′) is already given by an obvious ALP of length not exceeding $3 \dim \mathcal{P}$. Therefore we may carry out the recursive computation of the ALPs $\mathcal{A}(x,\ell)$ such that $\mathcal{K}(\mathcal{A}(x,\ell))$ is C-equivalent to $\Gamma(x,\ell)$, exactly as described in Section 10. The resources needed for the computation are still $CN \log N$ time and CN storage (with C depending only on m and n).

The new blobs $\Gamma(x,\ell)$ and convex sets $\sigma(x,\ell)$, that were constructed in the preceding paragraph, are the basic blobs that are relevant to the proof of Theorem 7, and they will replace the basic blobs defined in Section 10. Let us discuss the properties of the new basic blobs, in comparison to Section 13. Property 0 from Section 13 needs to be replaced with the following:

Property 0′:

(a) Let $F \in C^m(\mathbb{R}^n)$ and $M > 0$ be given. Assume that

$$\| F \|_{C^m(\mathbb{R}^n)} \leq M \text{ and } J_x(F) \in f(x) + M\sigma(x) \text{ for all } x \in E .$$

Then $J_x(F) \in \Gamma(x,\ell,C_\ell M)$ for all $x \in E$, $\ell \geq 0$, where C_ℓ depends solely on ℓ, m, n and W_0.

(b) Let $F \in C^m(\mathbb{R}^n)$ be such that

$$\| F \|_{C^m(\mathbb{R}^n)} \leq 1 \text{ and } J_x(F) \in \sigma(x) \text{ for all } x \in E .$$

Then $J_x(F) \in C_\ell \sigma(x,\ell)$ for all $x \in E$, $\ell \geq 0$, where C_ℓ depends solely on ℓ, m, n and W_0.

We claim that Property 0′, as well as Properties 1,...,4 from Section 13, hold also with our new definition of the Γ's and σ's, when the constants C, C', etc. are allowed now to depend also on W_0. Indeed, an inspection of the definition (2′) shows that Property 0′ from the present section, and Properties 1,...,4 from Section 13, all hold for $\ell = 0$, with constants depending only on m, n and W_0. As in Section 13, the proof for a general ℓ follows by induction. There are only the most trivial differences between the inductive proof in Section 13 and the argument needed here. We omit the straightforward details. We have thus explained how to establish Property 0′, as well as Properties 1,...,4 from Section 13, in the context of the new Γ's and σ's.

Except for the slightly different construction of the Γ's and σ's, in order to prove Theorem 7 we use the same algorithms and the same arguments as those used in the proof of Theorem 4, with the main difference being that the constants depend now also on W_0, in addition to m and n.

Recall the first paragraph from Section 31. According to this paragraph, Properties 0,...,4 from Section 13 are the only information we use regarding the Γ's and the σ's. (The only exception is (19) of Section 34, which trivially holds also here.) In order to adapt the arguments and algorithms that appear throughout the manuscript to suit the proof of Theorem 7, we simply note that Properties 1,...,4 from Section 13 may also be used in the new context, with the new Γ's and σ's. Indeed, this is the content of the previous paragraphs (with C, C' being constants depending only on m, n and W_0). Regarding Property 0 from Section 13, we would like to replace its use with Property 0′. In relation with the proof of Theorem 4, Property 0 was used only in (42) of Section 34. In that occurrence, the adaptation of the argument to fit into the proof of Theorem 7, using Property 0′ in place of Property 0 from Section 13, is very simple.

Thus, Theorem 7 follows along the lines of the proof of Theorem 4. We invite the reader to fill in the details.

We will conclude this section with a remark related to scale invariance. Let us denote $\tau_\delta(x) = \delta x$ for $\delta > 0$ and $x \in \mathbb{R}^n$. Suppose $F : \mathbb{R}^n \to \mathbb{R}$ is an unknown function, and that the only information we have regarding F is its C^m-norm, $\| F \|_{C^m(\mathbb{R}^n)}$. Suppose $\delta > 0$ is a known number, that may be very large or very small. Then it is impossible to guess what is even the order of magnitude of $\| F \circ \tau_\delta \|_{C^m(\mathbb{R}^n)}$, without having more information on the function F. In other words, the C^m-norm does not behave well under scaling. To overcome this irritating point, one might want to consider scaling-friendly versions of the C^m-norm. For instance, the semi-norm,

$$N_m(F) = \sup_{x \in \mathbb{R}^n} \max_{|\beta| = m} |\partial^\beta F(x)|$$

is an obvious candidate. Note that if $N_m(F) \le M$ for some function $F : \mathbb{R}^n \to \mathbb{R}$, then by Taylor's theorem,

(3) $J_x(F) - J_y(F) \in CMB(x, y)$ for all $x, y \in \mathbb{R}^n$.

Property (3) was almost the only property of the norm $\| \cdot \|_{C^m(\mathbb{R}^n)}$ that was relevant in this manuscript. It is thus possible to modify slightly our discussion, and obtain an extension algorithm, with respect to the semi-norm N_m. We will not carry out the details, but one may prove the following result.

Theorem 8. *Suppose we are given the following data:*

- *A finite set* $E \subset \mathbb{R}^n$.
- *For each* $x \in E$, *an* $(m-1)$*-jet* $f(x) \in \mathcal{P}$, *and a centrally-symmetric convex set* $\sigma(x) \subset \mathcal{P}$, *defined by at most, say,* $2\dim\mathcal{P}$ *linear inequalities.*

Assume that $\#(E) = N$, *and that for each* $x \in E$, *the set* $\sigma(x)$ *is Whitney* t*-convex at* x, *with Whitney constant* W_0.

Then, there exists $F \in C^m(\mathbb{R}^n)$ *with the following properties:*

(I) *If* $\tilde{F} \in C^m(\mathbb{R}^n)$ *and* $M > 0$ *satisfy*

$$N_m(\tilde{F}) \leq M \quad \textit{and} \quad J_x(\tilde{F}) \in f(x) + M\sigma(x) \quad \textit{for } x \in E,$$

then

$$N_m(F) \leq CM \quad \textit{and} \quad J_x(F) \in f(x) + CM\sigma(x) \quad \textit{for } x \in E.$$

Here, C *is a constant depending only on* m, n *and* W_0.

(II) *There is an algorithm, that takes the given data, performs one-time work, and then responds to queries.*

A query consists of a point $x \in \mathbb{R}^n$, *and the response to the query is the jet* $J_x^+(F)$.

The one-time work takes $C'N\log N$ *operations, and* $C'N$ *storage. The time to answer a query is* $C'\log N$.

Here, C' *is a constant depending only on* m *and* n.

Let us just mention briefly the main point of change between the proof of Theorem 8 and the proof of Theorem 4. Actually, all we need to do is define

$$\Gamma(x, 0, M) = f(x) + M\sigma(x), \quad \text{and} \quad \sigma(x, 0) = \sigma(x)$$

(compare with the definition from Section 10, or with (5) from Section 34). We may re-run our arguments, based on this new definition of $\Gamma(x, 0, M)$ and $\sigma(x, 0)$, and obtain a family of blobs and convex sets that satisfy the obvious analogues of Properties 0,. . . ,4 from Section 13. This leads to an analogue of the *Main Lemma* for Theorem 8.

To deduce Theorem 8 from the analogue of the *Main Lemma*, we may take advantage of scale-invariance and translation-invariance to assume that our set E is contained in a single dyadic cube of sidelength A_2^{-1}. This allows us to bypass the arguments in Section 34. Thus the proof of Theorem 8 is actually a rather straightforward generalization of the proof of Theorem 4. We omit the details.

Appendix - Computation in Finite Precision

37. Representing Real Numbers in the Computer

Our algorithms deal with real numbers. We need to store and retrieve real numbers from the computer memory; we add, subtract, multiply, divide and compare real numbers, and also, when computing the Caldéron-Zygmund cubes, we make use of the operations of logarithm, powers of two and rounding to the nearest integer.

As we are aiming at a rigorous, asymptotic analysis of our algorithms, we need to specify the precise abstract model of computation underlying the discussion. When we work with non-discrete objects, such as real numbers, selecting a computational model is not an obvious task.

A naïve approach, would be to consider a standard von Neumann computer, able to work with exact real numbers and perform all the above operations exactly, in infinite precision. It was brought to our attention that this model of computation, and in particular the unrestricted use of the "round to nearest integer" operation, leads to some suspiciously efficient algorithms. For example, it was shown in [28] (see also [23]) that in this model of computation there exists a polynomial-time algorithm, that solves a problem for which there is no known sub-exponential algorithm in the standard, discrete, models of computation.

Thus, some caution is needed when analyzing the performance of algorithms involving real numbers. In [19] we have described a simpler algorithm, and we have selected there a model of computation able to work with exact real numbers. In that model of computation, an operation is one of the following.

(1) An exact addition, subtraction, multiplication or division of real numbers.

(2) A comparison of two real numbers x and y, i.e. the decision as to whether $x > y, x = y$ or $x < y$.

(3) Reading or writing of a real number from a specified memory cell.

This model of computation, referred to as *real RAM* (RAM stands for Random Access Machine), is quite standard, see e.g. [27, Section 1.4]. It is also common to strengthen this model, by allowing exponents, logarithms and trigonometric functions (again see [27, Section 1.4]). We refer the reader e.g. to [24] for a critical discussion of this model. Unfortunately, this widely accepted model of computation does not suit the algorithms in this manuscript; we are currently unaware of an efficient computational approach to the space tilings $\mathsf{CZ}(\mathcal{A})$, that avoids the use of "rounding to the nearest integer".

There are several other reasonable models of computation, that are more appropriate for the analysis of our algorithms. We chose to use a finite-precision computer, in which a real number is represented, to some accuracy, using registers of $\bar{S}$ bits. It takes the computer one unit of time to perform simple manipulations on one or two registers. In particular, we may add, subtract, multiply, divide, round and compare $\bar{S}$-bit registers, within one unit of time. Since our registers are finite, these operations cannot be performed with perfect precision. We suppose, as in [25, Section 4.2.2], that these operations are as accurate as possible. We will assume explicit bounds for the error that may be caused by each operation. The list of allowed operations, and the assumptions we make on each of them, appear in the following section. In Sections 39–57 we provide a detailed analysis of the performance and accuracy of our algorithms, in the $\bar{S}$-bit-precision model of computation.

We have elected the finite-precision model of computation, since it seems to the authors close in spirit to our understanding of real-life digital computers. We would like to emphasize that the model of computation we chose is in no sense canonical. It is also possible to analyze our algorithms using other models. For instance, we could have considered a *real RAM*, with the addition of "rounding" and "powers of two" operations in a bounded domain; in this model all of the computations are exact, but we are allowed, for instance, to round real numbers to the nearest integer only if they lie in the interval $[0, S]$, for some given number S.

Our main goal in the analysis below is to verify that our algorithms are honest and make sense, and that we avoid the subtle problems related to computation with real numbers mentioned above. We confine attention to the main algorithm presented in this manuscript, i.e., the one which appears in the formulation of Theorem 2 or Theorem 4.

This discussion, of course, is of purely theoretical nature. It would be interesting to examine whether our algorithms, or at least some of the ideas in them, may be also of some practical use. Needless to say, all algorithms in this manuscript are well suited for implementation in any standard computer language, such as `FORTRAN`, `PASCAL` or `C`. A thorough study of an optimized implementation is beyond the scope of this article.

38. The Model of Computation

For an integer $\bar{S} \geq 1$, we work with "machine numbers" of the form $k \cdot 2^{-\bar{S}}$, with k an integer and $|k| \leq 2^{+2\bar{S}}$. Our model of computation is an idealized von Neumann computer [34], able to handle machine numbers. We also make the following assumptions:

- Two machine numbers x and y satisfying $|x| \leq 2^{\ell}$ and $|y| \leq 2^{\ell'}$ with

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

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 $|$"$1/x$" $- 1/x| \leq 2^{-\bar{S}}$.

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

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

- Two machine numbers x and y satisfying $|x| \leq \ell$ and $|y| \leq \ell'$ for integers ℓ and ℓ' such that $\ell + \ell' \leq 2^{\bar{S}}$ 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 $|\ell| \leq \bar{S}$, then we can compute the greatest integer $\leq 2^{\ell}x$. (If this integer lies outside $[-2^{\bar{S}}, +2^{\bar{S}}]$, 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 $\leq 2^{\bar{S}}$, 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 $2^{\bar{S}}$, then we obtain the greatest integer $\leq$ the real number x/y. If our desired answer lies outside $[-2^{\bar{S}}, +2^{\bar{S}}]$, then we produce an error message and abort our computation.

- Given integers x, y of absolute value $\leq 2^{\bar{S}}$, 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 $|\ell| \leq \bar{S}$, then we can compute exactly the machine number 2^ℓ. We assume this takes one unit of "work".
- We assume we can read or write a machine number from /to the RAM with one unit of "work".
- We assume we can store the address of any memory cell in a single $\bar{S}$-bit word.

Under these assumptions, we say that we work with "$\bar{S}$-bit machine numbers" (though the actual implementation of those machine numbers seems to require at least $2\bar{S}+2$ bits.) Note that it is possible to simulate arithmetic of $t\bar{S}$-bit machine numbers, using a computer working with $\bar{S}$-bit machine numbers. The amount of "work" for each elementary operation is a constant depending only on t. Consequently, we are only interested in the order of magnitude of $\bar{S}$.

We will show that when our algorithms receive their input as S_0-bit machine numbers, and if $\bar{S} \geq CS_0$, for a constant C depending only on m and n, then the output produced by our algorithm is accurate to within S_0 bits. We will verify that the work and storage required are as promised, $CN \log N$ for the one-time work and $C \log N$ for the query time, with CN storage, for C being a constant depending only on m and n. Since we are really interested only in the order of magnitude of $\bar{S}$, this implies that we can eventually take $\bar{S} = S_0$.

In Sections 39,…,57 we assume the above model of computation.

Throughout those sections, $\bar{S}$ will always denote the precision of our model of computation, as was just described.

39. Data Structures

Let $D \geq 1$ be given. We will work with ALPs in $\mathbb{R}^D$.

Let S be a positive integer, and let $\Upsilon \geq 1$. We define an "S-bit FALP with constant Υ" to be an ALP

(DS1) $\mathcal{A} = \left[(\lambda_{\ell j})_{\substack{1\leq\ell\leq L\\1\leq j\leq D}}, (b_\ell)_{1\leq\ell\leq L}, (\sigma_\ell)_{1\leq\ell\leq L}, M_*\right]$

in $\mathbb{R}^D$, with the following properties.

(DS2) $L \geq 1$.

(DS3) $2^{-S} \leq \sigma_\ell \leq 2^S$ for each ℓ $(1 \leq \ell \leq L)$.

(DS4) $2^{-S} \leq M_* \leq 2^S$.

(DS5) $|b_\ell| \leq 2^S$ for each ℓ $(1 \leq \ell \leq L)$.

(DS6) $|\lambda_{\ell j}| \le 2^S$ for each ℓ, j $(1 \le \ell \le L, 1 \le j \le D)$.

(DS7) Suppose we are given $\lambda'_{\ell j}$ $(1 \le \ell \le L, 1 \le j \le D)$ and b'_ℓ $(1 \le \ell \le L)$, such that for all ℓ, j we have

$$|\lambda'_{\ell j} - \lambda_{\ell j}| \le 2^{-S} \text{ and } |b'_\ell - b_\ell| \le 2^{-S}.$$

Then the ALP $\mathcal{A}$ is Υ-equivalent to the ALP

$$\mathcal{A}' = \left[(\lambda'_{\ell j})_{\substack{1\le\ell\le L\\1\le j\le D}}, (b'_\ell)_{1\le\ell\le L}, (\sigma_\ell)_{1\le\ell\le L}, M_*\right].$$

(Recall that two ALPs $\mathcal{A}, \mathcal{A}'$ are C-equivalent, if the blobs to which they give rise are C-equivalent.)

"FALP" stands for "Fault-tolerant ALP". If also each $\lambda_{\ell j}, b_\ell, \sigma_\ell$ and M_* in (DS1) is a machine number, then we say that $\mathcal{A}$ is an "S-bit MALP with constant Υ".

40. Remarks on FALPs and MALPs

Lemma 1. *Let*

$$\mathcal{A} = [(\lambda_{\ell j})_{\substack{1\le\ell\le L\\1\le j\le D}}, (b_\ell)_{1\le\ell\le L}, (\sigma_\ell)_{1\le\ell\le L}, M_*]$$

be an S-bit FALP *with constant Υ. Let*

$$\mathcal{A}' = [(\lambda'_{\ell j})_{\substack{1\le\ell\le L\\1\le j\le D}}, (b'_\ell)_{1\le\ell\le L}, (\sigma'_\ell)_{1\le\ell\le L}, M'_*],$$

with $|\lambda'_{\ell j} - \lambda_{\ell j}|$, $|b'_\ell - b_\ell|$, $|\sigma'_\ell - \sigma_\ell|$, $|M'_* - M_*| \le 2^{-(S+1)}$. Then $\mathcal{A}'$ is an $(S+1)$-bit FALP *with constant $4\Upsilon^2$. Moreover, $\mathcal{A}'$ is 2Υ-equivalent to $\mathcal{A}$.*

Proof. Let $\mathcal{A}'' = [(\lambda''_{\ell j}), (b''_\ell), (\sigma'_\ell), M'_*]$, with $|\lambda''_{\ell j} - \lambda'_{\ell j}|$, $|b''_\ell - b'_\ell| \le 2^{-(S+1)}$. We will show that $\mathcal{A}''$ is $4\Upsilon^2$-equivalent to $\mathcal{A}'$. To see this, note that $|\lambda''_{\ell j} - \lambda_{\ell j}|$, $|b''_\ell - b_\ell| \le 2^{-S}$. Hence, by (DS7), the ALPs

$$\tilde{\mathcal{A}}'' = [(\lambda''_{\ell j}), (b''_\ell), (\sigma_\ell), M_*] \quad \text{and} \quad \tilde{\mathcal{A}}' = [(\lambda'_{\ell j}), (b'_\ell), (\sigma_\ell), M_*]$$

are both Υ-equivalent to $\mathcal{A}$; hence they are Υ^2-equivalent to each other.

Since also $\frac{1}{2} \le \sigma'_\ell/\sigma_\ell \le 2$ and $\frac{1}{2} \le M'_*/M_* \le 2$, we know that $\mathcal{A}'$ is 2-equivalent to $\tilde{\mathcal{A}}'$, and similarly $\mathcal{A}''$ is 2-equivalent to $\tilde{\mathcal{A}}''$. Consequently, $\mathcal{A}'$ and $\mathcal{A}''$ are $4\Upsilon^2$-equivalent, which proves (DS7) for $\mathcal{A}'$, with $4\Upsilon^2$ and $S+1$ in place of Υ and S. Properties (DS2,...,DS6) hold trivially for $\mathcal{A}'$, with $S+1$ in place of S. Therefore, $\mathcal{A}'$ is an $(S+1)$-bit FALP with constant $4\Upsilon^2$. Also, since $\tilde{\mathcal{A}}'$ is Υ-equivalent to $\mathcal{A}$, and since $\tilde{\mathcal{A}}'$ is 2-equivalent to $\mathcal{A}'$, it follows that $\mathcal{A}$ is 2Υ-equivalent to $\mathcal{A}'$. ■

As a special case of the above, let

$$\mathcal{A} = [(\lambda_{\ell j}),\, (b_\ell),\, (\sigma_\ell),\, M_*]$$

be an S-bit FALP, with constant Υ.
Define

$$\mathcal{A}' = [(\lambda'_{\ell j}),\, (b_\ell),\, (\sigma_\ell),\, M_*]$$

by setting

$$\begin{aligned} \lambda'_{\ell j} &= \lambda_{\ell j} \text{ if } |\lambda_{\ell j}| > 2^{-(S+1)}, \\ \lambda'_{\ell j} &= 0 \text{ if } |\lambda_{\ell j}| \le 2^{-(S+1)}. \end{aligned}$$

Then $\mathcal{A}'$ is an $(S+1)$-bit FALP with constant $4\Upsilon^2$, and, moreover, $\mathcal{A}$ and $\mathcal{A}'$ are 2Υ-equivalent.

Passing from $\mathcal{A}$ to $\mathcal{A}'$, we can ensure that any non-zero $\lambda'_{\ell j}$ will have absolute value at least $2^{-(S+1)}$.

We give the name "Rounding Down" to the process of passing from $\mathcal{A}$ to $\mathcal{A}'$ as above.

If $\mathcal{A}$ is an S-bit MALP with constant Υ and length L in $\mathbb{R}^D$, then we can "round $\mathcal{A}$ down" in our model of computation, with work at most CDL, where C is a universal constant. (Of course, when $S \ge \bar{S}$, rounding down of an S-bit MALP requires no computer operations at all; in this case, $\mathcal{A} = \mathcal{A}'$.)

Lemma 2. *Suppose*

$$\mathcal{A} = \left[(\lambda_{\ell j})_{\substack{1\le \ell \le L\\ 1\le j\le D}},\, (b_\ell)_{1\le\ell\le L}, (\sigma_\ell)_{1\le\ell\le L}, M_*\right]$$

is an S-bit FALP *with constant Υ. Then the matrix $(\lambda_{\ell j})$ has rank D. (In particular, $L \ge D$.)*

Proof. Suppose not. Then there exists $v^0 \in \mathbb{R}^D$ with $v^0 = (v^0_1, \dots, v^0_D) \ne 0$, yet $\sum_j \lambda_{\ell j} v^0_j = 0$ for each ℓ. Fix j_0 with $v^0_{j_0} \ne 0$, and fix $M \ge M_*$, with $M\sigma_\ell \ge |b_\ell|$ for each ℓ. (Recall that $\sigma_\ell \ne 0$ since $\mathcal{A}$ is an S-bit FALP. Hence, we can find such an M.)

Set

$$\mathcal{A}' = [(\lambda_{\ell j} + 2^{-S}\delta_{\ell 1}\delta_{j j_0})_{\substack{1\le \ell \le L\\ 1\le j\le D}},\, (b_\ell)_{1\le\ell\le L},\, (\sigma_\ell)_{1\le\ell\le L}, M_*],$$

where δ is the Kronecker delta. Since $\mathcal{A}$ is an S-bit FALP, the ALPs $\mathcal{A}, \mathcal{A}'$ must be Υ-equivalent. In particular, for our M, we have $K_M(\mathcal{A}) \subseteq K_{\Upsilon M}(\mathcal{A}')$. On the other hand, for any $T \in \mathbb{R}$, we have $Tv^0 \in K_M(\mathcal{A})$, since for each ℓ we know that

$$\left|\sum_j \lambda_{\ell j} T v^0_j - b_\ell\right| = |b_\ell| \le M\sigma_\ell.$$

Hence, $T v^0 \in K_{\Upsilon M}(\mathcal{A}')$ for any $T \in \mathbb{R}$. In particular,

$$\left|\sum_j (\lambda_{1j} + 2^{-S}\delta_{jj_0}) T v_j^0 - b_1\right| \leq \Upsilon M \sigma_1$$

for all $T \in \mathbb{R}$. That is, $|(2^{-S} v_{j_0}^0) \cdot T - b_1| \leq \Upsilon M \sigma_1$ for all $T \in \mathbb{R}$. That's absurd, since S, $v_{j_0}^0$, b_1, M, σ_1, Υ are independent of T, and $v_{j_0}^0 \neq 0$. The proof is complete. ■

Lemma 3. *Let* $S \geq 1$ *be an integer, and let*

$$\mathcal{A} = [(\lambda_{\ell j})_{\substack{1 \leq \ell \leq D \\ 1 \leq j \leq D}}, (b_\ell)_{1 \leq \ell \leq D}, (\sigma_\ell)_{1 \leq \ell \leq D}, M_*]$$

be an ALP *of length* D *in* $\mathbb{R}^D$*, satisfying the following:*

(a) *All numbers* $|\lambda_{\ell j}|, |b_\ell|, \sigma_\ell, M_*$ are $\leq 2^S$;

(b) *All numbers* σ_ℓ, M_* *are* $\geq 2^{-S}$;

(c) $|\det(\lambda_{\ell j})| \geq 2^{-S}$.

Then $\mathcal{A}$ *is an* S'*-bit* FALP *with constant* Υ*, where* $S' = \hat{C}S$*,* $\Upsilon = 2$*, and* $\hat{C}$ *depends only on* D.

Proof. Let C, C', etc. denote constants depending only on D.
Let $\mathcal{A}' = [(\lambda_{\ell j} + \mu_{\ell j}), (b_\ell + \beta_\ell), (\sigma_\ell), M_*]$ with $|\mu_{\ell j}|, |\beta_\ell| < \epsilon$, and with $\epsilon > 0$ to be picked below.

We will show that the ALPs $\mathcal{A}$ and $\mathcal{A}'$ are 2-equivalent. To see this, let $M \geq M_*$. Then

$$K_M(\mathcal{A}) = \left\{v = (v_1, \ldots, v_D) \in \mathbb{R}^D : \left|\sum_j \lambda_{\ell j} v_j - b_\ell\right| \leq M\sigma_\ell \text{ for } \ell = 1, \ldots, D\right\}$$

and

$$K_M(\mathcal{A}') = \Bigg\{v = (v_1, \ldots, v_D) \in \mathbb{R}^D : \\ \left|\left[\sum_j \lambda_{\ell j} v_j - b_\ell\right] + \left[\sum_j \mu_{\ell j} v_j - \beta_\ell\right]\right| \leq M\sigma_\ell \;\text{ for } \ell = 1, \ldots, D\Bigg\}.$$

We have $|\sum_j \mu_{\ell j} v_j - \beta_\ell| \leq C\epsilon(1 + |v|)$ for any $v \in \mathbb{R}^D$. Suppose $v \in K_M(\mathcal{A})$. Then $|\sum_j \lambda_{\ell j} v_j| \leq |b_\ell| + M\sigma_\ell \leq 2^S(1+M) \leq C \cdot 2^{2S} M$ for each ℓ, thanks to our assumptions on b_ℓ, σ_ℓ, M_*.

Since $|\lambda_{\ell j}| \le 2^S$ and $|\det(\lambda_{\ell j})| \ge 2^{-S}$, it follows that

$$|v| \le C \cdot 2^{CS} M\,,$$

hence

$$\Big|\sum\nolimits_j \mu_{\ell j} v_j - \beta_\ell\Big| \le C\epsilon(1+|v|) \le 2^{-S}M \le M\sigma_\ell$$

provided we take $\epsilon < 2^{-CS}$ for large enough C. (Here again we use our assumptions on the size of M_*, σ_ℓ.) Thus,

$$\Big|\sum\nolimits_j \lambda_{\ell j} v_j - b_\ell\Big|,\ \Big|\sum\nolimits_j \mu_{\ell j} v_j - \beta_\ell\Big| \le M\sigma_\ell \quad \text{for each } \ell,$$

and consequently $v \in K_{2M}(\mathcal{A}')$. Hence, $v \in K_M(\mathcal{A})$ implies $v \in K_{2M}(\mathcal{A}')$.

Conversely, suppose $v \in K_M(\mathcal{A}')$. Then

$$\Big|\sum_j \lambda_{\ell j} v_j\Big| \le |b_\ell| + \Big[|\beta_\ell| + \sum_j |\mu_{\ell j}|\,|v_j|\Big] + M\sigma_\ell \le 2^S + C\epsilon(1+|v|) + M\sigma_\ell\ ,$$

for each ℓ. Since also $|\lambda_{\ell j}| \le 2^S$ and $|\det(\lambda_{\ell j})| \ge 2^{-S}$, it follows that

$$|v| \le C2^{CS} \cdot [1 + C\epsilon(1+|v|) + M\sigma_\ell] \le C'2^{CS} \cdot [1 + \epsilon|v| + M\sigma_\ell]\,.$$

Taking $\epsilon < 2^{-C''S}$, we can absorb the $\epsilon|v|$ on the far right into the left-hand side. Hence, $|v| \le C \cdot 2^{C \cdot S} \cdot M$. This in turn yields

$$|\beta_\ell| + \sum_j |\mu_{\ell j}|\,|v_j| \le C2^{C\cdot S}\epsilon M \le 2^{-S}M \le M\sigma_\ell\,,$$

provided we take $\epsilon < 2^{-C'''S}$ for a large enough C'''. (Here again, we use our hypothesis on the size of M_* and σ_ℓ.) Now we know that

$$\Big|\Big(\sum_j \lambda_{\ell j} v_j - b_\ell\Big) + \Big(\sum_j \mu_{\ell j} v_j - \beta_\ell\Big)\Big| \le M\sigma_\ell\,, \text{and}$$

$$\Big|\sum_j \mu_{\ell j} v_j - \beta_\ell\Big| \le |\beta_\ell| + \sum_j |\mu_{\ell j}||v_j| \le M\sigma_\ell\,,$$

for all ℓ. Hence,

$$\Big|\sum\nolimits_j \lambda_{\ell j} v_j - b_\ell\Big| \le 2M\sigma_\ell,$$

i.e., $v \in K_{2M}(\mathcal{A})$. Thus, $v \in K_M(\mathcal{A}')$ implies $v \in K_{2M}(\mathcal{A})$.

We have shown that $\mathcal{A}$ and $\mathcal{A}'$ are 2-equivalent, provided $\epsilon = 2^{-C\cdot S}$ for a large enough integer constant C, depending only on D.

This means that $\mathcal{A}$ satisfies (DS7), with $\hat{C} \cdot S$ in place of S, and with $\Upsilon = 2$. Also, we see that $\mathcal{A}$ satisfies (DS2...6), with $\hat{C} \cdot S$ in place of S. Thus, $\mathcal{A}$ is a $C \cdot S$ bit FALP, with $\Upsilon = 2$. The proof is complete. ■

As an example of the previous result, let $S_0 \geq 1$ be an integer, and assume that

(*) $\quad S_0 \geq C$ and $\bar{S} \geq CS_0$,

where C is a large enough constant depending only on D. Let δ be a machine number with $2^{-S_0} \leq \delta \leq 2^{+S_0}$, and let $\mathcal{B}_\delta$ be the ALP in $\mathbb{R}^D$ given by:

(a) $\quad \lambda_{\ell j} = \delta_{\ell j}$ (Kronecker delta) for $\ell, j = 1, \dots, D$;

(b) $\quad b_\ell = 0$ for $\ell = 1, \dots, D$;

(c) $\quad \sigma_\ell :=$ "δ^{m_ℓ}" for $\ell = 1, \dots, D$, where m_ℓ is an integer, and $0 \leq m_\ell \leq D$ for each ℓ;

(d) $\quad M_* = 2^{-S_0}$.

In (c), we attempt to compute δ^{m_ℓ} using our model of computation, so σ_ℓ will equal δ^{m_ℓ} plus a roundoff error. The roundoff error will be smaller than $\frac{1}{2}\delta^{m_\ell}$ because of (*).

Then the ALP $\mathcal{B}_\delta$ is a CS_0-bit MALP with constant 2, where C depends only on D. This follows at once from Lemma 3. The computation of the MALP $\mathcal{B}_\delta$ requires C work, with C depending only on D.

Lemma 4. *Let $\mathcal{A}$ be an S-bit* FALP *with constant Υ in $\mathbb{R}^D$. Let $M > 0$.*

Suppose $u, v \in \mathbb{R}^D$, with $v \in K_M(\mathcal{A})$ and $|u| \leq 2^{-2S}/D$.

Then $v + u \in K_{\Upsilon M}(\mathcal{A})$.

Proof. Let $u = (u_1, \dots, u_D)$, $v = (v_1, \dots, v_D)$, $\mathcal{A} = [(\lambda_{\ell j}), (b_\ell), (\sigma_\ell), M_*]$. By hypothesis, we have $|\lambda_{\ell j}| \leq 2^S$ and $|u_j| \leq \frac{2^{-2S}}{D}$, and therefore, setting

$$b'_\ell = b_\ell + \sum\nolimits_{j=1}^{D} \lambda_{\ell j} u_j,$$

we obtain $|b'_\ell - b_\ell| \leq 2^{-S}$ for each ℓ. Since $\mathcal{A}$ is an S-bit FALP with constant Υ, it follows that $\mathcal{A}$ and $\mathcal{A}'$ are Υ-equivalent, where $\mathcal{A}' = [(\lambda_{\ell j}), (b'_\ell), (\sigma_\ell), M_*]$.

Since $v \in K_M(\mathcal{A})$, we know that $M \geq M_*$, and $|\sum_j \lambda_{\ell j} v_j - b_\ell| \leq M\sigma_\ell$ for each ℓ.

Since

$$\sum\nolimits_j \lambda_{\ell j}(v_j + u_j) - b'_\ell = \sum\nolimits_j \lambda_{\ell j} v_j - b_\ell,$$

it follows that $|\sum_j \lambda_{\ell j}(v_j + u_j) - b'_\ell| \leq M\sigma_\ell$ for each ℓ, with $M \geq M_*$, i.e., $v + u \in K_M(\mathcal{A}')$. Recalling that $\mathcal{A}'$ and $\mathcal{A}$ are Υ-equivalent, we conclude that $v + u \in K_{\Upsilon M}(\mathcal{A})$. ■

41. Elementary Row Operations on FALPs

Permuting rows. Suppose $\mathcal{A}$ is an S-bit FALP with constant Υ, and let π be a permutation of the rows of $\mathcal{A}$.

Then $\mathcal{A}^\pi$ is again an S-bit FALP with constant Υ. Moreover, $\mathcal{A}^\pi$ is 1-equivalent to $\mathcal{A}$ (i.e., they give rise to the same blob).

Stripping away zeros. Suppose

$$\mathcal{A} = [(\lambda_{\ell j})_{\substack{1\le\ell\le L\\1\le j\le D}}, (b_\ell)_{1\le\ell\le L}, (\sigma_\ell)_{1\le\ell\le L}, M_*]$$

is an S-bit MALP with constant Υ, and suppose $\lambda_{\ell j} = 0$ whenever $\bar{L} < \ell \le L$, $1 \le j \le D$. Then

$$\tilde{\mathcal{A}} = [(\lambda_{\ell j})_{\substack{1\le\ell\le\bar{L}\\1\le j\le D}}, (b_\ell)_{1\le\ell\le\bar{L}}, (\sigma_\ell)_{1\le\ell\le\bar{L}}, \tilde{M}_*]$$

is the ALP arising from $\mathcal{A}$ by "stripping away zeros", where

$$\tilde{M}_* = \max\{M_*, \max\{|b_\ell|/\sigma_\ell :\ \bar{L} < \ell \le L\}\}.$$

(Recall that all σ_ℓ are non-zero.)

Trivially, $\tilde{\mathcal{A}}$ is a $2S$-bit FALP with constant Υ, because $|b_\ell|/\sigma_\ell \le 2^{2S}$, and hence $\tilde{M}_* \le 2^{2S}$. The ALPs $\mathcal{A}$ and $\tilde{\mathcal{A}}$ are 1-equivalent. We make here the assumption that

(0) $S \ge 100$ and $\bar{S} \ge 100S$.

Thus, in our model of computation, we can compute $\tilde{M}_*$ only up to a round-off error of absolute value $\le 2^{-\bar{S}} \le 2^{-10S}$. Our attempt to compute $\tilde{\mathcal{A}}$ with imperfect arithmetic yields

$$\tilde{\mathcal{A}}' = \left[(\lambda_{\ell j})_{\substack{1\le\ell\le\bar{L}\\1\le j\le D}}, (b_\ell)_{1\le\ell\le\bar{L}}, (\sigma_\ell)_{1\le\ell\le\bar{L}}, \tilde{M}'_*\right]$$

with

$$|\tilde{M}'_* - \tilde{M}_*| \le 2^{-10S}.$$

By Lemma 1 in Section 40, we see that $\tilde{\mathcal{A}}'$ is a $(2S+1)$-bit MALP with constant $4\Upsilon^2$. Also, obviously, $\tilde{\mathcal{A}}'$ is 2-equivalent to $\tilde{\mathcal{A}}$. Consequently,

(a) $\tilde{\mathcal{A}}'$ and $\mathcal{A}$ are 2-equivalent

(b) $\tilde{\mathcal{A}}'$ can be computed from $\mathcal{A}$ in our model of computation; the work is $\le CDL$, with C a universal constant

(c) $\tilde{\mathcal{A}}'$ is a $(2S+1)$-bit MALP with constant $4\Upsilon^2$.

Thus, we can "strip away zeros" in our model of computation.

Addition of rows. Let

$$\mathcal{A} \;=\; \left[(\lambda_{\ell j})_{\substack{1\le\ell\le L\\ 1\le j\le D}}\,,(b_\ell)_{1\le\ell\le L}\,,(\sigma_\ell)_{1\le\ell\le L}\,,M_*\right]$$

be an S-bit MALP with constant Υ. Assume that

(1) $\quad S \ge 1000$ and $\bar{S} \ge 1000S$.

Suppose we are given an integer ℓ_0 $(1 \le \ell_0 \le L)$ and machine numbers β_ℓ $(\ell = 1,\dots,L)$, satisfying:

(2) $\quad \beta_{\ell_0} = 0$;

(3) $\quad |\beta_\ell| \;\le\; 2^{2S}$ for all ℓ; and

(4) $\quad |\beta_\ell|\cdot\sigma_{\ell_0} \;\le\; 2\sigma_\ell$ for all ℓ.

Let

$$\tilde{\mathcal{A}} = \left[(\lambda_{\ell j} + \beta_\ell\lambda_{\ell_0 j})_{\substack{1\le\ell\le L\\ 1\le j\le D}}\,,\,(b_\ell + \beta_\ell b_{\ell_0})_{1\le\ell\le L}\,,\,(\sigma_\ell)_{1\le\ell\le L},M_*\right],$$

and let $\tilde{\mathcal{A}}'$ be the ALP arising from attempting to compute $\tilde{\mathcal{A}}$ in our model of computation.

Thus,

$$\tilde{\mathcal{A}}' = \left[(\lambda_{\ell j} + \beta_\ell\lambda_{\ell_0 j} + \epsilon_{\ell j})\,,\,(b_\ell + \beta_\ell b_{\ell_0} + \epsilon_\ell)\,,\,(\sigma_\ell)\,,M_*\right],$$

where the ϵ's are round-off errors; we have $|\epsilon_{\ell j}|$, $|\epsilon_\ell| \le 2^{-\bar{S}} \le 2^{-10^3 S}$,

All entries of $\tilde{\mathcal{A}}'$ are machine numbers. Note that $\mathcal{A}$ and $\tilde{\mathcal{A}}$ are 3-equivalent. In fact, suppose $\nu \in K_M(\mathcal{A})$. Then $M \ge M_*$; and we have

$$\left|\sum\nolimits_j \lambda_{\ell j}\nu_j - b_\ell\right| \le M\sigma_\ell \quad \text{for each } \ell,$$

and therefore

$$\left|\left[\sum_j \lambda_{\ell j}\nu_j - b_\ell\right] \;+\; \beta_\ell\left[\sum_j \lambda_{\ell_0 j}\nu_j - b_{\ell_0}\right]\right| \;\le\; M\sigma_\ell + M|\beta_\ell|\sigma_{\ell_0} \;\le\; 3M\sigma_\ell$$

by assumption (4). Hence, $\nu \in K_{3M}(\tilde{\mathcal{A}})$. Since $\beta_{\ell_0} = 0$, we can also write

$$\lambda_{\ell j} \;=\; \tilde{\lambda}_{\ell j} - \beta_{\ell_0}\tilde{\lambda}_{\ell_0 j},\; b_\ell \;=\; \tilde{b}_\ell - \beta_{\ell_0}\tilde{b}_{\ell_0}\,,$$

where $\tilde{\lambda}_{\ell j}$ and $\tilde{b}_\ell$ denote the entries of $\tilde{\mathcal{A}}$. Hence the same argument as above shows that

$$\nu \in K_M(\tilde{\mathcal{A}}) \text{ implies } \nu \in K_{3M}(\mathcal{A})\,.$$

Thus, $\mathcal{A}$ and $\tilde{\mathcal{A}}$ are 3-equivalent, as claimed. We will check that $\tilde{\mathcal{A}}$ is a $100S$-bit FALP, with constant 9Υ. To see this, we first recall the definition:

$$\tilde{\mathcal{A}} = [(\tilde{\lambda}_{\ell j}), (\tilde{b}_\ell), (\sigma_\ell), M_*]$$

with

$$\tilde{\lambda}_{\ell j} = \lambda_{\ell j} + \beta_\ell \lambda_{\ell_0 j}, \; \tilde{b}_\ell = b_\ell + \beta_\ell b_{\ell_0}.$$

Since $|\lambda_{\ell j}| \leq 2^S$ for all ℓ, j, and also $|b_\ell| \leq 2^S$ for all ℓ, and since $|\beta_\ell| \leq 2^{2S}$, we find that

$$|\tilde{\lambda}_{\ell j}|, |\tilde{b}_\ell| \leq 2^S + 2^{2S} \cdot 2^S \leq 2^{3S+1}.$$

Also, M_* and $\sigma_\ell \geq 2^{-S} > 2^{-(3S+1)}$. In addition, $\tilde{\mathcal{A}}$ has length $L \geq 1$, since the same is true of $\mathcal{A}$. Now suppose we compare $\tilde{\mathcal{A}}$ with

$$\tilde{\mathcal{A}}^{\#} = [(\tilde{\lambda}_{\ell j} + \theta_{\ell j}), (\tilde{b}_\ell + \theta_\ell), (\sigma_\ell), M_*],$$

with $|\theta_{\ell j}|, |\theta_\ell| \leq 2^{-100S}$.

We know by our previous argument applied to $\tilde{\mathcal{A}}^{\#}$, that $\tilde{\mathcal{A}}^{\#}$ is 3-equivalent to

$$\begin{aligned}\mathcal{A}^{\#} &= [(\tilde{\lambda}_{\ell j} + \theta_{\ell j} - \beta_\ell[\tilde{\lambda}_{\ell_0 j} + \theta_{\ell_0 j}]), (\tilde{b}_\ell + \theta_\ell - \beta_\ell[\tilde{b}_{\ell_0} + \theta_{\ell_0}]), (\sigma_\ell), M_*] \\ &= [(\lambda_{\ell j} + \{\theta_{\ell j} - \beta_\ell \theta_{\ell_0 j}\}), (b_\ell + \{\theta_\ell - \beta_\ell \theta_{\ell_0}\}), (\sigma_\ell), M_*],\end{aligned}$$

and the quantities in curly brackets are less than 2^{-S}.

Since $\mathcal{A}$ is an S-bit MALP with constant Υ, it follows that $\mathcal{A}^{\#}$ is Υ-equivalent to $\mathcal{A}$, which in turn is 3-equivalent to $\tilde{\mathcal{A}}$.

So: $\tilde{\mathcal{A}}^{\#}$ is 3-equivalent to $\mathcal{A}^{\#}$;
$\mathcal{A}^{\#}$ is Υ-equivalent to $\mathcal{A}$; and
$\mathcal{A}$ is 3-equivalent to $\tilde{\mathcal{A}}$.

Thus, $\tilde{\mathcal{A}}$ is 9Υ-equivalent to $\tilde{\mathcal{A}}^{\#}$, which proves (DS7) for $\tilde{\mathcal{A}}$, with 9Υ and $100S$ in place of Υ and S.
It follows that $\tilde{\mathcal{A}}$ is a $100S$-bit FALP with constant 9Υ.

Now, comparing $\tilde{\mathcal{A}}$ to $\tilde{\mathcal{A}}'$, and invoking Lemma 1 from Section 40, we see that $\tilde{\mathcal{A}}'$ is a $101S$-bit MALP, with constant $4 \cdot (9\Upsilon)^2$; and that $\tilde{\mathcal{A}}'$ is $2 \cdot (9\Upsilon)$-equivalent to $\tilde{\mathcal{A}}$, which is 3-equivalent to $\mathcal{A}$. Summarizing, we have the following results:

(A) $\tilde{\mathcal{A}}'$ is computable with work $\leq CDL$ in our model of computation, where C is a universal constant.

(B) $\tilde{\mathcal{A}}'$ and $\mathcal{A}$ are 200Υ-equivalent.

(C) $\tilde{\mathcal{A}}'$ is a $101S$-bit MALP with constant $10^3\Upsilon^2$.

Thus, we may perform "addition of rows" in our model of computation.

42. Echelon Form

(0) Let $\mathcal{A} = \left[(\lambda_{\ell j})_{\substack{1\leq\ell\leq L\\1\leq j\leq D}}, (b_\ell)_{1\leq\ell\leq L}, (\sigma_\ell)_{1\leq\ell\leq L}, M_*\right]$

be an S-bit FALP with constant Υ. Suppose we are given $\bar{L}$ with $0 \leq \bar{L} \leq D$. (Recall that $L \geq D$ for a FALP.)

We say that $\mathcal{A}$ is "in echelon form through row $\bar{L}$" if the following hold.

(EF1) $|\lambda_{\ell\ell}| \geq 2^{-S}$ for all $\ell \leq \bar{L}$.

(EF2) $\lambda_{\ell j} = 0$ for $\ell \leq \bar{L}$, $1 \leq j < \ell$.

(EF3) $\lambda_{\ell j} = 0$ for $\ell > \bar{L}$, $1 \leq j \leq \bar{L}$.

(Note that our definition of Echelon form for FALPs is slightly different from the one we used for ALPs; the pivots are no longer flexible. In this entire Appendix, we work only with the FALPs definition.)

If $\mathcal{A}$ is in echelon form through row L= length $(\mathcal{A})$, then we say it is in "echelon form".

Note that any S-bit FALP $\mathcal{A}$ is in echelon form through row zero, since (EF1, 2, 3) then hold vacuously.

Note also that any S-bit FALP $\mathcal{A}$ in echelon form through row $\bar{L}$ satisfies $D \geq \bar{L}$. Hence, any S-bit FALP $\mathcal{A}$ in echelon form satisfies $D \geq L$. Since we noted that $L \geq D$ for any FALP, it follows that length $(\mathcal{A}) = L = D$ for an S-bit FALP in echelon form.

Lemma 1. *Let $\mathcal{A}$ as in* (0) *be an S-bit* MALP *with constant Υ, in echelon form through row $\bar{L}$. Assume that*

(1) $S \geq 1000$ and $\bar{S} \geq 1000S$.

Then there exist S', Υ', and an ALP

(2) $\mathcal{A}' = [(\lambda'_{\ell j})_{\substack{1\leq\ell\leq L\\1\leq j\leq D}}, (b'_\ell)_{1\leq\ell\leq L}, (\sigma'_\ell)_{1\leq\ell\leq L}, M'_*]$,

with the following properties:

(3) *$\mathcal{A}'$ is an S'-bit* MALP *with constant Υ'.*

(4) *$S' = CS$ where C is a universal constant.*

(5) *Υ' is determined by Υ.*

(6) $\underline{\text{Either}}$ $\bar{L} = D$ $\underline{\text{or}}$ *$\mathcal{A}'$ is in echelon form through row $\bar{L} + 1$.*

(7) *$\mathcal{A}$ and $\mathcal{A}'$ are Υ'-equivalent.*

(8) *In our model of computation, $\mathcal{A}'$ can be computed from $\mathcal{A}$ with work $\leq C_0 DL$, for a universal constant C_0.*

Proof. First, we "round down". Thus, without loss of generality, we may suppose

(9) $|\lambda_{\ell j}| \geq 2^{-S}$ whenever $\lambda_{\ell j} \neq 0$.

If $\bar{L} = D$, then we may simply take $\mathcal{A}' = \mathcal{A}$, and properties (3), ..., (8) hold trivially.
Suppose $\bar{L} \neq D$. Then in fact $\bar{L} < D$. Since $\mathcal{A}$ is an S-bit FALP, we know that the matrix $(\lambda_{\ell j})$ has rank D. If we had $\lambda_{\ell,\bar{L}+1} = 0$ for all $\ell > \bar{L}$, then thanks to (EF2,3), any $D \times D$ submatrix of $(\lambda_{\ell j})$ would have determinant zero, as we see by expanding by minors using successively columns $1, 2, \ldots, \bar{L} + 1$. Consequently, $\lambda_{\ell,\bar{L}+1} \neq 0$ for some $\ell \geq \bar{L}+1$. For all $\ell \geq \bar{L}+1$ with $\lambda_{\ell,\bar{L}+1} \neq 0$, we compute

(10) $\sigma_\ell/|\lambda_{\ell,\bar{L}+1}|$ (with a roundoff error),

and pick ℓ_0 yielding the minimum computed ratio. Note that $2^{-S} \leq \sigma_\ell \leq 2^{+S}$, and $2^{-S} \leq |\lambda_{\ell,\bar{L}+1}| \leq 2^{+S}$ here, thanks to (9) and to (DS3), (DS6) from Section 39.

According to (1), the quotients (10) are computed to within a small percentage error. Hence, the ℓ_0 chosen by our computer will satisfy

(11) $\left|\frac{\lambda_{\ell,\bar{L}+1}}{\lambda_{\ell_0,\bar{L}+1}}\right| \sigma_{\ell_0} \leq 1.01\, \sigma_\ell$ for all ℓ.

By permuting rows, we may assume without loss of generality that $\ell_0 = \bar{L}+1$. In particular, $\lambda_{\bar{L}+1,\bar{L}+1} \neq 0$, so that (9) yields

(12) $|\lambda_{\bar{L}+1,\bar{L}+1}| \geq 2^{-S}$.

We now define $\beta_\ell = 0$ for $\ell \leq \bar{L}+1$, and $\beta_\ell = -\lambda_{\ell,\bar{L}+1}/\lambda_{\ell_0,\bar{L}+1}$ (as computed) for $\ell > \bar{L}+1$. Note that β_ℓ is computed to within a small percentage error by our computer, since either $\lambda_{\ell,\bar{L}+1} = 0$ or else numerator and denominator have absolute values between 2^{-S} and 2^{+S}, where $1000S \leq \bar{S}$. Therefore, (11) implies

(13) $|\beta_\ell|\sigma_{\ell_0} \leq 2\sigma_\ell$ for each ℓ, and we have $|\beta_\ell| \leq 2^{2S}$.

(Recall that by our assumptions from Section 38, $|x| \leq 2^\ell, |y| \leq 2^{\ell'} \Rightarrow |x \otimes y| \leq 2^{\ell+\ell'}$, and $|x| \geq 2^{-\ell} \Rightarrow |``1/x"| \leq 2^\ell$.)

Also,

(14) $\lambda_{\ell,\bar{L}+1} + \beta_\ell\lambda_{\ell_0,\bar{L}+1}$, as computed by our computer, will have absolute value at most $2^{100S} \cdot 2^{-\bar{S}}$.

We now perform "addition of rows" on $\mathcal{A}$, using the coefficients $\beta_\ell(1 \leq \ell \leq L)$, as explained in Section 41. This is allowed, thanks to (13) and to (1). We obtain an ALP $\mathcal{A}'$ satisfying (2) ,..., (5), as well as (7), (8). Moreover, $\mathcal{A}'$ is a $101S$-bit MALP, with constant $10^3\Upsilon$ that is 200Υ-equivalent to $\mathcal{A}$. Regarding (6), we would be happy if $\mathcal{A}'$ were in echelon form through row $\bar{L}+1$. Unfortunately, that isn't true, because of roundoff errors. More precisely, we have

$$\begin{aligned} |\lambda'_{\ell\ell}| \geq 2^{-S} &\quad \text{for } \ell = 1, \dots, \bar{L}+1, \\ \lambda'_{\ell j} = 0 &\quad \text{for } j < \ell \leq \bar{L}+1 \\ \lambda'_{\ell j} = 0 &\quad \text{for } j \leq \bar{L} \text{ and } \ell > \bar{L}+1 \end{aligned}$$

but merely

$$|\lambda'_{\ell,\bar{L}+1}| \leq 2^{100S} 2^{-\bar{S}} \text{ for } \ell > \bar{L}+1 \,.$$

To achieve (6), we "round down" our MALP $\mathcal{A}'$, regarding it as a $101S$-bit MALP, so that any $\lambda'_{\ell j}$ with $|\lambda'_{\ell j}| \leq 2^{-102S}$ is redefined to be zero. We obtain from $\mathcal{A}'$ the $102S$-bit MALP

$$\mathcal{A}'' = [(\lambda''_{\ell j}), (b''_\ell), (\sigma''_\ell), M''_*],$$

with the following properties:

$$\text{(15)} \qquad \left[\begin{array}{lcl} |\lambda''_{\ell\ell}| & \geq & 2^{-102S} \text{ for all } \ell = 1, \dots, \bar{L}+1 \,, \\ \lambda''_{\ell j} & = & 0 \text{ for } j < \ell \leq \bar{L}+1 \\ \lambda''_{\ell j} & = & 0 \text{ for } j \leq \bar{L}+1 \text{ and } \ell > \bar{L}+1 \end{array} \right]$$

Here, $\mathcal{A}''$ is a $102S$-bit MALP with constant Υ'' determined by Υ. Moreover, $\mathcal{A}''$ is 2000Υ-equivalent to $\mathcal{A}'$, which is 200Υ-equivalent to $\mathcal{A}$. Thus, enlarging Υ'', we find that

(16) $\mathcal{A}''$ is Υ''-equivalent to $\mathcal{A}$,

and that

(17) $\mathcal{A}''$ is a $102S$-bit MALP with constant Υ'', where

(18) Υ'' is determined by Υ.

In view of (15),...,(18) and the definition of Echelon form through row $\bar{L}+1$, we see that conclusions (2),...,(7) all hold for $\mathcal{A}''$. Moreover, reviewing how we obtained $\mathcal{A}''$ from $\mathcal{A}$, we see easily that (8) holds as well. The proof of the lemma is complete. ■

Repeatedly applying Lemma 1, then stripping away zeros, as for ALPs, we obtain the following result.

Lemma 2. *Let $\mathcal{A}$ be an S-bit MALP with constant Υ, of length L in $\mathbb{R}^{\mathsf{D}}$. Assume that*

(a) $\mathsf{S} \geq \hat{\mathsf{C}}$ *and* $\bar{\mathsf{S}} \geq \hat{\mathsf{C}}\mathsf{S}$ *where* $\hat{\mathsf{C}}$ *is a constant depending only on* D.

Then there exist S', Υ', *and an* S'*-bit MALP* $\mathcal{A}'$ *with constant* Υ', *satisfying the following conditions:*

(b) $\mathsf{S}' = \mathsf{C}\mathsf{S}$, *where* C *is an integer constant depending only on* D.

(c) Υ' *is determined by* Υ *and* D.

(d) $\mathcal{A}$ *and* $\mathcal{A}'$ *are* Υ'*-equivalent.*

(e) *As an* S'*-bit MALP,* $\mathcal{A}'$ *is in echelon form.*

(f) *In our model of computation,* $\mathcal{A}'$ *can be computed from* $\mathcal{A}$, *with work at most* $\mathsf{C}\mathsf{D}^2\mathsf{L}$, *where* C *is a universal constant.*

Note that we need to apply Lemma 1 at most D times, in order to prove Lemma 2. Thus, Lemma 2 holds provided we take $\hat{\mathsf{C}} \geq 10^5\mathsf{C}^{\mathsf{D}}$, where C is the constant from (4).

43. Applications of Echelon Form

In this section, we will make use of a certain constant $\mathsf{C}^{**}_{\mathsf{D}}$, depending only on the dimension D. We make the assumption that

(*) $\mathsf{C}^{**}_{\mathsf{D}}$ is a large enough constant determined by D.

Rather than specifying the value of $\mathsf{C}^{**}_{\mathsf{D}}$ here, we will assume several lower bounds for $\mathsf{C}^{**}_{\mathsf{D}}$ in this section. Those lower bounds will always be by quantities that depend solely on D. Eventually, we take $\mathsf{C}^{**}_{\mathsf{D}}$ to be a constant determined by D that satisfies those constraints.

Algorithm MALP1: *Given an* S*-bit* MALP $\mathcal{A}$ *in* $\mathbb{R}^{\mathsf{D}}$ *with constant* Υ *such that* $\mathsf{S} \geq \mathsf{C}^{**}_{\mathsf{D}}$ *and* $\bar{\mathsf{S}} \geq \mathsf{C}^{**}_{\mathsf{D}}\mathsf{S}$, *we produce an* S'*-bit* MALP $\mathcal{A}'$ *in* $\mathbb{R}^{\mathsf{D}}$ *with constant* Υ', *in echelon form, and* Υ'*-equivalent to* $\mathcal{A}$. *Here,* $\mathsf{S}' = \mathsf{C}_{\mathsf{D}}\mathsf{S}$ *for a constant* C_{D} *depending only on* D*; and* Υ' *depends only on* Υ *and* D.

Explanation: This is the content of Lemma 2 in the preceding section. (We assume that $\mathsf{C}^{**}_{\mathsf{D}} \geq \hat{\mathsf{C}}$ from Lemma 2.) The work of the algorithm is at most $\mathsf{C}_{\mathsf{D}}\cdot$ length $(\mathcal{A})$, with C_{D} depending only on D.

Algorithm MALP2: *Given an* S*-bit* MALP $\mathcal{A}$ *in* $\mathbb{R}^D$ *with constant* Υ *such that* $S \geq C_D^{**}$ *and* $\bar{S} \geq C_D^{**}S$*, we compute a number* $\widehat{M}$ *such that* $c_{\Upsilon,D}\widehat{M} \leq$ *onset* $(\mathcal{A}) \leq C_{\Upsilon,D}\widehat{M}$*, where* $c_{\Upsilon,D}$ *and* $C_{\Upsilon,D}$ *depend only on* Υ *and* D*.*

Explanation: We place $\mathcal{A}$ into echelon form using algorithm MALP1, then take $\widehat{M}$ to be the threshold of the MALP $\mathcal{A}'$ in echelon form. This works, since $\mathcal{A}'$ and $\mathcal{A}$ are Υ'-equivalent, and since the onset of a FALP in echelon form is equal to its threshold. The work of the algorithm is at most $C_D\cdot$ length $(\mathcal{A})$, where C_D depends only on the dimension D.

Algorithm MALP3: *Let* $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ *be the projection onto the last* $\bar{D}$ *coordinates. Given an* S*-bit* MALP $\mathcal{A}$ *in* $\mathbb{R}^D$ *with constant* Υ *such that* $S \geq C_D^{**}$ *and* $\bar{S} \geq C_D^{**}S$*, we produce an* S''*-bit* MALP $\mathcal{A}''$ *in* $\mathbb{R}^{\bar{D}}$ *with constant* Υ''*, with the following properties:*

- $\mathcal{K}(\mathcal{A}'')$ *is* $C_{\Upsilon,D}$*-equivalent to* $\pi\mathcal{K}(\mathcal{A})$*, where* $C_{\Upsilon,D}$ *depends only on* Υ, D*.*
- $S'' = C_D S$*, where* C_D *depends only on* D*.*
- Υ'' *depends only on* Υ *and* D*.*
- *length* $(\mathcal{A}'') = \bar{D}$*.*

Explanation: Using Algorithm MALP1, we find an S'-bit MALP $\mathcal{A}'$ in $\mathbb{R}^D$, with constant Υ', satisfying the conditions given in Algorithm MALP1. Set

$$\mathcal{A}' = \left[(\lambda'_{\ell j})_{\substack{1\leq\ell\leq D\\1\leq j\leq D}}, (b'_\ell)_{1\leq\ell\leq D}, (\sigma'_\ell)_{1\leq\ell\leq D}, M'_*\right].$$

(Recall that a MALP in echelon form must have length exactly D.) Then

$$\mathcal{A}'' = \left[(\lambda'_{\ell j})_{\substack{D-\bar{D}+1\leq\ell\leq D\\D-\bar{D}+1\leq j\leq D}}, (b'_\ell)_{D-\bar{D}+1\leq\ell\leq D}, (\sigma'_\ell)_{D-\bar{D}+1\leq\ell\leq D}, M'_*\right]$$

satisfies

$$\mathcal{K}(\mathcal{A}'') = \pi\mathcal{K}(\mathcal{A}'),$$

as we see by "backsolving" for $\nu_{D-\bar{D}}, \nu_{D-\bar{D}-1}, \dots, \nu_1$ to obtain $(\nu_1, \dots, \nu_D) \in K_M(\mathcal{A}')$ from any given $(\nu_{D-\bar{D}+1}, \dots, \nu_D) \in K_M(\mathcal{A}'')$. (Here, we use the triangular form of $(\lambda'_{\ell j})$ given by (EF1,2) from Section 42. See the corresponding argument in Section 5.)

Consequently, $\mathcal{K}(\mathcal{A}'')$ is $C_{\Upsilon,D}$-equivalent to $\pi\mathcal{K}(\mathcal{A})$. Also, length $(\mathcal{A}'') = \bar{D}$. It remains to check that $\mathcal{A}''$ is an S''-bit MALP with constant Υ'', as in the statement of the algorithm. To see this, we note that, since $\mathcal{A}'$ is an S'-bit MALP in echelon form, we have

$$\begin{aligned}&|\lambda'_{\ell j}|, |b'_\ell|, |\sigma'_\ell|, M'_* \leq 2^{S'},\\&\sigma'_\ell, M'_* \geq 2^{-S'}\end{aligned}$$

and

$$\left|\det\left((\lambda'_{\ell j})_{\substack{D-\bar{D}+1\leq\ell\leq D\\ D-\bar{D}+1\leq j\leq D}}\right)\right| \geq 2^{-\bar{D}S'},$$

thanks to the triangular form of $(\lambda'_{\ell j})$ and the estimates

$$|\lambda'_{\ell\ell}| \geq 2^{-S'}.$$

The above estimates, together with Lemma 3 in Section 40, show that $\mathcal{A}''$ is an S''-bit MALP with constant $\Upsilon'' = 2$, as claimed. Thus, $\mathcal{A}''$ does what we claimed it would do.

The work of the algorithm is at most $C_D\cdot$ length $(\mathcal{A})$, since we read off $\mathcal{A}''$ from $\mathcal{A}'$, which is produced from $\mathcal{A}$ by Algorithm MALP1.

Algorithm MALP4: *Let $S \geq 1$ be an integer such that $S \geq C_D^{**}$ and $\bar{S} \geq C_D^{**}S$. Let $T : \mathbb{R}^D \to \mathbb{R}^D$ be a linear map with $\| T \|, \| T^{-1} \| \leq 2^S$, where $\| \cdot \|$ stands, say, for the Hilbert-Schmidt norm. Suppose that " T" is a given matrix of machine numbers, whose elements differ from the corresponding elements in T by at most $2^{-\bar{S}/2}$.*

Let $\mathcal{A}$ be an S-bit MALP *in $\mathbb{R}^D$ with constant Υ. Then we produce an S'-bit* MALP *$\mathcal{A}'$ in $\mathbb{R}^D$ with constant Υ', such that $\mathcal{K}(\mathcal{A}')$ is Υ'-equivalent to $T\mathcal{K}(\mathcal{A})$. Here, $S' = C'_D S$ with C'_D depending only on D, and Υ' depends only on Υ and D. Also,* length$\,(\mathcal{A}') =$ length$\,(\mathcal{A})$.

Explanation: We write C_D, C'_D, etc. to denote constants depending only on D. Denote by "T^{-1}" the result of our attempt to compute T^{-1} in our model of computation (starting from the matrix "T"). Thus, the elements of "T^{-1}" differ from the corresponding elements of T^{-1} by roundoff errors that are at most $C_D 2^{C_D S} \cdot 2^{-\bar{S}/2}$ in absolute value. Let T^{-1} be given by the matrix $(\tau_{ij})_{\substack{1\leq i\leq D\\ 1\leq j\leq D}}$. Then if

$$\mathcal{A} = [(\lambda_{\ell j}), (b_\ell), (\sigma_\ell), M_*],$$

and if we could do arithmetic without roundoff errors, then we would set

$$\mathcal{A}' = \left[\left(\sum_i \lambda_{\ell i}\tau_{ij}\right), (b_\ell), (\sigma_\ell), M_*\right],$$

and we would have $\mathcal{K}(\mathcal{A}') = T\mathcal{K}(\mathcal{A})$. We examine the effect of roundoff errors. Since $|\lambda_{\ell i}| \leq 2^S$, $|\tau_{ij}| \leq 2^S$, it follows that the roundoff error in computing $\sum_i \lambda_{\ell i}\tau_{ij}$ is at most $C_D \cdot 2^{C_D S} \cdot 2^{-\bar{S}/2}$ in absolute value. (See Section 38.) So, in trying to compute $\mathcal{A}'$, we produce

$$(*1)\ \mathcal{A}'' = \left[\left(\sum_i \lambda_{\ell i}\tau_{ij} + \epsilon_{\ell j}\right), (b_\ell), (\sigma_\ell), M_*\right], \text{with } |\epsilon_{\ell j}| \leq C'_D \cdot 2^{C_D S} 2^{-\bar{S}/2}.$$

Let us examine also

(*2) $\mathcal{A}''' = \left[\left(\sum_i \lambda_{\ell i}\tau_{ij} + \epsilon_{\ell j} + \epsilon'''_{\ell j}\right), (b_\ell + \delta'''_\ell), (\sigma_\ell), M_*\right]$, with $|\epsilon'''_{\ell j}|, |\delta'''_\ell| \leq 2^{-3S}$.

Assume, as we may, that C^{**}_D is larger than $10(C_D + C'_D)$, and recall that $S \geq C^{**}_D$ and $\bar{S} \geq C^{**}_D S$. Under our assumptions on S and $\bar{S}$, we have $|\epsilon_{\ell j} + \epsilon'''_{\ell j}| \leq |\epsilon_{\ell j}| + |\epsilon'''_{\ell j}| \leq 2^{1-3S}$. Then,

$$\mathcal{K}(\mathcal{A}'') = T\mathcal{K}(\tilde{\mathcal{A}}) \quad \text{and} \quad \mathcal{K}(\mathcal{A}''') = T\mathcal{K}(\mathcal{A}^*),$$

where $\tilde{\mathcal{A}}$ and $\mathcal{A}^*$ are ALPs of the form

$$\tilde{\mathcal{A}} = [(\lambda_{\ell j} + \tilde{\eta}_{\ell j}), (b_\ell), (\sigma_\ell), M_*],$$
$$\mathcal{A}^* = \left[(\lambda_{\ell j} + \eta^*_{\ell j}), (b_\ell + \delta'''_\ell), (\sigma_\ell), M_*\right],$$

with

$$|\tilde{\eta}_{\ell j}| \leq C''_D 2^{C_D S} \cdot 2^{-\bar{S}/2}, \quad |\eta^*_{\ell j}| \leq C''_D \cdot 2^{-2S}, \quad |\delta'''_\ell| \leq 2^{-3S}.$$

(We used the fact that $\| T \| \leq 2^S$.) Assume, as we may, that C^{**}_D is larger than, say, $100(C''_D + C_D)$. Since $\mathcal{A}$ is an S-bit MALP with constant Υ,

(*3) It follows that $\tilde{\mathcal{A}}$ and $\mathcal{A}^*$ are both Υ-equivalent to $\mathcal{A}$; hence, they are Υ^2-equivalent to each other.

(*4) Consequently, $\mathcal{A}''$ and $\mathcal{A}'''$ are Υ^2-equivalent (to each other).

Moreover, we have

(*5) $2^{-S} \leq \sigma_\ell \leq 2^{+S}$ and $2^{-S} \leq M_* \leq 2^{+S}$, $|b_\ell| \leq 2^{+S}$

since $\mathcal{A}$ is an S-bit MALP. We have also

(*6) $\left|\sum_i \lambda_{\ell i}\tau_{ij} + \epsilon_{\ell j}\right| \leq D2^{2S} + C'_D \cdot 2^{C_D S} 2^{-\bar{S}/2} \leq 2^{C'''_D S}$.

Thus, (*1) and (*2) imply (*4); and we have (*5) and (*6). Comparing these results with the definition of an S-bit FALP, we see that $\mathcal{A}''$ is a $C_D S$-bit FALP with constant Υ^2. Since $\mathcal{A}''$ arises from a machine computation, its entries are machine numbers. Thus,

(*7) $\mathcal{A}''$ is a $C_D S$-bit MALP with constant Υ^2.

On the other hand, from (*3), we see that

(*8) $\mathcal{K}(\mathcal{A}'')$ is Υ-equivalent to $T\mathcal{K}(\mathcal{A})$.

Since also length $(\mathcal{A}'')$ = length $(\mathcal{A})$, we obtain the conclusions asserted in Algorithm MALP4, with $\Upsilon' = \Upsilon^2, S' = C_D S$. The work of the algorithm is at most $C_D\cdot$ length $(\mathcal{A})$.

Algorithm MALP5: *Let* $\mathsf{D} = \mathsf{D}_1 + \mathsf{D}_2$. *Given* $\mathcal{A}^i$ *an* S*-bit* MALP *in* $\mathbb{R}^{\mathsf{D}_i}$ *with constant* Υ *such that* $\mathsf{S} \geq \mathsf{C}^{**}_{\mathsf{D}}$ *and* $\bar{\mathsf{S}} \geq \mathsf{C}^{**}_{\mathsf{D}}\mathsf{S}$, *for* $i = 1, 2$*; we produce an* S'*-bit* MALP $\tilde{\mathcal{A}}$ *with constant* Υ' *in* $\mathbb{R}^{\mathsf{D}_1+\mathsf{D}_2}$, *with length* $(\tilde{\mathcal{A}}) = \mathsf{D}_1 + \mathsf{D}_2$, *such that* $\mathcal{K}(\tilde{\mathcal{A}})$ *is* Υ'*-equivalent to* $\mathcal{K}(\mathcal{A}^1) \times \mathcal{K}(\mathcal{A}^2)$. *Here,* $\mathsf{S}' = \mathsf{C}_{\mathsf{D}_1,\mathsf{D}_2}\mathsf{S}$ *with* $\mathsf{C}_{\mathsf{D}_1,\mathsf{D}_2}$ *depending only on* D_1 *and* D_2*; and* Υ' *depends only on* Υ, $\mathsf{D}_1, \mathsf{D}_2$.

Explanation: We write $\mathsf{C}, \mathsf{C}', \ldots$ for constants depending only on $\mathsf{D}_1, \mathsf{D}_2$; and we write Υ', Υ'', to denote constants depending only on $\Upsilon, \mathsf{D}_1, \mathsf{D}_2$. Using Algorithm MALP1, we first place $\mathcal{A}^i$ in echelon form in $\mathbb{R}^{\mathsf{D}_i}$. Let $\tilde{\mathcal{A}}^i$ be an S'-bit MALP in echelon form in $\mathbb{R}^{\mathsf{D}_i}$, with constant Υ', and with $\tilde{\mathcal{A}}^i$ being Υ'-equivalent to $\mathcal{A}^i$ ($\mathsf{S}' = \mathsf{C}\mathsf{S}$). It is enough to carry out Algorithm MALP5 with $\mathcal{A}^i$ replaced by $\tilde{\mathcal{A}}^i$. In an obvious way, we produce an ALP $\tilde{\mathcal{A}}$ in $\mathbb{R}^{\mathsf{D}_1+\mathsf{D}_2}$, such that $\mathcal{K}(\tilde{\mathcal{A}}) = \mathcal{K}(\tilde{\mathcal{A}}^1) \times \mathcal{K}(\tilde{\mathcal{A}}^2)$. We must show that $\tilde{\mathcal{A}}$ is an S''-bit MALP with constant Υ''. (Here, $\mathsf{S}'' = \mathsf{C}'\mathsf{S}$.) To see this, we first note that $\tilde{\mathcal{A}}$ satisfies (DS2 ,..., 6) in the definition of FALPs, with S replaced by S', simply because $\tilde{\mathcal{A}}^1$, $\tilde{\mathcal{A}}^2$ are S'-bit MALPs. Also, the entries of $\tilde{\mathcal{A}}$ are machine numbers, since $\tilde{\mathcal{A}}^1$, $\tilde{\mathcal{A}}^2$ are MALPs. It remains to show that (DS7) holds for suitable S'', Υ''. To see this, we use the fact that $\tilde{\mathcal{A}}^1, \tilde{\mathcal{A}}^2$ are in echelon form. In particular, we have

$$\tilde{\mathcal{A}}^i = \left[\left(\tilde{\lambda}^i_{\ell j}\right)_{\substack{1 \leq \ell \leq \mathsf{D}_i \\ 1 \leq j \leq \mathsf{D}_i}}, \, (\tilde{b}^i_\ell)_{1 \leq \ell \leq \mathsf{D}_i}, \, (\tilde{\sigma}^i_\ell)_{1 \leq \ell \leq \mathsf{D}_i}, \tilde{M}^i_* \right] \quad \text{for } i = 1, 2,$$

with

$$\tilde{\Lambda}^i = \left(\tilde{\lambda}^i_{\ell j}\right) = \begin{pmatrix} \tilde{\lambda}^i_{11} & & * \\ & \ddots & \\ 0 & & \tilde{\lambda}^i_{\mathsf{D}_i\mathsf{D}_i} \end{pmatrix},$$

where the diagonal entries $\tilde{\lambda}^i_{\ell\ell}$ have absolute value at least $2^{-\mathsf{S}'}$.

For the "direct sum"

$$\tilde{\mathcal{A}} = [(\tilde{\lambda}_{\ell j}), (\tilde{b}_\ell), (\tilde{\sigma}_\ell), \tilde{M}_*],$$

the matrix $(\tilde{\lambda}_{\ell j})$ has the form

$$(\tilde{\lambda}_{\ell j}) = \begin{pmatrix} \tilde{\Lambda}^1 & 0 \\ 0 & \tilde{\Lambda}^2 \end{pmatrix}.$$

Hence, $|\det(\tilde{\lambda}_{\ell j})| = |\det \tilde{\Lambda}^1| \cdot |\det \tilde{\Lambda}^2| \geq 2^{-(\mathsf{D}_1+\mathsf{D}_2)\mathsf{S}'}$. We know that $|\tilde{\lambda}_{\ell j}|$, $|\tilde{b}_\ell|$, $\tilde{\sigma}_\ell$, $\tilde{M}_* \leq 2^{\mathsf{S}'}$ and that $\tilde{\sigma}_\ell$, $\tilde{M}_* \geq 2^{-\mathsf{S}'}$.

It now follows from Lemma 3 in Section 40 that $\tilde{\mathcal{A}}$ is a $\mathsf{C}\mathsf{S}'$-bit FALP with constant 2. Hence, our $\tilde{\mathcal{A}}$ has all the properties asserted in Algorithm MALP5. (Clearly, length $(\tilde{\mathcal{A}}) = \mathsf{D}_1+\mathsf{D}_2$.) The work of the algorithm is at most $\mathsf{C} \cdot$(length($\mathcal{A}^1$)+ length($\mathcal{A}^2$)), with C depending only on D_1 and D_2.

Algorithm MALP6: *Given two* S*-bit* MALPs $\mathcal{A}, \mathcal{B}$ *in* $\mathbb{R}^D$ *with constant* Υ *such that* $S \geq C^{**}_D$ *and* $\bar{S} \geq (C^{**}_D)^2 S$*, we compute an* S'*-bit* MALP $\mathcal{C}$ *in* $\mathbb{R}^D$ *with constant* Υ'*, such that* length$(\mathcal{C}) = D$*, and* $\mathcal{K}(\mathcal{C})$ *is* Υ'*-equivalent to* $\mathcal{K}(\mathcal{A}) + \mathcal{K}(\mathcal{B})$*. Here,* $S' = CS$ *with* C *depending only on* D*; and* Υ' *is determined by* Υ *and* D*.*

Explanation: Write C, C', etc. for constants depending only on D; and $\Upsilon', \Upsilon'', \dots$ for constants depending only on Υ, D. Using Algorithm MALP5, we produce a CS-bit MALP $\mathcal{C}^+$ in $\mathbb{R}^{2D}$, with constant Υ', such that

(†1) $\mathcal{K}(\mathcal{C}^+)$ is Υ'-equivalent to $\mathcal{K}(\mathcal{A}) \times \mathcal{K}(\mathcal{B})$.

Next, let $T : \mathbb{R}^D \oplus \mathbb{R}^D \to \mathbb{R}^D \oplus \mathbb{R}^D$ be defined by

(†2) $T(v, w) = (v, v + w)$.

Recall that $\mathcal{C}^+$ is a CS-bit MALP with constant Υ'. Then $\bar{S} \geq C^{**}_D \cdot CS$ and $CS \geq C^{**}_D$ under the legitimate assumption that $C^{**}_D \geq C$. Using Algorithm MALP4, we produce a $C'S$-bit MALP $\mathcal{C}^{++}$ in $\mathbb{R}^{2D}$ with constant Υ'', such that

(†3) $\mathcal{K}(\mathcal{C}^{++})$ is Υ''-equivalent to $T(\mathcal{K}(\mathcal{C}^+))$.

Finally, let $\pi : \mathbb{R}^D \oplus \mathbb{R}^D \to \mathbb{R}^D$ be the projection $(v, w) \mapsto w$. Since $C^{**}_D \geq C'$, then $\bar{S} \geq C^{**}_D \cdot C'S$, and we may apply Algorithm MALP3 for $\mathcal{C}^{++}$. Using Algorithm MALP3, we produce a $C''S$-bit MALP $\mathcal{C}$ in $\mathbb{R}^D$, with constant Υ''', such that

(†4) $\mathcal{K}(\mathcal{C})$ is Υ'''-equivalent to $\pi\mathcal{K}(\mathcal{C}^{++})$.

Then (†1),..., (†4) show that $\mathcal{K}(\mathcal{C})$ is Υ''''-equivalent to $\mathcal{K}(\mathcal{A}) + \mathcal{K}(\mathcal{B})$.

Also, length $(\mathcal{C}) = D$, since $\mathcal{C}$ arises by applying Algorithm MALP3. Thus, $\mathcal{C}$ has the desired properties.

The work of the algorithm is at most $C_D\cdot$ (length$(\mathcal{A})$ + length $(\mathcal{B})$), where C_D depends only on D.

Not all of our ALP algorithms go over to MALPs. In particular, if $\mathcal{A}$ is an S-bit MALP with constant Υ in $\mathbb{R}^D$, and if $T : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ is the injection $(v_1, \dots, v_D) \mapsto (v_1, \dots, v_D, 0, \dots, 0)$, with $\bar{D} > D$, then $T\mathcal{K}(\mathcal{A})$ has the form $\mathcal{K}(\mathcal{A}^+)$ with the matrix $(\lambda_{\ell j})$ of the ALP $\mathcal{A}^+$ having rank $< \bar{D}$; consequently, by Lemma 2 from Section 40, $\mathcal{A}^+$ is not an S'-bit MALP for any S'.

Algorithm MALP7: *Suppose we are given* S*-bit* MALPs $\mathcal{A}^i$ *with constant* Υ *in* $\mathbb{R}^D$*, for* $i = 1, \dots, T$*. We compute an* S*-bit* MALP $\mathcal{A}$ *with constant* Υ *in* $\mathbb{R}^D$*, such that* $\mathcal{K}(\mathcal{A}) = \mathcal{K}(\mathcal{A}^1) \cap \cdots \cap \mathcal{K}(\mathcal{A}^T)$*, and* length$(\mathcal{A})$ = length(A^1) $+ \cdots +$ length$(\mathcal{A}^T)$.

Explanation: We concatenate the λ's, b's, σ's arising from the $\mathcal{A}^i$ in a obvious way, and we take the maximum of the M_*'s arising from the $\mathcal{A}^i$. Thus, we form $\mathcal{A}$. Since the $\mathcal{A}^i$ are S-bit MALPs with constant Υ, so is $\mathcal{A}$. (That's trivial from the definition of FALPs and MALPs.) Also, one sees at once that $\mathcal{K}(\mathcal{A})$ and length $(\mathcal{A})$ are as claimed . The work of the algorithm is at most $C_D \cdot (\text{length}(\mathcal{A}^1) + \cdots + \text{length}(\mathcal{A}^T))$, with C_D depending only on D.

Algorithm MALP8: *Let* $\pi : \mathbb{R}^{D_1} \oplus \mathbb{R}^{D_2} \longrightarrow \mathbb{R}^{D_2}$ *be the projection*

$$(v_1, \ldots, v_{D_1}, w_1, \ldots, w_{D_2}) \mapsto (w_1, \ldots, w_{D_2}).$$

We are given an S*-bit* MALP $\mathcal{A}$ *in* $\mathbb{R}^{D_1} \oplus \mathbb{R}^{D_2}$ *with constant* Υ *such that* $S \geq C^{**}_{D_1+D_2}$ *and* $\bar{S} \geq C^{**}_{D_1+D_2} \cdot S$, *and a vector* $(\bar{w}_1, \ldots, \bar{w}_{D_2}) \in \mathbb{R}^{D_2}$ *where* $|\bar{w}_i| \leq 2^S$ *for each* i, *and each* $\bar{w}_i$ *is a machine number. Then we produce* $(\bar{v}_1, \ldots, \bar{v}_{D_1}) \in \mathbb{R}^{D_1}$, *with each* $\bar{v}_i$ *a machine number of absolute value* $\leq 2^{CS}$, *and having the following property*:

Suppose $(v_1, \ldots, v_{D_1}) \in \mathbb{R}^{D_1}$ *and* $M > 0$ *satisfy*

(#1) $(v_1, \ldots, v_{D_1}, \bar{w}_1, \ldots, \bar{w}_{D_2}) \in K_M(\mathcal{A})$.

Then

(#2) $(\bar{v}_1, \ldots, \bar{v}_{D_1}, \bar{w}_1, \ldots, \bar{w}_{D_2}) \in K_{\Upsilon^* M}(\mathcal{A})$,
where Υ^* *depends only on* Υ, D_1, D_2.

Here, C *depends only on* D_1 *and* D_2.

Explanation: We write C, C', etc., for constants depending only on D_1, D_2; and we write Υ', Υ'', ... for constants depending only on Υ, D_1, D_2. Using Algorithm MALP1, we first produce from $\mathcal{A}$ a CS-bit MALP $\mathcal{A}'$ with constant Υ' in $\mathbb{R}^{D_1} \oplus \mathbb{R}^{D_2}$, such that $\mathcal{A}'$ is in echelon form, and $\mathcal{A}'$ is Υ'-equivalent to $\mathcal{A}$.

Then, in (#1) and (#2), we may replace $\mathcal{A}$ by $\mathcal{A}'$. From now on, we suppose this has been done. We write vectors in $\mathbb{R}^{D_1} \oplus \mathbb{R}^{D_2}$ as

$$(v_1, \ldots, v_{D_1}, v_{D_1+1}, \ldots, v_{D_1+D_2}),$$

and we suppose $\mathcal{A}' = [(\lambda_{\ell j}), (b_\ell), (\sigma_\ell), M_*]$, where ℓ and j vary from 1 to $D_1 + D_2$. Since $\mathcal{A}'$ is in echelon form, we have (with $S' = CS$):

(#3) $|\lambda_{\ell j}|$, $|b_\ell|$, σ_ℓ, $M_* \leq 2^{S'}$;

(#4) $|\lambda_{\ell\ell}|$, σ_ℓ, $M_* \geq 2^{-S'}$;

(#5) $\lambda_{\ell j} = 0$ for $j < \ell$.

Given $(\bar{w}_1, \dots, \bar{w}_{D_2}) = (\bar{v}_{D_1+1}, \dots, \bar{v}_{D_1+D_2}) \in \mathbb{R}^{D_2}$, we attempt to compute successively $\bar{v}_{D_1}, \bar{v}_{D_1-1}, \dots, \bar{v}_1$ by "backsolving" the equations

(#6) $\sum_j \lambda_{\ell j} \bar{v}_j = b_\ell$ for $\ell = D_1, D_1 - 1, \dots, 1$

using our model of computation. Because of round-off errors, the $(\bar{v}_1, \dots, \bar{v}_D)$ we compute does not exactly solve (#6). However, thanks to (#3), (#4) and (#5), and our assumptions on the $\bar{w}_i$ we obtain

(#7) $|\bar{v}_j| \leq 2^{CS'}$ for $j = 1, \dots, D_1$

and

(#8) $\left|\sum_j \lambda_{\ell j} \bar{v}_j - b_\ell\right| \leq C2^{CS'}2^{-\bar{S}}$ for $\ell = 1, \dots, D_1$.

Let $\mathcal{A}'' = [(\lambda_{\ell j}), (b''_\ell), (\sigma_\ell), M_*]$, with

$$b''_\ell = b_\ell \quad \text{for } \ell > D_1, \quad \text{and}$$
$$b''_\ell = \sum_j \lambda_{\ell j}\bar{v}_j \text{ for } \ell \leq D_1 .$$

Then

(#9) $|b''_\ell - b_\ell| \leq C2^{CS'} \cdot 2^{-\bar{S}}$ for all ℓ,

thanks to (#8).

Hence, because $\mathcal{A}'$ is an S'-bit MALP with constant Υ', $\bar{S} \geq (C^{**}_{D_1+D_2}/C')S'$, and since $C^{**}_{D_1+D_2}$ is a large enough constant, we have that

(#10) $\mathcal{A}''$ is Υ'-equivalent to $\mathcal{A}'$.

Note that $\mathcal{A}''$ depends on $(\bar{w}_1, \dots, \bar{w}_{D_2})$. That won't matter.

Now suppose $(v_1, \dots, v_{D_1}) \in \mathbb{R}^{D_1}$ and $M > 0$ satisfy

(#11) $(v_1, \dots, v_{D_1}, \bar{v}_{D_1+1}, \dots, \bar{v}_{D_1+D_2}) \in K_M(\mathcal{A}')$.

Then (#10) gives

(#12) $(v_1, \dots, v_{D_1}, \bar{v}_{D_1+1}, \dots, \bar{v}_{D_1+D_2}) \in K_{\Upsilon' M}(\mathcal{A}'')$.

In particular, for $\ell > D_1$, we have

(#13) $\left|\sum_{j \geq \ell} \lambda_{\ell j} \bar{v}_j - b''_\ell\right| \leq \Upsilon' M \, \sigma_\ell$.

(Here, we use (#5).) Also, (#12) gives

(#14) $\Upsilon' M \geq M_*$.

Note that (#13) holds also for $\ell \leq D_1$, since (by definition of the b''_ℓ) the left-hand side of (#13) is zero. Hence, (#13) and (#14) yield

$$(\bar{v}_1, \dots, \bar{v}_{D_1}, \bar{v}_{D_1+1}, \dots, \bar{v}_{D_1+D_2}) \in K_{\Upsilon' M}(\mathcal{A}'') .$$

(Again, we use (#5).) By (#10), it now follows that

$$\text{(\#15)} \quad (\bar{v}_1, \dots, \bar{v}_{D_1}, \bar{v}_{D_1+1}, \dots, \bar{v}_{D_1+D_2}) \in K_{(\Upsilon')^2 M}(\mathcal{A}').$$

Thus, (#11) implies (#15), which shows that our $(\bar{v}_1, \dots, \bar{v}_{D_1})$ has the desired property.

The work of the algorithm is at most $C\cdot$ length $(\mathcal{A})$, with C depending only on D_1 and D_2. (That's the work of Algorithm MALP1; once we obtain $\mathcal{A}'$, then the "backsolving" takes work at most C, a constant depending only on D_1 and D_2.)

Remark. Algorithm MALP8 replaces our discussion of original vectors for ALPs.

Algorithm MALP9: *Let* $D, S, \Upsilon \geq 1$ *be given. We write* c, C, C', *etc., to denote constants depending only on* D, *and we write* Υ', Υ^*, *etc., to denote constants depending only on* Υ *and* D.

Let $T : \mathbb{R}^D \to \mathbb{R}^D$ *be a linear map, with* $\| T \|, \| T^{-1} \| \leq 2^S$.

Suppose we are given the matrix "T*", whose elements are machine numbers that differ from the corresponding elements of* T *by at most* $2^{-\bar{S}/2}$.

Let $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ *be the projection onto the last* $\bar{D}$ *coordinates.*

Let $\mathcal{A}$ *be an* S*-bit* MALP *in* $\mathbb{R}^D$, *with constant* Υ *and length* D. *Assume that* $S \geq C^{**}_D$ *and* $\bar{S} \geq (C^{**}_D)^2 S$.

Let $v^0 = (v^0_1, \dots, v^0_D) \in \mathbb{R}^D$, *where the* v^0_i *are machine numbers, with* $|v^0_i| \leq 2^S$.

From the above data, we compute a vector $v^1 = (v^1_1, \dots, v^1_D) \in \mathbb{R}^D$, *with the following properties:*

(a) $|v^1_i| \leq 2^{C'S}$ *for* $i = 1, \dots, D$.

(b) *Let* $w \in \mathbb{R}^D$ *and* $M > 0$. *Assume that*

(i) $\pi T(w - v^0) = 0$ *and* $w \in K_M(\mathcal{A})$.

Then we can express v^1 *as a sum*

(ii) $v^1 = v^2 + v^3$, *where*

(iii) $\pi T(v^2 - v^0) = 0$ *and* $v^2 \in K_{\Upsilon^* M}(\mathcal{A})$, *and*

(iv) $|v^3| \leq 2^{C'S} \cdot 2^{-\bar{S}/2}$.

Moreover, if (i) *holds, then*

(v) $v^1 \in K_{\Upsilon^* M}(\mathcal{A})$.

Explanation: Note that we will not compute v^2, v^3 above; we merely assert that they exist. Their components need not be machine numbers.

We explain how to compute v^1 having the desired properties. Let $\bar{v}^0 = Tv^0$, and let $\bar{\bar{v}}^0$ be our machine approximation to $\bar{v}^0$. We can compute $\bar{\bar{v}}^0$, and we have $\bar{v}^0 = (\bar{v}^0_1, \dots, \bar{v}^0_D)$, $\bar{\bar{v}} = (\bar{\bar{v}}^0_1, \dots, \bar{\bar{v}}^0_D)$, with $|\bar{\bar{v}}^0_i| \leq 2^{CS}$ and $|\bar{\bar{v}}^0_i - \bar{v}^0_i| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$, for $i = 1, \dots, D$. Applying Algorithm MALP4 we obtain a CS-bit MALP $\bar{\mathcal{A}}$ with constant Υ' and length $\leq D$, such that $T\mathcal{K}(\mathcal{A})$ and $\mathcal{K}(\bar{\mathcal{A}})$ are Υ'-equivalent.

We now apply Algorithm MALP8 to the vector $\pi\bar{\bar{v}}^0$ and the MALP $\bar{\mathcal{A}}$. (This is allowed since $\bar{S} \geq C^{**}_D \cdot CS$.) Thus, we compute a vector

$$\bar{v}^1 = (\bar{v}^1_1, \dots, \bar{v}^1_D) \in \mathbb{R}^D,$$

such that $\bar{v}^1_i$ is a machine number and $|\bar{v}^1_i| \leq 2^{CS}$ for each $i = 1, \dots, D$, and having the following property:

(†) Let $\bar{v} \in \mathbb{R}^D$ and $M > 0$ satisfy $\pi\bar{v} = \pi\bar{\bar{v}}^0$ and $\bar{v} \in K_M(\bar{\mathcal{A}})$.

Then $\pi\bar{v}^1 = \pi\bar{\bar{v}}^0$ and $\bar{v}^1 \in K_{\Upsilon^* M}(\bar{\mathcal{A}})$.

By our assumptions on the matrix "T", we may compute a matrix "T^{-1}" of machine numbers, that differ from the actual elements of T^{-1} by at most $2^{CS} \cdot 2^{-\bar{S}/2}$. Finally, let $\hat{v}^1 = T^{-1}\bar{v}^1$, and let v^1 be our machine approximation to $\hat{v}^1$. Thus, $|\hat{v}^1 - v^1| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$, hence $|\bar{v}^1 - Tv^1| \leq 2^{C'S} \cdot 2^{-\bar{S}/2}$. We have computed the vector v^1. We will show that it has the desired properties. To see that v^1 satisfies (a), we just recall that $|\bar{v}^1| \leq 2^{CS}$, hence $|\hat{v}^1| \leq \| T^{-1} \| \, |\bar{v}^1| \leq 2^{C'S}$, hence $|v^1| \leq 2^{C''S}$. This proves (a).

To see that v^1 satisfies (b), let $w \in \mathbb{R}^D$ and $M > 0$ satisfy $\pi T(w - v^0) = 0$ and $w \in K_M(\mathcal{A})$. Then

$$|\pi(Tw) - \pi\bar{\bar{v}}^0| = |\pi T(w - v^0) + \pi(\bar{v}^0 - \bar{\bar{v}}^0)| = |\pi(\bar{v}^0 - \bar{\bar{v}}^0)| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$$

since $\bar{v}^0 = Tv^0$ and $|\bar{v}^0_i - \bar{\bar{v}}^0_i| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$. Hence, there exists $u \in \mathbb{R}^D$, with $\pi(Tw + u) - \pi\bar{\bar{v}}^0 = 0$, and $|u| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$. Since $w \in K_M(\mathcal{A})$, we have $Tw \in K_{\Upsilon' M}(\bar{\mathcal{A}})$. To summarize, $\bar{\mathcal{A}}$ is a CS-bit MALP with constant Υ', and $Tw \in K_{\Upsilon' M}(\bar{\mathcal{A}})$, $|u| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$. Hence, by Lemma 4 from Section 40, we have $(Tw + u) \in K_{\Upsilon'' M}(\bar{\mathcal{A}})$. Thus,

$$(Tw + u) \in K_{\Upsilon'' M}(\bar{\mathcal{A}}) \quad \text{and} \quad \pi(Tw + u) = \pi\bar{\bar{v}}^0.$$

Consequently, (†) yields the following: $\pi\bar{v}^1 = \pi\bar{\bar{v}}^0$, and $\bar{v}^1 \in K_{\Upsilon''' M}(\bar{\mathcal{A}})$. In particular,

(∗1) $\hat{v}^1 = T^{-1}\bar{v}^1 \in K_{\Upsilon^{\#} M}(\mathcal{A})$, since $\mathsf{T}\mathcal{K}(\mathcal{A})$ is Υ'-equivalent to $\mathcal{K}(\bar{\mathcal{A}})$. Also,

$$\pi\mathsf{T}\hat{v}^1 = \pi\bar{v}^1 = \pi\bar{\bar{v}}^0.$$

Let us set $\bar{u} = \bar{v}^0 - \bar{\bar{v}}^0$ and let $\hat{u} = \mathsf{T}^{-1}\bar{u}$. Then $|\bar{u}| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$ and hence $|\hat{u}| \leq \| \mathsf{T}^{-1} \| \cdot |\bar{u}| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$. In addition,

$$(*2) \qquad \pi\mathsf{T}(\hat{v}^1 + \hat{u}) = \pi\bar{\bar{v}}^0 + \pi\bar{u} = \pi\bar{v}^0 = \pi\mathsf{T}(v^0).$$

Since $\hat{v}^1 \in \mathsf{K}_{\Upsilon\# M}(\mathcal{A})$ by (∗1) and $|\hat{u}| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$; and since $\mathcal{A}$ is an S-bit MALP with constant Υ, it follows from Lemma 4 in Section 40 that

$$(*3) \qquad \hat{v}^1 + \hat{u} \in \mathsf{K}_{\Upsilon^* M}(\mathcal{A}).$$

We set $v^2 = \hat{v}^1 + \hat{u}$, and $v^3 = v^1 - v^2$.

Thus, the desired properties (ii), (iii), are immediate from (∗2), (∗3), and the definitions of v^2, v^3.

It remains to check properties (iv) and (v). We have

$$v^3 = v^1 - v^2 = (v^1 - \hat{v}^1) - \hat{u}.$$

Since we have already seen that

$$|v^1 - \hat{v}^1| \leq 2^{CS} \cdot 2^{-\bar{S}/2} \quad \text{and} \quad |\hat{u}| \leq 2^{CS} \cdot 2^{-\bar{S}/2},$$

we obtain (iv).

To check property (v), we note that, as we have already shown, $\hat{v}^1 \in \mathsf{K}_{\Upsilon\# M}(\mathcal{A})$ by (∗1), $|v^1 - \hat{v}^1| \leq 2^{CS} \cdot 2^{-\bar{S}/2}$, and $\mathcal{A}$ is an S-bit MALP with constant Υ. Applying Lemma 4 from Section 40, we conclude that $v^1 \in \mathsf{K}_{\Upsilon^* M}(\mathcal{A})$. That is, (v) holds.
Thus, (ii),...,(v) hold, completing the proof of (b).

Consequently, the vector v^1 has the properties asserted in the statement of Algorithm MALP9.

The work of the algorithm is at most a constant determined by D.

Remark. Algorithm MALP9 will allow a version of Find-Neighbor in our model of computation. To carry this out, we make the following definitions.

Recall that $\mathcal{P}$ is the vector space of $(m-1)^{\text{rst}}$ degree polynomials on $\mathbb{R}^n$, and $D = \dim \mathcal{P}$. We identify $\mathcal{P}$ with $\mathbb{R}^D$, by identifying $P \in \mathcal{P}$ with $(\partial^\alpha P(0))_{|\alpha| \leq m-1}$. In particular, any ALP $\mathcal{A}$ in $\mathbb{R}^D$ gives rise to a blob $\mathcal{K}(\mathcal{A})$ in $\mathcal{P}$, via this identification.

We say that $P \in \mathcal{P}$ is a "machine polynomial" if $\partial^\alpha P(0)$ is a machine number for each α (with $|\alpha| \leq m-1$). To specify a machine polynomial P is to specify $\partial^\alpha P(0)$ for $|\alpha| \leq m-1$.

Next, let $x \in \mathbb{R}^n$ be given, and let $\mathcal{A}$ be a given set of multi-indices α of order $|\alpha| \le m-1$. In a moment, we will apply Algorithm MALP9, taking $T : \mathbb{R}^D \to \mathbb{R}^D$ to be the map that sends $(\partial^\alpha P(0))_{|\alpha|\le m-1}$ to $(\partial^\alpha P(x))_{|\alpha|\le m-1}$, for $P \in \mathcal{P}$. When $x = (x_1, ..., x_n)$ with x_i being a machine number of absolute value $\le 2^S$, it is straightforward to compute a matrix "T" that satisfies the requirements of Algorithm MALP9, provided that $\bar{S}/S$ is sufficiently large.

Also, $\bar{D} = \#(\mathcal{A})$ and we take $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ to be the projection

$$(\partial^\alpha P(x))_{|\alpha|\le m-1} \mapsto (\partial^\alpha P(x))_{\alpha \in \mathcal{A}}\,.$$

Note that $\pi = \pi_{\mathcal{A},x}$ where $\pi_{\mathcal{A},x}$ was defined in Section 30.

From Algorithm MALP9, we obtain the following algorithm.

Algorithm MALP10: *Suppose we are given the following data:*

- *An* S*-bit* MALP $\mathcal{A}'$ *in* $\mathcal{P}$*, with constant* Υ *and length* D*, such that* $S \ge C^{**}_D$ *and* $\bar{S} \ge C^{**}_D S$.
- *A point* $x = (x_1, \dots, x_n) \in \mathbb{R}^n$*, where each* x_i *is a machine number of absolute value at most* 2^S.
- *A set* $\mathcal{A}$ *of multi-indices of order at most* $m-1$.
- *A machine polynomial* P_0*, with* $|\partial^\alpha P_0(0)| \le 2^S$ *for* $|\alpha| \le m-1$.

Then we compute a machine polynomial P_1*, with the following properties:*

(a) $|\partial^\alpha P_1(0)| \le 2^{CS}$ *for* $|\alpha| \le m-1$.

(b) *Suppose* $P \in \mathcal{P}$ *and* $M \in (0,\infty)$ *satisfy*

$$\partial^\alpha(P-P_0)\,(x) \;=\; 0 \text{ for } \alpha \in \mathcal{A}\,,\ \text{and } P \in K_M(\mathcal{A}')\,.$$

Then we can express $P_1 = P_{main} + P_{err}$*, with*

$$\partial^\alpha(P_{main} - P_0)(x) = 0 \text{ for } \alpha \in \mathcal{A}\,,\ \text{and } P_{main} \in K_{\Upsilon^* M}(\mathcal{A}')\,;$$

and

$$|\partial^\alpha P_{err}(0)| \le 2^{CS} \cdot 2^{-\bar{S}/2} \text{ for } |\alpha| \le m-1\,.$$

Moreover, $P_1 \in K_{\Upsilon^* M}(\mathcal{A}')$. *Here,* C *depends only on* m *and* n*; and* Υ^* *depends only on* $\Upsilon, m,$ *and* n.

The work of the algorithm is less than a constant depending only on m and n.

44. Some Low-Level Algorithms

Low-Level Algorithm 0: *Let* $S \geq 1$ *be an integer, let* $a \in [2^{-S}, 2^{+S}]$ *be a machine number, and let* p *be a non-zero integer, with* $|p| \leq d$.
Assume $\bar{S} \geq CS$ *for large enough* C *depending only on* d. *Then we produce a positive machine number* t *such that* $c' < at^p < C'$, *with* c', C' *depending only on* d.

Explanation: In our model of computation, it takes one unit of work to produce an integer ℓ such that $2^\ell \leq a \leq 2^{\ell+1}$; and we have $|\ell| \leq S+1$. We can just set $t = 2^{-\lfloor \ell/p \rfloor}$. The work of the algorithm is at most 100.

Low-Level Algorithm 1: *Let* $S \geq 1$ *be an integer, let* $a, a' \in [2^{-S}, 2^{+S}]$ *be machine numbers, and let* k, k' *be distinct integers satisfying* $0 \leq k, k' \leq d$.

Assume $\bar{S} \geq CS$ *for large enough* C *depending only on* d. *Then we partition* $[2^{-S}, 2^{+S}]$ *into three intervals* I, I', J *(any of which may be empty), with the following properties:*

- *The endpoints of any non-empty interval* I, I' *or* J *are machine numbers.*
- $a't^{k'} \leq \frac{1}{200d}\, at^k$ *for all* $t \in I$,
- $at^k \leq \frac{1}{200d}\, a't^{k'}$ *for all* $t \in I'$,
- $\displaystyle\int_J \frac{dt}{t} \leq C'$, *with* C' *depending only on* d.

Explanation: Using Low-Level Algorithm 0, we can find t_0 such that $c \cdot (a't_0^{k'}) < (at_0^k) < C \cdot (a't_0^{k'})$ with c, C depending only on d. (The round-off error in computing a/a' doesn't hurt.)

We take our three intervals to be $[t_0/2^{\bar{C}}, t_0 \cdot 2^{\bar{C}}] \cap [2^{-S}, 2^S]$, and the components of $[2^{-S}, 2^S] \setminus [t_0/2^{\bar{C}}, t_0 \cdot 2^{\bar{C}}]$ for a large enough integer constant $\bar{C}$ depending only on d. It is easy to see that these intervals have the required properties, and that the round-off errors arising in calculating $t_0/2^{\bar{C}}$, $t_0 \cdot 2^{\bar{C}}$ don't hurt. The work of the algorithm is bounded by a universal constant.

Low-Level Algorithm 2: *Let* $S \geq 1$ *be an integer. Suppose we are given a machine number* $a \in [2^{-S}, 2^{+S}]$, *an integer* p *with* $|p| \leq d$, *and an interval* $I = [t_{\ell o}, t_{hi}] \subseteq [2^{-S}, 2^{+S}]$, *where* $t_{\ell o}$ *and* t_{hi} *are machine numbers.*
Assume $\bar{S} \geq CS$, *where* C *is a large enough constant depending only on* d. *Then we produce one of the following three outcomes:*

$(\mathcal{O}1)$ *We guarantee that* $at^p \geq c'$ *for all* $t \in I$.

$(\mathcal{O}2)$ *We guarantee that* $at^p \leq C'$ *for all* $t \in I$.

$(\mathcal{O}3)$ *We produce a machine number* $t_0 \in I$ *such that* $c' \leq at_0^p \leq C'$.

Here, c' *and* C' *depend only on* d.

Explanation: If $p = 0$, then by examining a we can trivially produce outcome $(\mathcal{O}1)$ or $(\mathcal{O}2)$. Otherwise, we use Low-Level Algorithm 0 to produce a machine number t_0, with $c' < at_0^p < C'$. If $t_0 \in I$, then we have produced $(\mathcal{O}3)$. If $t_0 \notin I$, then we can trivially produce outcome $(\mathcal{O}1)$ or $(\mathcal{O}2)$, thanks to the monotonicity of $t \mapsto at^p$ on $(0, \infty)$. The work of the algorithm is bounded by a universal constant.

Let $\rho(t) > 0$ be a function on an interval $I \subset (0, \infty)$, and let $C > 0$ be a constant. We say that $\rho(t)$ is "C-stable" on I if, for any $t_1, t_2 \in I$, $t_1 \leq t_2 \leq 2t_1$ implies $C^{-1}\rho(t_1) \leq \rho(t_2) \leq C\rho(t_1)$.

Low-Level Algorithm 3: *Let $S \geq 1$ be an integer. Suppose we are given a partition of $[2^{-S}, 2^{S}]$ into intervals $I_1, \dots, I_{\nu_{\max}}$, whose endpoints are machine numbers. For each $\nu = 1, \dots, \nu_{\max}$, suppose we are given a machine number $a_\nu \in [2^{-\bar{S}}, 2^{+\bar{S}}]$ and an integer p_ν, with $|p_\nu| \leq d$. Define $\rho(t)$ on $[2^{-S}, 2^{+S}]$ by setting $\rho(t) = a_\nu t^{p_\nu}$ on I_ν, $\nu = 1, \dots, \nu_{\max}$.*
Assume that $\rho(t)$ is C_1-stable on $[2^{-S}, 2^{S}]$. (In particular, $\rho(t) > 0$ on $[2^{-S}, 2^{S}]$.)
Assume also that $\bar{S} \geq C_2 S$, for a large enough C_2 depending only on C_1 and d. Then we produce one of the following three outcomes.

$(\hat{\mathcal{O}}1)$ *We guarantee that $\rho(t) \geq c'$ on all of $[2^{-S}, 2^{+S}]$.*

$(\hat{\mathcal{O}}2)$ *We guarantee that $\rho(t) \leq C'$ on all of $[2^{-S}, 2^{+S}]$.*

$(\hat{\mathcal{O}}3)$ *We produce a machine number $t_0 \in [2^{-S}, 2^{+S}]$ such that $c' \leq \rho(t_0) \leq C'$.*

Here, c' and C' depend only on C_1 and d.

Explanation: For each ν, we apply Low-Level Algorithm 2 to a_ν, p_ν, I_ν. If for some ν we reach outcome $(\mathcal{O}3)$, then we can trivially produce outcome $(\hat{\mathcal{O}}3)$.

If for all ν we reach outcome $(\mathcal{O}1)$, then we have produced outcome $(\hat{\mathcal{O}}1)$.

If for all ν we reach outcome $(\mathcal{O}2)$, then we have produced outcome $(\hat{\mathcal{O}}_2)$.

The only remaining case is as follows:
For each ν, we reach either outcome $(\mathcal{O}1)$ or outcome $(\mathcal{O}2)$.
For some ν, we reach $(\mathcal{O}1)$, and for some other ν, we reach $(\mathcal{O}2)$.
In this case, we can find two intervals $I_\nu, I_{\nu'}$, with an endpoint t_0 in common, and with I_ν leading to outcome $(\mathcal{O}1)$ while $I_{\nu'}$ leads to outcome $(\mathcal{O}2)$. Since $\rho(t)$ is C_1-stable, it follows that $c'' \leq \rho(t_0) \leq C''$, where c'', C'' depend only on C_1 and d. Thus, we have produced outcome $(\hat{\mathcal{O}}3)$.

The work of the algorithm is bounded by a constant depending only on the number of intervals I_ν.

45. Algorithms for Rational Functions

Algorithm RF1: *Let* $S \geq 1$ *be an integer. Suppose we are given non-zero polynomials* $p(t) = a_0 + a_1 t + \cdots + a_d t^d$ *and* $q(t) = b_0 + b_1 t + \cdots + b_d t^d$, *with* $d \geq 1$.

For each coefficient $\lambda = a_i$ *or* b_i, *assume that* λ *is a machine number, and that either* $\lambda = 0$ *or* $|\lambda| \in [2^{-S}, 2^S]$. *Assume* $\bar{S} \geq CS$ *and* $S \geq C$ *for a large enough* C *depending only on* d.

Then we produce a partition of $[2^{-S}, 2^{+S}]$ *into intervals* $I_1, \dots, I_{\nu_{\max}}$, *such that:*

(a) *Each* I_ν *is marked as being "easy" or "hard".*

(b) *If* I_ν *is easy, then we produce a machine number* λ_ν *and an integer* p_ν, *with* $|\lambda_\nu| \in [2^{-3S}, 2^{3S}]$ *and* $|p_\nu| \leq d$, *such that*

$$\left| \frac{p(t)}{q(t)} - \lambda_\nu t^{p_\nu} \right| \leq \frac{1}{2} |\lambda_\nu t^{p_\nu}| \text{ for all } t \in I_\nu .$$

(c) *If* I_ν *is hard, then* $\displaystyle\int_{I_\nu} \frac{dt}{t} < C$, *with* C *depending only on* d.

(d) $\nu_{\max} \leq C$, *with* C *depending only on* d.

(e) *The endpoints of the intervals* I_ν *are machine numbers.*

Explanation: For each pair k, k' with $a_k, a_{k'} \neq 0$, $k \neq k'$, we apply Low-Level Algorithm 1, thus partitioning $[2^{-S}, 2^S]$ into (possibly empty) intervals $I(k,k')$, $I'(k,k')$, $J(k,k')$, such that

(1) $\quad |a_{k'} t^{k'}| \leq \frac{1}{200d} |a_k t^k|$ for all $t \in I(k,k')$,

(2) $\quad |a_k t^k| \leq \frac{1}{200d} |a_{k'} t^{k'}|$ for all $t \in I'(k,k')$,

(3) $\quad \displaystyle\int_{J(k,k')} \frac{dt}{t} \leq C'$, with C' depending only on d.

We make an analogous construction for the monomials appearing in $q(t)$, thus obtaining intervals $\hat{I}(k,k')$, $\hat{I}'(k,k')$, $\hat{J}(k,k')$. The endpoints of all the (non-empty) intervals obtained above subdivide $[2^{-S}, 2^S]$ into subintervals $I_1, \dots, I_{\nu_{\max}}$. Note that the endpoints of the I_ν are machine numbers, and that $\nu_{\max} \leq C$, with C depending only on d.

We call a given I_ν "hard" if it is contained in some interval $J(k,k')$ or $\hat{J}(k,k')$; if I_ν is not "hard", then it is "easy". If I_ν is hard, then $\int_{I_\nu} \frac{dt}{t} \leq C'$, with C' depending only on d, thanks to (3) and its analogue for the $\hat{J}(k,k')$.

Suppose I_ν is easy, and suppose $a_k, a_{k'} \neq 0$, with $k \neq k'$. Then we have either $I_\nu \subseteq I(k,k')$, $I_\nu \subseteq I'(k,k')$, or $I_\nu \subseteq J(k,k')$. The third possibility is excluded, since I_ν is easy. Hence, by (1) and (2), either

$$|a_k t^k| \leq \frac{1}{200d} |a'_k t^{k'}| \text{ on all of } I_\nu \,,$$

or else

$$|a_{k'} t^{k'}| \leq \frac{1}{200d} |a_k t^k| \text{ on all of } I_\nu \,.$$

It follows that there exists $k(\nu)$, with $a_{k(\nu)} \neq 0$, such that

$$|a_{k'} t^{k'}| \leq \frac{1}{200d} |a_{k(\nu)} t^{k(\nu)}|$$

on all of I_ν, for each $k' \neq k(\nu)$. Consequently,

$$|p(t) - a_{k(\nu)} t^{k(\nu)}| \leq \frac{1}{200} |a_{k(\nu)} t^{k(\nu)}| \quad \text{for all } t \in I_\nu.$$

Similarly, for some $\hat{k}(\nu)$ with $b_{\hat{k}(\nu)} \neq 0$, we have

$$|q(t) - b_{\hat{k}(\nu)} t^{\hat{k}(\nu)}| \leq \frac{1}{200} |b_{\hat{k}(\nu)} t^{\hat{k}(\nu)}| \text{ for all } t \in I_\nu \,.$$

Thus, we obtain

$$\text{(4)} \qquad \left| \frac{p(t)}{q(t)} - \frac{a_{k(\nu)}}{b_{\hat{k}(\nu)}} t^{k(\nu)-\hat{k}(\nu)} \right| \leq \frac{1}{50} \left| \frac{a_{k(\nu)}}{b_{\hat{k}(\nu)}} t^{k(\nu)-\hat{k}(\nu)} \right| \text{ on } I_\nu \,.$$

We can easily compute $k(\nu)$ and $\hat{k}(\nu)$ for each I_ν. We set $p_\nu = k(\nu) - \hat{k}(\nu)$, and let λ_ν be our machine approximation to $a_{k(\nu)}/b_{\hat{k}(\nu)}$.

Since $|a_{k(\nu)}|, |b_{\hat{k}(\nu)}| \in [2^{-S}, 2^S]$ and $\bar{S} \geq CS$, we know that

$$\text{(5)} \qquad \left| \frac{a_{k(\nu)}}{b_{\hat{k}(\nu)}} - \lambda_\nu \right| \leq \frac{1}{50} \left| \frac{a_{k(\nu)}}{b_{\hat{k}(\nu)}} \right| .$$

In particular, $|\lambda_\nu| \in [2^{-3S}, 2^{+3S}]$. From (4) and (5) we obtain

$$\left| \frac{p(t)}{q(t)} - \lambda_\nu t^{p_\nu} \right| \leq \frac{1}{20} |\lambda_\nu t^{p_\nu}| \text{ for all } t \in I_\nu \,.$$

We have now demonstrated all the assertions (a),...,(e).

The work of the algorithm is at most a constant depending only on d.

Algorithm RF2: *Let* S, S' *be positive integers, and suppose we are given non-zero polynomials* $p(t) = a_0 + a_1 t + \cdots + a_d t^d$ *and* $q(t) = b_0 + b_1 t + \cdots + b_d t^d$, *with* $d \geq 1$*. Assume that each non-zero coefficient of* p *or* q *is a machine number whose absolute value lies in* $[2^{-S'}, 2^{+S'}]$.
Assume that $\rho(t) = \frac{p(t)}{q(t)}$ *is* C_1*-stable on* $[2^{-S}, 2^{S}]$*. (In particular, we assume that* $\rho(t) > 0$ *on* $[2^{-S}, 2^{S}]$*.)*
Assume also that $\bar{S} \geq C \cdot (S + S')$ *and* $S \geq C$*, where* C *is a large enough constant determined by* C_1 *and* d*. Then we can produce intervals* I_ν*, machine numbers* λ_ν*, and integers* p_ν*, for* $\nu = 1, \dots, \nu_{\max}$*, with the following properties:*

(a) $|\lambda_\nu| \in [2^{-C(S+S')}, 2^{+C(S+S')}]$ *for an integer constant* C *depending only on* d.

(b) $|p_\nu| \leq d$.

(c) *The* I_ν *form a partition of* $[2^{-S}, 2^{+S}]$.

(d) *The endpoints of* I_ν *are machine numbers.*

(e) $c'\rho(t) \leq \lambda_\nu t^{p_\nu} \leq C'\rho(t)$ *for all* $t \in I_\nu$*, with* c' *and* C' *depending only on* C_1 *and* d.

(f) $\nu_{\max} \leq C'$*, with* C' *depending only on* d.

Explanation: Apply Algorithm RF1 to p and q, with $S+S'$ in place of S. Thus, we obtain a partition of $[2^{-(S'+S)}, 2^{+(S'+S)}]$ into intervals. Intersecting these intervals with $[2^{-S}, 2^{+S}]$, and discarding any empty intervals that arise, we obtain a partition $I_1, \dots, I_{\nu_{\max}}$ of $[2^{-S}, 2^{+S}]$ into intervals, with the following properties.

- Each I_ν is marked as being "easy" or "hard".
- Each hard interval I_ν satisfies $\int_{I_\nu} \frac{dt}{t} \leq C'$, with C' depending only on d.
- Each easy interval I_ν is marked with a machine number λ_ν and an integer p_ν, such that $|p_\nu| \leq d$, $|\lambda_\nu| \in [2^{-3(S+S')}, 2^{3(S+S')}]$, and

$$\left| \frac{p(t)}{q(t)} - \lambda_\nu t^{p_\nu} \right| \leq \frac{1}{2} |\lambda_\nu t^{p_\nu}| \text{ for all } t \in I_\nu.$$

The endpoints of the I_ν are machine numbers, and $\nu_{\max}$ is less than a constant determined by d.

It remains to deal with the "hard" intervals I_ν.

Fix a hard interval I_ν, and let t_ν be an endpoint of I_ν. (So, $t_\nu \in [2^{-S}, 2^S]$.) Using our model of computation, we compute $t_{\nu,\ell} :=$ (our approximation of) $2^\ell \cdot t_\nu$, for all integers ℓ with $|\ell| \leq \underline{C}$. Here, $\underline{C}$ is a large enough constant depending only on d, to be determined later on.

Assume, as we may, that $S \geq 2^{\underline{C}}$. It follows that either all the $t_{\nu,\ell}$ with $0 \leq \ell \leq \underline{C}$, or all the $t_{\nu,\ell}$ with $-\underline{C} \leq \ell \leq 0$, belong to $[2^{-S}, 2^S]$. We know also that $\nu_{\max} \leq C$ and $\int_{I_{\nu'}} \frac{dt}{t} \leq C$ for each hard interval $I_{\nu'}$; here, C depends only on d. Consequently, for an appropriate choice of the constant $\underline{C}$, one of the $t_{\nu,\ell}$ belongs to $[2^{-S}, 2^S]$ minus the union of all the hard intervals. We define $\underline{C}$ to be such an appropriate constant, depending only on d.

Fix such a $t_{\nu,\ell}$; we can easily find it.

Since the $I_{\nu'}$ $(\nu' = 1, \ldots, \nu_{\max})$ form a partition of $[2^{-S}, 2^S]$, we know that our $t_{\nu,\ell}$ belongs to an easy interval I_μ. We can easily find I_μ.

Note that $t_{\nu,\ell} \in [2^{-S}, 2^{+S}]$, and that $\hat{c} \leq t/t_{\nu,\ell} \leq \hat{C}$ for all $t \in I_\nu$, with $\hat{c}$ and $\hat{C}$ determined by d. (Here, we use the fact that $t_{\nu,\ell}$ differs from an endpoint of I_ν by (approximately) a factor 2^ℓ, with $|\ell| \leq \underline{C}$.) Since $\rho(t)$ is C_1-stable on $[2^{-S}, 2^S]$, it follows that

(*1) $c'\rho(t_{\nu,\ell}) \leq \rho(t) \leq C'\rho(t_{\nu,\ell})$ for all $t \in I_\nu$, with c' and C' depending only on d and C_1.

On the other hand, since $t_{\nu,\ell}$ belongs to the easy interval I_μ, we have

(*2) $c \cdot \lambda_\mu(t_{\nu,\ell})^{p_\mu} \leq \rho(t_{\nu,\ell}) \leq C \cdot \lambda_\mu(t_{\nu,\ell})^{p_\mu}$ with c and C depending only on d.

We have already computed the λ_μ and p_μ. We now define $p_\nu := 0$, and we take λ_ν to be

(*3) (our machine approximation of) $\lambda_\mu \cdot (t_{\nu,\ell})^{p_\mu}$.

The roundoff error in computing λ_ν is at most $\frac{1}{100}\lambda_\nu$. From (*1) and (*2), we therefore obtain:

(*4) $c''\lambda_\nu t^{p_\nu} \leq \rho(t) \leq C''\lambda_\nu t^{p_\nu}$ for all $t \in I_\nu$, with c'', C'' determined by d and C_1.

Also, since $t_{\nu,\ell} \in [2^{-S}, 2^{+S}]$ and λ_μ, p_μ satisfy (a) and (b) in the statement of Algorithm RF2, then a glance at (*3) shows that λ_ν and p_ν also satisfy (a) and (b). From (*4) we obtain (e). Recall that I_ν was an arbitrary hard interval. Thus, we have satisfied (a), (b), and (e) also in the hard case.

Properties (c), (d), (f) hold, since the I_ν arose from Algorithm RF1. So, our $I_\nu, \lambda_\nu, p_\nu$ have all the properties asserted in Algorithm RF2.

The work of the algorithm is at most a constant depending only on d and C_1.

46. Systems of Inequalities with Parameters

Algorithm SIP1: *Let* S *be a positive integer. Suppose we are given machine numbers* $\lambda_{\ell j}$ $(1 \le j \le D, 1 \le \ell \le L)$, b_ℓ $(1 \le \ell \le L)$, σ_ℓ $(1 \le \ell \le L)$ *and integers* m_ℓ $(1 \le \ell \le L)$.
Assume that $0 \le m_\ell \le d$, $|\lambda_{\ell j}| \le 2^S$, $|b_\ell| \le 2^S$, $2^{-S} \le \sigma_\ell \le 2^S$.
Assume also that $\bar{S} \ge CS$ *and* $S \ge C$, *for a large enough* C *determined by* d, D, L. *Then we produce one of the following three outcomes.*

(I) *We guarantee that, for any* $\delta \in [2^{-S}, 2^{+S}]$, *there exists* $(v_1, \dots, v_D) \in \mathbb{R}^D$ *such that*

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right| \le C\sigma_\ell \delta^{-m_\ell} \quad \text{for } \ell = 1, \dots, L;$$

and $|v_j| \le C \cdot 2^S$ *for* $j = 1, \dots, D$. *Here* C *depends only on* d, D, L.

(II) *We guarantee that, for any* $\delta \in [2^{-S}, 2^S]$, *there does not exist* $(v_1, \dots, v_D) \in \mathbb{R}^D$ *such that*

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right| \le c\sigma_\ell \delta^{-m_\ell} \quad \text{for } \ell = 1, \dots, L;$$

and $|v_j| \le c \cdot 2^S$ *for* $j = 1, \dots, D$. *Here,* c *depends only on* d, D, L.

(III) *We produce a machine number* $\delta \in [2^{-S}, 2^S]$ *such that*

(A) *There exists* $(v_1, \dots, v_D) \in \mathbb{R}^D$ *such that*

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right| \le C\sigma_\ell \delta^{-m_\ell} \quad \text{for } \ell = 1, \dots, L;$$

and $|v_j| \le C \cdot 2^S$ *for* $j = 1, \dots, D$. *Here,* C *depends only on* d, D, L.

(B) *There does not exist* $(v_1, \dots, v_D) \in \mathbb{R}^D$ *such that*

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right| \le c\sigma_\ell \delta^{-m_\ell} \quad \text{for } \ell = 1, \dots, L\,;$$

and $|v_j| \le c \cdot 2^S$ *for* $j = 1, \dots, D$. *Here,* c *depends only on* d, D, L.

Explanation: We write, c, C, C_1, C', etc. to denote constants depending only on d, D, L. For $\delta > 0$ and $v = (v_1, \dots, v_D) \in \mathbb{R}^D$, let

$$(1) \qquad Q(v, \delta) = \sum_{\ell=1}^{L} \left(\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right)^2 \sigma_\ell^{-2} \delta^{2m_\ell} + \sum_{j=1}^{D} 2^{-2S}(v_j)^2 + 2^{-2S}.$$

For a fixed δ, $Q(\nu, \delta)$ is a quadratic function in ν:

$$Q(\nu, \delta) = \sum_{i,j=1}^{D} \Lambda_{ij} \nu_i \nu_j + \sum_{i=1}^{D} E_i \nu_i + H.$$

Here, $\Lambda_{ij} = \Lambda_{ji}$, and Λ_{ij}, E_i, H are polynomials in $\delta, \lambda_{\ell j}, b_\ell, \sigma_\ell^{-1}, 2^{-S}$ with rational coefficients and degree at most C. Note that, as matrices, $(\Lambda_{ij}) \geq 2^{-2S} I$, where I denotes the identity matrix.
Hence, as ν varies and all else remains fixed, $Q(\nu, \delta)$ attains a minimum. The minimizer is obtained by solving:

$$2\sum_{j=1}^{D} \Lambda_{ij}\nu_j + E_i = 0 \text{ for } i = 1, \dots, D.$$

Hence,

$$\text{(2)} \qquad \rho(\delta) := \min_{\nu \in \mathbb{R}^D} Q(\nu, \delta)$$

may be expressed in the form

$$\text{(3)} \qquad \rho(\delta) = p/q,$$

where p and q are polynomials in $(\delta, \lambda_{\ell j}, b_\ell, \sigma_\ell^{-1}, 2^{-S})$, with rational coefficients and degree at most C. The coefficients of p and q depend only on d, D, L. In fact, $q = \det^2(\Lambda_{ij})$, and consequently

$$\text{(4)} \qquad q \geq 2^{-CS} \text{ for any } (\delta, \lambda_{\ell j}, b_\ell, \sigma_\ell, S),$$

since $(\Lambda_{ij}) \geq 2^{-2S} I$. By the definition of $Q(\nu, \delta)$, we have $Q(\nu, \delta) \geq 2^{-2S}$, and

$$Q(\nu, \delta') \leq Q(\nu, \delta) \leq \left(\frac{\delta}{\delta'}\right)^{2d} Q(\nu, \delta') \text{ for } 0 < \delta' < \delta.$$

(Recall that $m_\ell \geq 0$ for $\ell = 1, \dots, L$.) Hence, by the definition of $\rho(\delta)$, we have

$$\text{(5)} \qquad \rho(\delta) \geq 2^{-2S}, \text{ and}$$

$$\text{(6)} \qquad \rho(\delta') \leq \rho(\delta) \leq \left(\tfrac{\delta}{\delta'}\right)^{2d} \rho(\delta') \text{ for } 0 < \delta' < \delta.$$

In particular, $\rho(\delta)$ is C_1-stable on $(0, \infty)$, with C_1 depending only on d.

Now recall that $|\lambda_{\ell j}|, |b_\ell| \leq 2^S$, $\sigma_\ell^{-1} \leq 2^S$. Then p and q may be regarded as polynomials in δ, with degree $\leq C$ and coefficients of absolute value $\leq 2^{CS}$.

We call these polynomials $p(\delta)$ and $q(\delta)$, respectively. We attempt to compute the coefficients of $p(\delta), q(\delta)$ from the given $\lambda_{\ell j}, b_\ell, \sigma_\ell, S$ in our model of computation. Thus, we obtain polynomials $\bar{p}(\delta)$, $\bar{q}(\delta)$ whose coefficients differ from those of $p(\delta)$, $q(\delta)$ by at most $2^{CS} \cdot 2^{-\bar{S}}$, due to round-off error.

Hence,

(7) $|\bar{p}(\delta) - p(\delta)|\,, |\bar{q}(\delta) - q(\delta)| \leq 2^{C'S} \cdot 2^{-\bar{S}}$ for $0 < \delta \leq 2^S$.

Also, our estimates for the size of the coefficients of $p(\delta)$, $q(\delta)$ yield

(8) $|p(\delta)|\,, |q(\delta)| \leq 2^{CS}$ for $0 < \delta \leq 2^S$.

From (4), (5), (8), we learn that

$$p(\delta),\; q(\delta) \in [2^{-CS}, 2^{+CS}] \text{ for } 0 < \delta \leq 2^S,$$

and therefore (7) yields

$$|\bar{p}(\delta) - p(\delta)| \leq 10^{-3} p(\delta) \text{ and } |\bar{q}(\delta) - q(\delta)| \leq 10^{-3} q(\delta) \text{ for } 0 < \delta \leq 2^S.$$

Consequently,

(9) $\bar{p}(\delta),\; \bar{q}(\delta) \in [2^{-CS}, 2^{+CS}]$ for $0 < \delta \leq 2^S$, and

(10) $\frac{1}{2}\frac{\bar{p}(\delta)}{\bar{q}(\delta)} \leq \rho(\delta) \leq 2\frac{\bar{p}(\delta)}{\bar{q}(\delta)}$ for $0 < \delta \leq 2^S$, thanks to (3).

Let us write

(11) $\bar{p}(\delta) = \sum_{0 \leq k \leq k_{\max}} \bar{p}_k \delta^k$ and $\bar{q}(\delta) = \sum_{0 \leq k \leq k_{\max}} \bar{q}_k \delta^k$.

We define new polynomials $\bar{\bar{p}}(\delta)$, $\bar{\bar{q}}(\delta)$ by deleting from (11) all terms with $|\bar{p}_k| \leq 2^{-C^*S}$ or $|\bar{q}_k| \leq 2^{-C^*S}$, respectively. Here, C^* is a large enough integer constant determined from d, D, L. Taking C^* large enough, we learn from (9) that

$$|\bar{\bar{p}}(\delta) - \bar{p}(\delta)| \leq 10^{-3} \bar{p}(\delta)\,, \text{ and } |\bar{\bar{q}}(\delta) - \bar{q}(\delta)| \leq 10^{-3} \bar{q}(\delta) \text{ for } 0 < \delta \leq 2^S.$$

Hence, (9) and (10) yield

(12) $\frac{1}{4}\frac{\bar{\bar{p}}(\delta)}{\bar{\bar{q}}(\delta)} \leq \rho(\delta) \leq 4\frac{\bar{\bar{p}}(\delta)}{\bar{\bar{q}}(\delta)}$ for $0 < \delta \leq 2^S$, and

(13) $\bar{\bar{p}}(\delta),\; \bar{\bar{q}}(\delta) \in [2^{-CS},\, 2^{+CS}]$ for $0 < \delta \leq 2^S$.

By construction,

(14) $\bar{\bar{p}}$, $\bar{\bar{q}}$ are non-zero polynomials of degree $\leq C$, whose non-zero coefficients are all machine numbers belonging to

$$[2^{-CS}, 2^{+CS}] \cup [-2^{+CS},\, -2^{-CS}].$$

Also, since $\rho(\delta)$ is C_1-stable on $(0, \infty)$, it follows from (12) that

(15) $\bar{\bar{\rho}}(\delta) := \bar{\bar{p}}(\delta)/\bar{\bar{q}}(\delta)$ is C_1'-stable on $[2^{-S}, 2^{+S}]$.

We can now apply Algorithm RF2 to the polynomials $\bar{\bar{p}}(\delta)$ and $\bar{\bar{q}}(\delta)$. (See Section 45.) The algorithm applies, thanks to (14) and (15). From Algorithm RF2, we obtain intervals I_ν, machine numbers λ_ν, and integers p_ν $(1 \leq \nu \leq \nu_{\max})$, with the following properties:

$$(16) \qquad \left[\begin{array}{l} 2^{-CS} \leq |\lambda_\nu| \leq 2^{CS},\ |p_\nu| \leq C\,, \nu_{\max} \leq C\,; \\ \text{the endpoints of the } I_\nu \text{ are machine numbers;} \\ \text{the } I_\nu \text{ form a partition of } [2^{-S}, 2^S]\,; \text{and} \\ c\lambda_\nu \delta^{p_\nu} \leq \frac{\bar{\bar{p}}(\delta)}{\bar{\bar{q}}(\delta)} \leq C\lambda_\nu \delta^{p_\nu} \text{ on } I_\nu\,. \end{array}\right]$$

We define $\hat{\rho}(t) = \lambda_\nu t^{p_\nu}$ for $t \in I_\nu$. Thus, $\hat{\rho}$ is defined on $[2^{-S}, 2^{+S}]$. From (15) and (16), we have

(17) $\hat{\rho}(t)$ is C-stable on $[2^{-S}, 2^{+S}]$.

Thanks to (16), (17), we can apply Low-Level Algorithm 3 from Section 44. (To apply that algorithm, we first extend $\hat{\rho}(t)$ to $[2^{-CS}, 2^{CS}]$ by taking it to be constant on $[2^{-CS}, 2^{-S}]$ and on $[2^S, 2^{CS}]$.) Using that algorithm, we produce one of the following outcomes:

($\hat{\mathcal{O}}1$) We guarantee that $\hat{\rho}(t) \geq c$ on all of $[2^{-S}, 2^{+S}]$.

($\hat{\mathcal{O}}2$) We guarantee that $\hat{\rho}(t) \leq C$ on all of $[2^{-S}, 2^{+S}]$.

($\hat{\mathcal{O}}3$) We produce a machine number $t_0 \in [2^{-S}, 2^{+S}]$, such that $c < \hat{\rho}(t_0) < C$.

(In case where Low-Level Algorithm 3 outputs $t_0 \in [2^{-CS}, 2^{-S})$ such that $c < \hat{\rho}(t_0) < C$, we replace t_0 with 2^{-S}. Note that ($\hat{\mathcal{O}}3$) still holds, since $\hat{\rho}$ is constant on $[2^{-CS}, 2^{-S}]$. Similarly, if Low-Level Algorithm 3 outputs $t_0 \in (2^S, 2^{CS}]$, we replace t_0 with 2^S.)

From (12) and (16), we see that

$$c\hat{\rho}(t) \leq \rho(t) \leq C\hat{\rho}(t) \quad \text{on } [2^{-S}, 2^{+S}],$$

and therefore we may take $\rho(t)$ in place of $\hat{\rho}(t)$ in ($\hat{\mathcal{O}}1$), ($\hat{\mathcal{O}}2$), ($\hat{\mathcal{O}}3$). Recalling the definition (2) and the fact that $Q(\nu, \delta)$ increases with δ, we conclude that we have produced one of the following three outcomes:

($\hat{\mathcal{O}}1$) We guarantee that, for every $\delta \in [2^{-S}, 2^{+S}]$, there does not exist $\nu \in \mathbb{R}^D$ such that $Q(\nu, \delta) \leq c$.

($\hat{\mathcal{O}}2$) We guarantee that, for every $\delta \in [2^{-S}, 2^{+S}]$, there exists $\nu \in \mathbb{R}^D$ such that $Q(\nu, \delta) \leq C$.

($\hat{\mathcal{O}}3$) We produce a machine number $\delta \in [2^{-S}, 2^{S}]$, with the following properties:

(A) There exists $\nu \in \mathbb{R}^D$ such that $Q(\nu, \delta) \leq C$.

(B) There does not exist $\nu \in \mathbb{R}^D$ such that $Q(\nu, \delta) \leq c$.

We now recall the definition (1) of $Q(\nu, \delta)$.
From (1), we learn the following.

$$\left[\begin{array}{l} \text{Let } \nu \in \mathbb{R}^D,\ \delta > 0\,,\ \text{and suppose } \ Q(\nu,\delta) \ \leq \ C\,.\ \text{Then} \\ \left|\sum_{j=1}^{D} \lambda_{\ell j} \nu_j \ - \ b_\ell\right| \ \leq \ C\sigma_\ell \delta^{-m_\ell} \text{ for } \ell \ = \ 1, \ldots, L\,;\ \text{and} \\ |\nu_j| \ \leq \ C \ \cdot \ 2^S \text{ for } j \ = \ 1, \ldots, D\,. \end{array}\right] \tag{18}$$

$$\left[\begin{array}{l} \text{Let } \nu \in \mathbb{R}^D\,, \delta \ > \ 0\,,\ \text{and suppose that for some } K > 0 \text{ we have} \\ \left|\sum_{j=1}^{D} \lambda_{\ell j} \nu_j \ - \ b_\ell\right| \ \leq \ K\sigma_\ell \delta^{-m_\ell} \text{ for } \ell \ = \ 1, \ldots, L;\ \text{and} \\ |\nu_j| \ \leq \ K \ \cdot \ 2^S \text{ for } j \ = \ 1, \ldots, D\,. \\ \text{Then } Q(\nu,\delta) \ \leq \ (L \ + \ D)\, K^2 \ + \ 2^{-2S}\,. \end{array}\right] \tag{19}$$

In particular, suppose c depends only on d, D, L (as in ($\hat{\mathcal{O}}1$), ..., ($\hat{\mathcal{O}}3$)). Then we have $(L + D)\, K^2 + 2^{-2S} < c$ for K a small enough constant depending only on d, D, L. (Here, we use our assumption that S exceeds a large enough constant depending only on d, D, L.) Hence, (18) and (19) allow us to understand what happens for each of the outcomes ($\hat{\mathcal{O}}1$), ($\hat{\mathcal{O}}2$), ($\hat{\mathcal{O}}3$). Thus, we produce one of the following outcomes:

($\hat{\mathcal{O}}1$) We guarantee that, for all $\delta \in [2^{-S}, 2^{+S}]$, there does not exist $(\nu_1, \ldots, \nu_D) \in \mathbb{R}^D$ such that

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} \nu_j \ - \ b_\ell\right| \ \leq \ c\sigma_\ell \delta^{-m_\ell} \ \text{ for } \ \ell \ = \ 1, \ldots, L$$

and $|\nu_j| \ \leq \ c2^S$ for $j \ = \ 1, \ldots, D\,.$

($\hat{\mathcal{O}}2$) We guarantee that, for all $\delta \in [2^{-S}, 2^{+S}]$, there exists $(\nu_1, \ldots, \nu_D) \in \mathbb{R}^D$ such that

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} \nu_j \ - \ b_\ell\right| \ \leq \ C\sigma_\ell \delta^{-m_\ell} \text{ for } \ell \ = \ 1, \ldots, L$$

and $|\nu_j| \ \leq \ C2^S$ for $j \ = \ 1, \ldots, D\,.$

($\hat{\mathrm{O}}3$) We produce a machine number $\delta \in [2^{-S}, 2^S]$, with the following properties:

(A) There exists $(v_1, \dots, v_D) \in \mathbb{R}^D$ such that

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right| \leq C\sigma_\ell \delta^{-m_\ell} \text{ for } \ell = 1, \dots, L$$

and $|v_j| \leq C\,2^S$ for $j = 1, \dots, D$.

(B) There does not exist $(v_1, \dots, v_D) \in \mathbb{R}^D$ such that

$$\left|\sum_{j=1}^{D} \lambda_{\ell j} v_j - b_\ell\right| \leq c\sigma_\ell \delta^{-m_\ell} \text{ for } \ell = 1, \dots, L$$

and $|v_j| \leq c \cdot 2^S$ for $j = 1, \dots, D$.

The above three outcomes are precisely those promised in the statement of Algorithm SIP1.

Thus, we have carried out that algorithm.

The work of the algorithm is at most a constant depending only on d, D, L.

47. Equivalence above a Threshold

Let $\mathcal{K} = (K_M)_{M>0}$ and $\mathcal{K}' = (K'_M)_{M>0}$ be two blobs in a vector space V; and let $M_0 \geq 0$ and $A \geq 1$ be real numbers. We say that $\mathcal{K}$ and $\mathcal{K}'$ are "A-equivalent above M_0" if they satisfy

$$K_M \subseteq K'_{AM} \text{ for } M \geq M_0\,, \text{ and } K'_M \subseteq K_{AM} \text{ for } M \geq M_0\,.$$

The following remarks are obvious:

- If $\mathcal{K}$ and $\mathcal{K}'$ are A-equivalent, then they are A-equivalent above M_0. More generally, if $\mathcal{K}, \mathcal{K}'$ are A-equivalent above M_0, then they are A'-equivalent above M'_0, for any $M'_0 \geq M_0, A' \geq A$.
- If $\mathcal{K}$ and $\mathcal{K}'$ are A-equivalent above M_0, then they are TA-equivalent above M_0/T, for all $T \geq 1$.
- If $\mathcal{K}$ and $\mathcal{K}'$ are A-equivalent above M_0, and if $\mathcal{K}'$ and $\mathcal{K}''$ are $\tilde{A}$-equivalent above $\tilde{M}_0$, then $\mathcal{K}$ and $\mathcal{K}''$ are $A \cdot \tilde{A}$-equivalent above $\max(M_0, \tilde{M}_0)$.
- Let $\mathcal{K}_\nu$, $\mathcal{K}'_\nu$ be A-equivalent above M_0, for each $\nu = 1, \dots, \nu_{\max}$. Then $\cap_{1\leq\nu\leq\nu_{\max}} \mathcal{K}_\nu$ is A-equivalent to $\cap_{1\leq\nu\leq\nu_{\max}} \mathcal{K}'_\nu$ above M_0.
- If $\mathcal{K}_1, \mathcal{K}'_1$ are A-equivalent above M_0, and if $\mathcal{K}_2, \mathcal{K}'_2$ are A-equivalent above M_0, then $\mathcal{K}_1 + \mathcal{K}_2$ and $\mathcal{K}'_1 + \mathcal{K}'_2$ are A-equivalent above M_0.

48. Some Particular FALPs and MALPs

Recall that $\mathcal{P}$ stands for the vector space of all polynomials of degree at most $m-1$ on $\mathbb{R}^n$, and let $D = \dim \mathcal{P}$. Recall the remark following Algorithm MALP9 from Section 43. We identify $\mathcal{P}$ with $\mathbb{R}^D$, by identifying $P \in \mathcal{P}$ with $(\partial^\alpha P(0))_{|\alpha| \leq m-1} \in \mathbb{R}^D$. We write c, C, C', etc. for constants depending only on m and n.

For $x \in \mathbb{R}^n$ and $\delta > 0$, and for an integer S, we set $\mathcal{B}^{[S]}(x,\delta) = (B^{[S]}(x,\delta,M))_{M>0}$, where

$$B^{[S]}(x,\delta,M) = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M\delta^{m-|\alpha|} \text{ for } |\alpha| \leq m-1\} \text{ when } M \geq 2^{-S}$$

and

$$B^{[S]}(x,\delta,M) = \emptyset \text{ for } M < 2^{-S}.$$

Note that $\mathcal{B}^{[S]}(x,\delta)$ is 1-equivalent to $\mathcal{B}(x,\delta)$ above 2^{-S}, where $\mathcal{B}(x,\delta)$ is the blob considered throughout this manuscript (see, e.g., Section 10).

Algorithm PFM1: *Let S be a positive integer. Assume $\bar{S} \geq CS$ and $S \geq C$ for a large enough C. Given machine numbers $x_1, \ldots, x_n, \delta$, with $|x_i| \leq 2^S$ for each i, and with $2^{-S} \leq \delta \leq 2^S$, we construct a $C'S$-bit* MALP *$\mathcal{A}$ with constant C', such that the blob arising from $\mathcal{A}$ is C'-equivalent to the blob $\mathcal{B}^{[S]}(x,\delta)$, where $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Moreover, length $(\mathcal{A}) = D$.*

Explanation: Since

$$\partial^\alpha P(x) = \sum_{|\beta| \leq m-1-|\alpha|} \frac{1}{\beta!} \partial^{\beta+\alpha} P(0) x^\beta,$$

the blob $\mathcal{B}^{[S]}(x,\delta)$ arises from the ALP

$$\mathcal{A}_{x,\delta} = [(\lambda_{\alpha,\gamma})_{|\alpha|,|\gamma| \leq m-1}, (b_\alpha)_{|\alpha| \leq m-1}, (\sigma_\alpha)_{|\alpha| \leq m-1}, M_*],$$

with $\lambda_{\alpha,\alpha+\beta} = x^\beta/\beta!$ for $|\alpha| + |\beta| \leq m-1$; all other $\lambda_{\alpha,\gamma} = 0$; $b_\alpha = 0$, $\sigma_\alpha = \delta^{m-|\alpha|}$ for $|\alpha| \leq m-1$; and $M_* = 2^{-S}$.

Under our ordering on multi-indices, α always precedes $\alpha + \beta$ (for non-zero multi-indices β), and therefore our matrix $(\lambda_{\alpha,\gamma})$ is upper triangular, with 1's on the main diagonal. Consequently, Lemma 3 in Section 40 guarantees that $\mathcal{A}_{x,\delta}$ is a $C'S$-bit FALP with constant 2.

Using our model of computation, we produce machine approximations to the $\lambda_{\alpha,\gamma}$, σ_α; the b_α and M_* can be represented perfectly by machine numbers. Thus, we obtain an ALP $\mathcal{A}$. Since roundoff errors change the $\lambda_{\alpha,\gamma}$, σ_α by at most $2^{CS} \cdot 2^{-\bar{S}}$, it follows from Lemma 1 in Section 40 that $\mathcal{A}$ is $C'S$-bit MALP with constant C', and that $\mathcal{A}$ is C'-equivalent to $\mathcal{A}_{x,\delta}$. Clearly length $(\mathcal{A}) = D$. Thus, we have produced the desired MALP $\mathcal{A}$. The work of the algorithm is at most C'.

Next, suppose we are given $x \in \mathbb{R}^n$, $\sigma > 0$, $t \in \mathbb{R}$, and an integer S. We define the blob $\Gamma[x, \sigma, t, S] = (\Gamma_M)_{M>0}$, where

$$\Gamma_M = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M \quad \text{for } |\alpha| \leq m-1 \text{ and } |P(x) - t| \leq M\sigma\}$$

when $M \geq 2^{-S}$; and $\Gamma_M = \emptyset$ for $M < 2^{-S}$.

Algorithm PFM2: *Let* S *be a positive integer. Assume* $\bar{S} \geq CS$ *and* $S \geq C$ *for large enough* C*. Given machine numbers* $x_1, \ldots x_n, \sigma, t$*, with* $|x_i| \leq 2^S$ *for each* i*;* $2^{-S} \leq \sigma \leq 2^S$*;* $|t| \leq 2^S$*; we produce a* $C'S$*-bit* MALP $\mathcal{A}$ *with constant* C'*, such that the blob arising from* $\mathcal{A}$ *is* C'*-equivalent to* $\Gamma[x, \sigma, t, S]$*, with* $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$*. Moreover,* length $(\mathcal{A}) = D$.

Explanation: Fix $x_1, \ldots, x_n$, σ, t, S as above. Let Γ^0 and Γ^1 be the blobs defined as follows.

$\Gamma^0 = \left(\Gamma^0_M\right)_{M>0}$ and $\Gamma^1 = \left(\Gamma^1_M\right)_{M>0}$,

$\Gamma^0_M = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \leq M$ for $|\alpha| \leq m-1\}$ when $M \geq 2^{-S}$,

$\Gamma^0_M = \emptyset$ when $M < 2^{-S}$,

$\Gamma^1_M = \{P \in \mathcal{P} : |P(x) - t| \leq M\sigma$, $|\partial^\alpha P(x)| \leq M$ for $0 < |\alpha| \leq m-1\}$ when $M \geq 2^{-S}$;

$\Gamma^1_M = \emptyset$ when $M < 2^{-S}$.

Then $\Gamma[x, \sigma, t, S]$ is the intersection of Γ^0 and Γ^1. Since $\Gamma^0 = \mathcal{B}^{[S]}(x, 1)$, our previous algorithm (Algorithm PFM1) constructs a $C'S$-bit MALP $\mathcal{A}^0$ with constant C', such that Γ^0 is C'-equivalent to the blob arising from $\mathcal{A}^0$.

We study Γ^1 similarly. With $\lambda_{\alpha,\gamma}$ as in the explanation of Algorithm PFM1, we define an ALP

$$\mathcal{A}^1_{x,\sigma,t,S} = [(\lambda_{\alpha,\gamma})_{|\alpha|,|\gamma| \leq m-1}, (b_\alpha)_{|\alpha| \leq m-1}, (\sigma_\alpha)_{|\alpha| \leq m-1}, M_*],$$

by setting

$$b_0 = t, b_\alpha = 0 \text{ for } \alpha \neq 0, \sigma_0 = \sigma, \sigma_\alpha = 1 \text{ for } \alpha \neq 0, M_* = 2^{-S}.$$

Then Γ^1 is precisely the blob arising from $\mathcal{A}^1_{x,\sigma,t,S}$.

On the other hand, Lemmas 1 and 3 in Section 40 show that

- $\mathcal{A}^1_{x,\sigma,t,S}$ is a $C'S$-bit FALP with constant C',

and that

- If we approximate the $\lambda_{\alpha,\gamma}$ to within an error $2^{C'S} \cdot 2^{-\bar{S}}$ by machine numbers, then in place of $\mathcal{A}^1_{x,\sigma,t,S}$, we obtain a $C'S$-bit MALP $\mathcal{A}^1$ with constant C', such that $\mathcal{A}^1$ and $\mathcal{A}^1_{x,\sigma,t,S}$ are C'-equivalent.

Using our model of computation, we can compute $\mathcal{A}^1$. Thus, Γ^1 is C'-equivalent to the blob arising from $\mathcal{A}^1$.

We now apply Algorithm MALP7 followed by Algorithm MALP1, to form an "approximate intersection" of the blobs arising from $\mathcal{A}^1$ and $\mathcal{A}^0$. (See Section 43.)

Thus, we obtain a $C'S$-bit MALP $\mathcal{A}$ with constant C', of length D, such that the blob arising from $\mathcal{A}$ is C'-equivalent to $\Gamma^0 \cap \Gamma^1 = \Gamma[x, \sigma, t, S]$.

We have succeeded in constructing $\mathcal{A}$ with the desired properties.

The work of the algorithm is at most C'.

49. Set-Up

Fix $m, n \geq 1$. Fix also a positive integer S_0. The basic precision of our algorithm will have the order of magnitude of S_0 bits.

We will again be using the constants $p_\#, A_0, A_1(\mathcal{A}), A_2, A_3(\mathcal{A})$ from Section 17, and the constant ℓ_* from Section 14. Recall that, in Section 17, we demanded that A_0 and $p_\#$ exceed certain constants depending only on m, n. We will impose additional similar lower bounds on A_0 in the sections to follow.

We assume the following conditions.

(SU1) S_0 exceeds a large enough constant determined by $A_0, p_\#$, m, n, and

(SU2) $\bar{S}/S_0$ exceeds a large enough constant determined by $A_0, p_\#$, m, n.

(Recall from Section 38 that $\bar{S}$ is the precision of our model of computation.)

In the end, we will take A_0 and $p_\#$ to be the smallest powers of two that satisfy all the lower bounds we have imposed on them. Thus, finally, $p_\#, A_0, A_2$ will depend only on m and n; while $A_1(\mathcal{A})$ and $A_3(\mathcal{A})$ will depend only on $m, n, \mathcal{A}$. Also, finally, (SU1) and (SU2) simply assert that S_0 and $\bar{S}/S_0$ exceed a large enough constant determined by m and n.

We are given a finite set E of points in $\mathbb{R}^n$. Each coordinate x_j of each point $x = (x_1, \dots, x_n) \in E$ is assumed to be a machine number. We assume that $|x| \leq 2^{S_0}$ for any $x \in E$, and that $|x - y| \geq 2^{-S_0}$ for any two distinct points $x, y \in E$.

We are given functions $f : E \longrightarrow \mathbb{R}$ and $\sigma : E \longrightarrow \mathbb{R}^+$. For each $x \in E$, we assume that $f(x)$ is a machine number, with $|f(x)| \leq 2^{S_0}$, and that $\sigma(x)$ is a machine number, with $2^{-S_0} \leq \sigma(x) \leq 2^{S_0}$.

We write $\mathcal{P}$ for the vector space of (real) $(m-1)^{\text{rst}}$ degree polynomials on $\mathbb{R}^n$, and we set $D = \dim \mathcal{P}$. We identify $\mathcal{P}$ with $\mathbb{R}^D$, by identifying $P \in \mathcal{P}$ with $(\partial^\alpha P(0))_{|\alpha| \leq m-1} \in \mathbb{R}^D$. To "compute a polynomial" is to compute a vector in $\mathbb{R}^D$, which we identify with a polynomial $P \in \mathcal{P}$, as above.

50. The Basic Blobs

Here, $C, C', \tilde{C}$, etc. denote constants depending only on m and n. From our earlier paper [19], and from Section 9 we recall the tree $\mathcal{T}$, whose nodes $A, B, \dots$ are subsets of E. We recall also the Callahan-Kosaraju decomposition $\mathcal{L}$, whose elements are pairs (Λ_1, Λ_2), with Λ_i being a set of nodes of $\mathcal{T}$.

The Callahan-Kosaraju decomposition may be implemented in our model of computation, under the assumptions (SU1) and (SU2) from Section 49. Indeed, $\mathcal{T}$ and $\mathcal{L}$, the output of the Callahan-Kosaraju algorithm, are "inherently discrete" objects. The issues of precision that arise when attempting to compute $\mathcal{T}$ and $\mathcal{L}$ are minor. In particular, the "fair split tree" of [11] may be computed with perfect precision: We just need to work with boxes whose vertices all have coordinates that are integer multiples of 2^{-CS_0}, for an integer constant C, determined by n. In addition, for any $x, y \in E$, we may compute in our model of computation a machine number that approximates $|x-y|^2$ to within a factor of 2. It is straightforward to see, that when applying the "$\varkappa$-WSPD algorithm" in our model of computation, we obtain a list that satisfies the requirements of a $4\varkappa$-WSPD. Consequently, using $CN \log N$ work and CN storage, we may compute a Callahan-Kosaraju decomposition $\mathcal{T}, \mathcal{L}$, that satisfy the properties from Section 9 for $\varkappa = 1/2$. (Another possibility, is to switch from Euclidean norms to ℓ_∞ norms, which may be computed exactly in our model of computation. Only the most trivial changes are needed, in order to transform the arguments above to suit the ℓ_∞ metric.)

We recall the construction of our basic blobs, from Section 10.

We set $\Gamma_f(x, 0) = (\Gamma_f(x, 0, M))_{M>0}$, for all $x \in E$, where

$$\Gamma_f(x, 0, M) = \{P \in \mathcal{P} : |\partial^\alpha P(x)| \le M \text{ for } |\alpha| \le m-1, |P(x) - f(x)| \le M\sigma(x)\}.$$

Once we have constructed all the blobs $\Gamma_f(x, \ell)$ $(x \in E)$ for a fixed $\ell \ge 0$, we define the $\Gamma_f(x, \ell+1)$ as follows:

<u>Step 1</u>: For $A \in T$, we define $\Gamma_f(A, \ell) = \bigcap_{x \in A} \{\Gamma_f(x, \ell) + \mathcal{B}(x, \mathsf{diam}_\infty(A))\}$.

<u>Step 2</u>: For $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ and $i = 1, 2$, define

$$\Gamma_{f,i}(\Lambda_1, \Lambda_2, \ell) = \bigcap_{A \in \Lambda_i} \{\Gamma_f(A, \ell) + \mathcal{B}(x_A, \mathsf{diam}_\infty(\cup \Lambda_i))\}.$$

<u>Step 3</u>: For $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, define

$$\bar{\Gamma}_f(\Lambda_1, \Lambda_2, \ell) = \Gamma_{f,1}(\Lambda_1, \ell) \cap \{\Gamma_{f,2}(\Lambda_2, \ell) + \mathcal{B}(x_{\Lambda_1}, |x_{\Lambda_1} - x_{\Lambda_2}|_{\ell_\infty})\}.$$

$\underline{\mathsf{Step\ 4}}$: For $A \in T$, define

$$\bar{\Gamma}_f(A,\ell) = \bigcap_{\substack{(\wedge_1,\wedge_2)\in\mathcal{L} \\ \wedge_1 \ni A}} \bar{\Gamma}_f(\wedge_1,\wedge_2,\ell)\,.$$

$\underline{\mathsf{Step\ 5}}$: For $x \in E$, define

$$\Gamma_f(x,\ell+1) = \Gamma_f(x,\ell) \cap \bigcap_{\substack{A\in T \\ A\ni x}} \bar{\Gamma}_f(A,\ell)\,.$$

(Here, Step 3 differs trivially from its analogue in Section 10.)

We recall some of the notation from [19] (and from Section 9) that we used in the above five steps. For each $A \in T$, x_A is a "representative", satisfying $x_A \in A$. For each $(\wedge_1,\wedge_2) \in \mathcal{L}$ and $i = 1, 2$, we write $\cup\wedge_i$ for the union $\cup_{A\in\wedge_i} A$, and we take $x_{\wedge_i}$ to be a "representative", i.e., an element of $\cup\wedge_i$. We write $|x-y|_{\ell_\infty}$ to denote $\max_{1\le k\le n} |x_k - y_k|$, where $x = (x_1,\dots,x_n)$ and $y = (y_1,\dots,y_n)$ are two given points in $\mathbb{R}^n$.

We will produce $C_\ell S_0$-bit MALPs $\mathcal{A}(x,\ell)$ in $\mathbb{R}^D$, with constants C'_ℓ, such that the blobs $\mathcal{K}(\mathcal{A}(x,\ell))$ arising from $\mathcal{A}(x,\ell)$ will be C''_ℓ-equivalent to $\Gamma_f(x,\ell)$ above $C'''_\ell \cdot 2^{-S_0}$, for $\ell = 0,\dots,\ell_*$. Here, $C_\ell, C'_\ell,\dots$ denote constants depending only on m, n, ℓ. Since we will always have $0 \le \ell \le \ell_*$, this means that eventually we could treat C_ℓ, C'_ℓ etc. as constants depending only on m and n. Recall that $\mathcal{P}$ is identified with $\mathbb{R}^D$ by identifying $P \in \mathcal{P}$ with $(\partial^\alpha P(0))_{|\alpha|\le m-1} \in \mathbb{R}^D$.

Using our work on Fault-Tolerant ALPs, we can easily mimic the above five-step inductive procedure.

In fact, for $\ell = 0$, we create a CS_0-bit MALP $\mathcal{A}(x,0)$ with constant C', for each $x \in E$, such that $\mathcal{K}(\mathcal{A}(x,0))$ is C''-equivalent to $\Gamma_f(x,0)$ above 2^{-S_0}. The construction of such MALPs was carried out in Section 48.

Next, suppose that, for some given $0 \le \ell \le \ell_* - 1$, we have already computed $C_\ell S_0$-bit MALPs $\mathcal{A}(x,\ell)$ (for $x \in E$), with constant C'_ℓ, such that the blob $\mathcal{K}(\mathcal{A}(x,\ell))$ is C''_ℓ-equivalent to $\Gamma_f(x,\ell)$ above $C'''_\ell \cdot 2^{-S_0}$.

Then we proceed as follows.

$\underline{\mathsf{Step\ 1'}}$: For each $A \in T$, we compute a $\tilde{C}_\ell S_0$-bit MALP $\mathcal{A}(A,\ell)$ with constant $\tilde{C}'_\ell$, such that $\mathcal{K}(\mathcal{A}(A,\ell))$ is $\tilde{C}''_\ell$-equivalent to the intersection over $x \in A$ of $\mathcal{K}(\mathcal{A}(x,\ell)) + \mathcal{B}(x, \mathsf{diam}_\infty(A))$ above $\tilde{C}'''_\ell \cdot 2^{-S_0}$.

This can be done, thanks to Algorithm PFM1 from Section 48 and to Algorithms MALP6 and MALP7 from Section 43. Note that we may invoke those algorithms, thanks to our assumptions (SU1) and (SU2). Indeed, in order to apply Algorithm PFM1, Algorithm MALP6 and Algorithm MALP7, we need to make sure that $\bar{S} \geq (C^{**}_D)^2(C_\ell + C)S_0$ and $S_0 \geq (C^{**}_D + C)$ where $D = \dim \mathcal{P}$ is a constant depending only on m and n. That holds in view of our assumptions (SU1), (SU2) and $0 \leq \ell \leq \ell_* - 1$.

Step 2′: For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ and $i = 1, 2$, we compute a $\tilde{\tilde{C}}_\ell S_0$-bit MALP $\mathcal{A}_i(\Lambda_1, \Lambda_2, \ell)$ with constant $\tilde{\tilde{C}}'_\ell$, such that $\mathcal{K}(\mathcal{A}_i(\Lambda_1, \Lambda_2, \ell))$ is $\tilde{\tilde{C}}''_\ell$-equivalent to the intersection over $A \in \Lambda_i$ of $\mathcal{K}(\mathcal{A}_i(A, \ell)) + \mathcal{B}(x_A\,,\ \mathrm{diam}_\infty(\cup\Lambda_i))$ above $\tilde{\tilde{C}}'''_\ell \cdot 2^{-S_0}$.

Again, this can be done, thanks to Algorithm PFM1 from Section 48 and Algorithms MALP6 and MALP7 from Section 43. We may invoke those algorithms, as we have $\bar{S} \geq (C^{**}_D)^2(\tilde{C}_\ell + C)S_0$ and $S_0 \geq (C^{**}_D + C)$, in view of our assumptions (SU1), (SU2) and $0 \leq \ell \leq \ell_* - 1$.

Step 3′: For each $(\Lambda_1, \Lambda_2) \in \mathcal{L}$, we compute a $\hat{C}_\ell S_0$-bit MALP $\bar{\mathcal{A}}(\Lambda_1, \Lambda_2, \ell)$ with constant $\hat{C}'_\ell$, such that the blob $\mathcal{K}(\bar{\mathcal{A}}(\Lambda_1, \Lambda_2, \ell))$ is $\hat{C}''_\ell$-equivalent to

$$\mathcal{K}(\mathcal{A}_1(\Lambda_1, \Lambda_2, \ell)) \cap \{\mathcal{K}(\mathcal{A}_2(\Lambda_1, \Lambda_2, \ell)) + \mathcal{B}(x_{\Lambda_1}, |x_{\Lambda_1} - x_{\Lambda_2}|_{\ell^\infty})\}$$

above $\hat{C}'''_\ell \cdot 2^{-S_0}$. Again, this can be carried out, thanks to Algorithm PFM1 and Algorithms MALP6 and MALP7, with the help of the assumptions (SU1), (SU2) and $0 \leq \ell \leq \ell_* - 1$.

Step 4′: For each $A \in T$, we compute a $\hat{\hat{C}}_\ell S_0$-bit MALP $\bar{\mathcal{A}}(A, \ell)$ with constant $\hat{\hat{C}}'_\ell$, such that the blob $\mathcal{K}(\bar{\mathcal{A}}(A, \ell))$ is $\hat{\hat{C}}''_\ell$-equivalent to the intersection over all $(\Lambda_1, \Lambda_2) \in \mathcal{L}$ with $\Lambda_1 \ni A$ of $\mathcal{K}(\bar{A}(\Lambda_1, \Lambda_2, \ell))$, above $\hat{\hat{C}}'''_\ell \cdot 2^{-S_0}$. Again, this can be carried out, thanks to Algorithm MALP7, with the help of (SU1), (SU2) and the fact that $0 \leq \ell \leq \ell_*$.

Step 5′: For each $x \in E$, we compute a $\check{C}'_\ell S_0$-bit MALP $\mathcal{A}(x, \ell + 1)$ with constant $\check{C}'_\ell$, such that the blob $\mathcal{K}(\mathcal{A}(x, \ell + 1))$ is $\check{C}''_\ell$-equivalent to the intersection of $\mathcal{K}(\mathcal{A}(x, \ell))$ with all $\mathcal{K}(\bar{\mathcal{A}}(A, \ell))$ such that $A \ni x$, above $\check{C}'''_\ell \cdot 2^{-S_0}$. Again, this can be carried out, thanks to Algorithm MALP7, and to (SU1), (SU2) and the fact that $0 \leq \ell \leq \ell_*$.

Thus, we can compute a $C_\ell S_0$-bit MALP $\mathcal{A}(x, \ell)$, for $x \in E$ and $0 \leq \ell \leq \ell_*$, with constant C'_ℓ, such that the blob $\mathcal{K}(\mathcal{A}(x, \ell))$ is C''_ℓ-equivalent to $\Gamma_f(x, \ell)$ above $C'''_\ell \cdot 2^{-S_0}$. The $\mathcal{A}(x, \ell)$ all have length $D = \dim \mathcal{P}$.

One shows, as in the corresponding section with ALPs, that the work of creating all the $\mathcal{A}(x,\ell)$, as above, is at most $CN\log N$, using storage at most CN.

Next, we discuss the convex sets $\sigma(x,\ell)$ $(x \in E, 0 \le \ell \le \ell_*)$.

Recall that, when we replace f by zero, then in place of $\Gamma_f(x,\ell)$ we obtain homogeneous blobs $\Gamma_0(x,\ell)$, having the form $\Gamma_0(x,\ell) = (M\sigma(x,\ell))_{M>0}$ for (non-empty, symmetric) convex sets $\sigma(x,\ell) \subset \mathcal{P}$. Carrying out our computation of the MALPs $\mathcal{A}(x,\ell)$ $(x \in E, 0 \le \ell \le \ell_*)$ as above, with f replaced by zero, we obtain MALPs $\mathcal{A}_0(x,\ell)$ $(x \in E, 0 \le \ell \le \ell_*)$, such that:

- $\mathcal{A}_0(x,\ell)$ is a $C_\ell S_0$-bit MALP with constant C'_ℓ; and
- $\mathcal{K}(\mathcal{A}_0(x,\ell))$ is C''_ℓ-equivalent to $\Gamma_0(x,\ell)$ above $C'''_\ell \cdot 2^{-S_0}$.

Let

$$\mathcal{A}_0(x,\ell) = \left[(\lambda^{[\ell]}_{i\alpha})_{\substack{1\le i\le D\\ |\alpha|\le m-1}}, (b^{[\ell]}_i)_{1\le i\le D}, (\sigma^{[\ell]}_i)_{1\le i\le D}, M^{[\ell]}_*\right].$$

By induction on ℓ, one checks easily that the $b^{[\ell]}_i$ are all equal to zero. Hence, $\mathcal{K}(\mathcal{A}_0(x,\ell)) = (K_M(\mathcal{A}_0(x,\ell)))_{M>0}$, with

$$K_M(\mathcal{A}_0(x,\ell)) = \left\{P \in \mathcal{P} : \left|\sum_{|\alpha|\le m-1} \lambda^{[\ell]}_{i\alpha}\partial^\alpha P(0)\right| \le M\sigma^{[\ell]}_i\right\} \text{ if } M \ge M^{[\ell]}_*;$$

$$K_M(\mathcal{A}_0(x,\ell)) = \emptyset \text{ if } M < M^{[\ell]}_*.$$

Since $\mathcal{K}(\mathcal{A}_0(x,\ell))$ is C''_ℓ-equivalent to $(M\sigma(x,\ell))_{M>0}$ above $C'''_\ell \cdot 2^{-S_0}$, it follows that $M^{[\ell]}_* \le C''''_\ell \cdot 2^{-S_0}$, and that $\sigma(x,\ell)$ is $C^\#_\ell$-equivalent to

$$\left\{P \in \mathcal{P} : \left|\sum_{|\alpha|\le m-1} \lambda^{[\ell]}_{i\alpha}\partial^\alpha P(0)\right| \le \sigma^{[\ell]}_i \text{ for } i = 1,\dots,D\right\}.$$

Hence, we have computed the $\Gamma_f(x,\ell)$ up to C-equivalence above $C \cdot 2^{-S_0}$; and we have computed the $\sigma(x,\ell)$ up to C-equivalence.

The computations of the approximation to the $\Gamma_f(x,\ell)$ and $\sigma(x,\ell)$, as above, require a total work of $CN\log N$, using storage CN. It will be convenient also to rewrite the blobs $\mathcal{K}(\mathcal{A}_0(x,\ell))$ in a different form, as follows. For $x \in E$, let $T_x : \mathbb{R}^D \longrightarrow \mathbb{R}^D$ be the linear map that takes $(\partial^\alpha P(0))_{|\alpha|\le m-1}$ to $(\partial^\alpha P(x))_{|\alpha|\le m-1}$ for each $P \in \mathcal{P}$. Then $\| T_x \|, \| T_x^{-1} \| \le 2^{CS_0}$, since $|x| \le C2^{S_0}$. Using Algorithm MALP4 from Section 43, and thanks to (SU1) and (SU2), we may compute from $\mathcal{A}_0(x,\ell)$, a CS_0-bit MALP $\tilde{\mathcal{A}}_0(x,\ell)$ with constant C, such that $\mathcal{K}(\tilde{\mathcal{A}}_0(x,\ell))$ is C-equivalent to $T_x\mathcal{K}(\mathcal{A}_0(x,\ell))$. Hence,

(∗) $\mathcal{K}(\tilde{\mathcal{A}}_0(x,\ell))$ is C-equivalent to $(\{(\partial^\alpha P(x))_{|\alpha|\le m-1} : P \in M\sigma(x,\ell)\})_{M>0}$ above $C \cdot 2^{-S_0}$.

Here, C depends only on m and n, since we take $0 \le \ell \le \ell_*$. Note that $\tilde{\mathcal{A}}_0(x,\ell)$ has length D.

Let

$$\tilde{\mathcal{A}}_0(x,\ell) = [(\tilde{\lambda}^{\ell,x}_{i,\alpha})_{\substack{1\le i\le D\\ |\alpha|\le m-1}}, (\tilde{b}^{\ell,x}_i)_{1\le i\le D}, (\tilde{\sigma}^{\ell,x}_i)_{1\le i\le D}, \tilde{M}^{\ell,x}_*].$$

Since the b^ℓ_i appearing in $\mathcal{A}_0(x,\ell)$ are all zero, it follows that the $\tilde{b}^{\ell,x}_i$ are all zero. Hence, applying $(*)$ with $M = 1$ (note that $\tilde{M}^{\ell,x}_* \le C2^{-S_0} \le 1$ from (*)), we find that, for $x \in E$ and $0 \le \ell \le \ell_*$,

$$(**) \quad \left[\begin{array}{l} \{(\partial^\alpha P(x))_{|\alpha|\le m-1} : P \in \sigma(x,\ell)\} \text{ is } C\text{-equivalent to} \\ \left\{ (\xi^\alpha)_{|\alpha|\le m-1} \in \mathbb{R}^D : \left| \sum_{|\alpha|\le m-1} \tilde{\lambda}^{\ell,x}_{i,\alpha}\, \xi^\alpha \right| \le \tilde{\sigma}^{\ell,x}_i \text{ for } i = 1,\dots,D \right\}. \end{array}\right]$$

Here, C depends only on m and n. The $\tilde{\lambda}^{\ell,x}_{i,\alpha}$, $\tilde{\sigma}^{\ell,x}_i$ may be computed, for all $x \in E$, $0 \le \ell \le \ell_*$, with total work at most $CN \log N$, and with storage at most CN.

51. The Basic Lengthscales

Here, $c, C, \dots$ stand for constants depending only on m, n. Recall that we assume (SU1) and (SU2) from Section 49.

Recall that $\sigma(x) \ge 2^{-S_0}$ for each $x \in E$, and that $|x - y| \ge 2^{-S_0}$ for $x, y \in E$, $x \ne y$. Hence, given any $x^0 \in E$ and any $(\xi^\alpha)_{|\alpha|\le m-1} \in \mathbb{R}^D$, with $|\xi^\alpha| \le 2^{-(m-|\alpha|)S_0}$ for each α, there exists $F \in C^m(\mathbb{R}^D)$, with $\| F \|_{C^m(\mathbb{R}^n)} \le C$, $|F(x)| \le \sigma(x)$ for each $x \in E$, and $\partial^\alpha F(x^0) = \xi^\alpha$ for $|\alpha| \le m-1$. Therefore, Property 0 in Section 13 tells us that

$$(1) \quad \{(\xi^\alpha)_{|\alpha|\le m-1} : |\xi^\alpha| \le 2^{-(m-|\alpha|)S_0} \text{ for each } \alpha\}$$

$$\subset C\{(\partial^\alpha P(x^0))_{|\alpha|\le m-1} : P \in \sigma(x^0,\ell)\}$$

for each $x^0 \in E$, $0 \le \ell \le \ell_*$.

On the other hand, we have $\sigma(x^0,\ell) \subseteq C\sigma(x^0,0)$; and $|\partial^\alpha P(x^0)| \le C$ for each α, whenever $P \in \sigma(x^0,0)$. Hence,

(2) $\{(\partial^\alpha P(x^0))_{|\alpha|\le m-1} : P \in \sigma(x^0,\ell)\} \subset C\{(\xi^\alpha)_{|\alpha|\le m-1} : \forall\alpha,\ |\xi^\alpha| \le 1\}$ for each $x^0 \in E$, $0 \le \ell \le \ell_*$.

In the previous section, we computed numbers $\tilde{\lambda}^{\ell,x^0}_{i,\alpha}$ and $\tilde{\sigma}^{\ell,x^0}_i$, such that

$$
\begin{aligned}
(3) \qquad & c\left\{(\xi^\alpha)_{|\alpha|\le m-1} : \left|\sum_{|\alpha|\le m-1} \tilde\lambda^{\ell,x^0}_{i,\alpha}\,\xi^\alpha\right| \le \tilde\sigma^{\ell,x^0}_i \text{ for } i=1,\dots,D\right\}\\
& \subseteq \left\{(\partial^\alpha P(x^0))_{|\alpha|\le m-1} : P\in\sigma(x^0,\ell)\right\}\\
& \subseteq C\cdot\left\{(\xi^\alpha)_{|\alpha|\le m-1} : \left|\sum_{|\alpha|\le m-1} \tilde\lambda^{\ell,x^0}_{i,\alpha}\,\xi^\alpha\right| \le \tilde\sigma^{\ell,x^0}_i \text{ for } i=1,\dots,D\right\}.
\end{aligned}
$$

Recall the constants $A_0, A_1(\mathcal{A}), A_2$ from Section 17, and $\ell(\mathcal{A})$ from Section 14. The goal of this section is to present the following algorithm.

Algorithm BL1: Given $x^0 \in E$, and non-empty $\mathcal{A} \subseteq \mathcal{M}$, we compute a machine number $\delta(x^0,\mathcal{A}) > 0$, with the following properties.

(OK1) There exist polynomials $P_\alpha \in \mathcal{P}$, indexed by $\alpha \in \mathcal{A}$, such that $\partial^\beta P_\alpha(x^0) = \delta_{\beta\alpha}$ for $\beta,\alpha \in \mathcal{A}$; and

$$|\partial^\beta P_\alpha(x^0)| \le CA_1(\mathcal{A})\cdot(A_2\delta(x^0,\mathcal{A}))^{|\alpha|-|\beta|} \text{ for } \alpha\in\mathcal{A},\ \beta\in\mathcal{M},\ \beta\ge\alpha;$$

$$(A_2\delta(x^0,\mathcal{A}))^{m-|\alpha|}P_\alpha \in CA_1(\mathcal{A})\cdot\sigma(x^0,\ell(\mathcal{A})) \text{ for } \alpha\in\mathcal{A}.$$

(OK2) There do not exist polynomials $P_\alpha \in \mathcal{P}$, indexed by $\alpha \in \mathcal{A}$, such that $\partial^\beta P_\alpha(x^0) = \delta_{\beta\alpha}$ for $\beta,\alpha\in\mathcal{A}$; and

$$|\partial^\beta P_\alpha(x^0)| \le cA_1(\mathcal{A})\cdot(A_2\delta(x^0,\mathcal{A}))^{|\alpha|-|\beta|} \text{ for } \alpha\in\mathcal{A},\ \beta\in\mathcal{M},\ \beta\ge\alpha;$$

$$(A_2\delta(x^0,\mathcal{A}))^{m-|\alpha|}P_\alpha \in c\,A_1(\mathcal{A})\cdot\sigma(x^0,\ell(\mathcal{A})) \text{ for } \alpha\in\mathcal{A}.$$

Explanation: First, suppose $\mathcal{A}\ne\mathcal{M}$. It is enough to find $\delta(x^0,\mathcal{A})$ as above, with $\sigma(x^0,\ell(\mathcal{A}))$ replaced by a convex, symmetric polyhedron $\tilde\sigma$ that is C-equivalent to $\sigma(x^0,\ell(\mathcal{A}))$.
In view of (3), we can take

$$(4) \qquad \tilde\sigma = \{P\in\mathcal{P} : |\sum_{\beta\in\mathcal{M}} \tilde\lambda_{i,\beta}\,\partial^\beta P(x^0)| \le \tilde\sigma_i \text{ for } i = 1,\dots,D\},$$

where the $\tilde\lambda_{i,\beta} = \tilde\lambda^{\ell(\mathcal{A}),x^0}_{i,\beta}$, $\tilde\sigma_i = \tilde\sigma^{\ell(\mathcal{A}),x^0}_i$ have already been computed.

We set $\xi_{\alpha,\beta} = \partial^\beta P_\alpha(x^0)$ for $\alpha\in\mathcal{A}$, $\beta\in\mathcal{M}\smallsetminus\mathcal{A}$, so that the desired property of $\delta(x^0,\mathcal{A})$ may be rewritten as follows. We seek $\delta\in(2^{-CS_0},2^{+CS_0})$ that satisfies:

(OK1)′ There exist $(\xi_{\alpha,\beta})_{\alpha\in\mathcal{A},\beta\in\mathcal{M}\smallsetminus\mathcal{A}}$, such that

$$|\xi_{\alpha,\beta}| \le CA_1(\mathcal{A})\cdot\delta^{|\alpha|-|\beta|} \text{ for } \alpha\in\mathcal{A},\ \beta>\alpha, \quad \text{and}$$

$$\left|\tilde\lambda_{i,\alpha} + \sum_{\beta\in\mathcal{M}\smallsetminus\mathcal{A}} \tilde\lambda_{i,\beta}\,\xi_{\alpha,\beta}\right| \le CA_1(\mathcal{A})\cdot\tilde\sigma_i\delta^{|\alpha|-m} \text{ for } i=1,\dots,D \text{ and } \alpha\in\mathcal{A}.$$

(OK2)$'$ There does not exist $(\xi_{\alpha,\beta})_{\alpha\in\mathcal{A},\beta\in\mathcal{M}\setminus\mathcal{A}}$, such that

$$|\xi_{\alpha,\beta}| \le cA_1(\mathcal{A}) \cdot \delta^{|\alpha|-|\beta|} \text{ for } \alpha\in\mathcal{A},\ \beta>\alpha\,, \quad \text{and}$$
$$\left|\tilde{\lambda}_{i,\alpha} + \sum_{\beta\in\mathcal{M}\setminus\mathcal{A}} \tilde{\lambda}_{i,\beta}\,\xi_{\alpha,\beta}\right| \le cA_1(\mathcal{A}) \cdot \tilde{\sigma}_i\delta^{|\alpha|-m} \text{ for } i=1,\dots,D \text{ and } \alpha\in\mathcal{A}.$$

Once we have found $\delta \in (2^{-CS_0}, 2^{+CS_0})$ satisfying (OK1)$'$, (OK2)$'$, we can then set $\delta(x^0,\mathcal{A}) =$ (our machine approximation to) δ/A_2. The percentage error in dividing δ by A_2 will be small, by (SU1) and (SU2) from Section 49, and therefore $A_2\delta(x^0,\mathcal{A}) \in [\frac{1}{2}\delta, 2\delta]$. Consequently, $\delta(x^0,\mathcal{A})$ satisfies (OK1), (OK2), since δ satisfies (OK1)$'$, (OK2)$'$. Thus, it is enough to find $\delta \in (2^{-CS_0}, 2^{+CS_0})$ satisfying (OK1)$'$, (OK2)$'$.

Algorithm SIP1 from Section 46 produces one of the following outcomes. (Again, we use (SU1) and (SU2) in order to apply Algorithm SIP1.)

Outcome 1: We guarantee that, for each $\delta \in (2^{-CS_0}, 2^{+CS_0})$, (OK1)$'$ holds.

Outcome 2: We guarantee that, for each $\delta \in (2^{-CS_0}, 2^{+CS_0})$, (OK2)$'$ holds.

Outcome 3: We have computed $\delta \in (2^{-CS_0}, 2^{+CS_0})$ satisfying (OK1)$'$, (OK2)$'$.

We will check that Outcomes 1 and 2 cannot occur here. This will complete our specification of Algorithm BL1.

Recall that $\sigma(x^0, \ell(\mathcal{A}))$ and $\tilde{\sigma}$ in (4) are C-equivalent. Hence (1) and (2) imply the following, for vectors $(\hat{\xi}_{\alpha,\beta})_{\alpha\in\mathcal{A}\,,\,\beta\in\mathcal{M}}$.

$$\text{(5)}\qquad \left[\begin{array}{l} |\hat{\xi}_{\alpha,\beta}| \le 2^{-(m-|\beta|)S_0} \text{ for } \alpha\in\mathcal{A},\ \beta\in\mathcal{M} \text{ implies} \\ \left|\sum_{\beta\in\mathcal{M}} \tilde{\lambda}_{i,\beta}\,\hat{\xi}_{\alpha,\beta}\right| \le C\tilde{\sigma}_i \text{ for } i=1,\dots,D\,,\ \alpha\in\mathcal{A}\,. \end{array}\right]$$

$$\text{(6)}\qquad \left[\begin{array}{l} \left|\sum_{\beta\in\mathcal{M}} \tilde{\lambda}_{i,\beta}\,\hat{\xi}_{\alpha,\beta}\right| \le \tilde{\sigma}_i \text{ for } i=1,\dots D \text{ and } \alpha\in\mathcal{A} \text{ implies} \\ |\hat{\xi}_{\alpha,\beta}| \le C \text{ for } \alpha\in\mathcal{A}, \beta\in\mathcal{M}\,. \end{array}\right]$$

Suppose (OK1)$'$ holds for a given $\delta \in (2^{-CS_0}, 2^{+CS_0})$. Let $(\xi_{\alpha,\beta})_{\alpha\in\mathcal{A},\,\beta\in\mathcal{M}\setminus\mathcal{A}}$ be as in (OK1)$'$, and let

$$\hat{\xi}_{\alpha,\beta} = [CA_1(\mathcal{A})]^{-1}\,\delta^{m-|\alpha|}\,\xi_{\alpha,\beta} \text{ for } \alpha\in\mathcal{A},\ \beta\in\mathcal{M}\setminus\mathcal{A}$$
$$\hat{\xi}_{\alpha,\beta} = [CA_1(\mathcal{A})]^{-1}\,\delta^{m-|\alpha|}\,\delta_{\alpha,\beta} \text{ for } \alpha\in\mathcal{A},\ \beta\in\mathcal{A}\,.$$

Here, we take C as in (OK1)$'$. Then, for $\alpha\in\mathcal{A}$,

$$\Big|\sum_{\beta\in\mathcal{M}} \tilde\lambda_{i,\beta}\,\hat\xi_{\alpha,\beta}\Big| = [CA_1(\mathcal{A})]^{-1}\,\delta^{m-|\alpha|}\Big|\sum_{\beta\in\mathcal{M}\smallsetminus\mathcal{A}}\tilde\lambda_{i,\beta}\,\xi_{\alpha,\beta} + \tilde\lambda_{i,\alpha}\Big| \le \tilde\sigma_i$$

for each $i = 1,\dots,D$ and $\alpha \in \mathcal{A}$. Hence, (6) yields $|\hat\xi_{\alpha,\alpha}| \le C'$ for each $\alpha \in \mathcal{A}$, i.e., $[CA_1(\mathcal{A})]^{-1}\ \delta^{m-|\alpha|} \le C'$ for each $\alpha \in \mathcal{A}$. This cannot hold for arbitrary $\delta \in [2^{-CS_0}, 2^{+CS_0}]$, provided S_0 is sufficiently large. (Here, we use (SU1) from Section 49.) Therefore, Outcome 1 is impossible here.

Next, suppose we are given $\delta \in (2^{-CS_0},\, 2^{-2S_0})$.

Let

$$\hat\xi_{\alpha,\beta} = (\hat C)^{-1}2^{-(m-|\alpha|)S_0}\ \cdot\ \delta_{\alpha,\beta}$$

for $\alpha \in \mathcal{A}$, $\beta \in \mathcal{M}$, with $\hat C$ large enough. Then by (5), we have

$$(7)\quad (\hat C)^{-1}\cdot 2^{-(m-|\alpha|)S_0}\cdot|\tilde\lambda_{i,\alpha}| = \Big|\sum_{\beta\in\mathcal{M}}\tilde\lambda_{i,\beta}\,\hat\xi_{\alpha,\beta}\Big| \le \tilde\sigma_i \text{ for } i = 1,\dots,D \text{ and } \alpha \in \mathcal{A}.$$

Since $\delta \le 2^{-2S_0}$ then $\delta^{|\alpha|-m} \ge 2^{2(m-|\alpha|)S_0}$ for $\alpha \in \mathcal{A}$, hence, with c as in (OK2)$'$,

$$(8)\quad cA_1(\mathcal{A})\cdot\tilde\sigma_i\cdot\delta^{|\alpha|-m} \ge [cA_1(\mathcal{A})\cdot 2^{(m-|\alpha|)S_0}]\,2^{(m-|\alpha|)S_0}\,\tilde\sigma_i \ge \hat C\cdot 2^{(m-|\alpha|)S_0}\tilde\sigma_i,$$

provided S_0 is sufficiently large (as in (SU1)). Taking now $\xi_{\alpha,\beta} = 0$ for $\alpha \in \mathcal{A}$, $\beta \in \mathcal{M}\smallsetminus\mathcal{A}$, we obtain

$$|\xi_{\alpha,\beta}| = 0 \le cA_1(\mathcal{A})\ \cdot\ \delta^{|\alpha|-|\beta|} \text{ for } \alpha \in \mathcal{A},\ \beta \in \mathcal{M}\smallsetminus\mathcal{A},\ \beta > \alpha;$$

and by (7) and (8),

$$\begin{aligned}\Big|\sum_{\beta\in\mathcal{M}\smallsetminus\mathcal{A}}\tilde\lambda_{i,\beta}\,\xi_{\alpha,\beta} + \tilde\lambda_{i,\alpha}\Big| &= |\tilde\lambda_{i,\alpha}| \le \hat C\ \cdot\ 2^{(m-|\alpha|)S_0}\tilde\sigma_i\\ &\le cA_1(\mathcal{A})\ \cdot\ \tilde\sigma_i\ \cdot\ \delta^{|\alpha|-m},\ \text{with } c \text{ as in (OK2)}'.\end{aligned}$$

In particular, there does exist $(\xi_{\alpha,\beta})_{\alpha\in\mathcal{A},\beta\in\mathcal{M}\smallsetminus\mathcal{A}}$ satisfying the inequalities in (OK2)$'$. Thus (OK2)$'$ cannot hold when $\delta \in [2^{-CS_0}, 2^{-2S_0}]$. Consequently, Outcome 2 cannot occur here.

Therefore, only Outcome 3 can occur.

This completes our specification of Algorithm BL1, in the case $\mathcal{A} \ne \mathcal{M}$. The case $\mathcal{A} = \mathcal{M}$ is an easier variant of the above, in which the variables $\xi_{\alpha,\beta}$ $(\alpha \in \mathcal{A}, \beta \in \mathcal{M}\setminus\mathcal{A})$ do not arise. Details are left to the reader.
The work of the algorithm is at most C, for a given $x^0 \in E$, once we have computed the ALPs $\tilde{\mathcal{A}}_0(x^0,\ell)$ $(0 \le \ell \le \ell_*)$ as in Section 50.

Hence, the total work to compute all the $\delta(x,\mathcal{A})$ (all $x \in E$, $\mathcal{A} \subseteq \mathcal{M}$, $\mathcal{A} \ne \emptyset$), given the $\tilde{\mathcal{A}}_0(x,\ell)$ (all $x \in E$, $0 \le \ell \le \ell_*$), is at most CN.

We define $\delta(x,\mathcal{A}) := +\infty$ for $\mathcal{A} = \emptyset$, since the inequalities in (OK1) can always be satisfied vacuously.

52. Dyadic Cubes and Cuboids

Recall the Calderón-Zygmund decompositions from Chapter III. We make use of terminology from that chapter.

All the algorithms from Sections 23,...,27 involving the BBD Tree can be carried out in our model of computation, without error. Indeed, we may decide, with perfect precision, whether a given point $x \in \mathbb{R}^n$ whose coordinates are machine numbers, belongs to any given dyadic cube or cuboid. (Here, we assume that the coordinates of the vertices of the cuboid are machine numbers, i.e., the cuboid is not absurdly large, absurdly small, or absurdly far away.) Thus, it is straightforward to verify that the construction of the BBD Tree in [1] may be carried out, without error.

Similarly, all the algorithms in Section 25 and Section 27 carry over to our model of computation, without any precision issues arising. Those algorithms are all based on the BBD Tree, and they do not use imperfect arithmetic operations at all (they use only comparisons, to detect whether a point belongs to a dyadic cuboid and to perform tasks of similar nature). Regarding the approximate nearest neighbor algorithm from [1] mentioned in Section 23: Let $x, y \in \mathbb{R}^n$ be given points, all of whose coordinates are machine numbers. No errors arise when we compute $|x - y|_{\ell_\infty}$. Note that the algorithms in [1], as quoted in Section 23, are well adapted to the ℓ_∞ metric (see [1, Theorem 1]). Thus, if we confine ourselves to the ℓ_∞ metric, in place of the Euclidean metric, no numerical errors arise when computing nearest neighbors. It is straightforward to verify that switching from the Euclidean norm to the ℓ_∞ norm causes only the most obvious changes to the arguments in Sections 22,...,26. (See the remark in Section 26.) Thus, the BBD Tree algorithms are well suited for our model of computation.

We take the constant c_G to be a (negative) integer power of 2, say $c_G = 1/32$. That way, if Q is a dyadic cube, then $(1 + c_G)Q$ will be a union of at most C dyadic cubes, with C depending only on n. Hence, given $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ (with each x_i a machine number of absolute value at most $2^{\bar{S}/2}$), and given a dyadic cube Q whose vertices have coordinates that are all machine numbers, we can decide whether $x \in (1 + c_G)Q$ with work C (depending only on n). Similarly, we can decide with work C whether $x \in (1 + c_G/2)Q$.

In particular, for each dyadic cube Q and $\mathcal{A} \subseteq \mathcal{M}$, we may detect whether $Q \in CZ(\mathcal{A})$, provided Q has vertices whose coordinates are machine numbers, and whether Q contains a point $x = (x_1, \dots, x_n)$, where each x_j is a machine number of absolute value at most $2^{\bar{S}/2}$. Also, for any $x = (x_1, \dots, x_n)$ such that each x_i is a machine number with $|x_i| \leq 2^{\bar{S}/2}$ for all i, and for each $\mathcal{A} \subseteq \mathcal{M}$, we can compute the list of all cubes $Q \in CZ(\mathcal{A})$ such that

$(1+c_G)Q \ni x$. The work of the computation is at most $C \log N$, with C depending only on m and n. (Of course, we assume here that we have already done the relevant one-time work; see Sections 22,...,26.)

53. Finding Neighbors

In this section, we describe a straightforward application of Algorithm MALP10 from Section 43. Here C, C', etc, denote constants determined by m and n.

Algorithm Find-Neighbor $(P_0, \mathcal{A}_0, x)$

/* Inputs: $P_0 \in \mathcal{P}$, $\mathcal{A}_0 \subset \mathcal{M}$ such that $\mathcal{A}_0 \neq \mathcal{M}$, $x \in E$, and we assume that $\partial^\alpha P_0(0)$ is a given machine number of absolute value $\leq 2^{A_1(\mathcal{A}_0)S_0}$ for each α $(|\alpha| \leq m-1)$.

Outputs: $P_1 =$ Find-Neighbor $(P_0, \mathcal{A}_0, x)$ is a polynomial in $\mathcal{P}$. It is guaranteed that $\partial^\alpha P_1(0)$ is a machine number of absolute value at most $2^{A_1(\mathcal{A}_0^-)S_0}$ for each α $(|\alpha| \leq m-1)$. We compute the $\partial^\alpha P_1(0)$ for each $\alpha(|\alpha| \leq m-1)$. The polynomial P_1 is guaranteed to have the following property:

> Suppose $P \in \mathcal{P}$ and $M \geq 2^{-S_0}$ satisfy
> $\partial^\alpha(P - P_0)(x) = 0$ for all $\alpha \in \mathcal{A}_0$, and $P \in \Gamma(x, \ell(\mathcal{A}_0) - 1, M)$.
> Then we can express $P_1 = P_{main} + P_{err}$; with $P_{main}, P_{err} \in \mathcal{P}$;
> $\partial^\alpha(P_{main} - P_0)(x) = 0$ for all $\alpha \in \mathcal{A}_0$,
> and $P_{main} \in \Gamma(x, \ell(\mathcal{A}_0) - 1,\ C'M)$;
> and $|\partial^\alpha P_{err}(0)| \leq 2^{A_1(\mathcal{A}_0^-)S_0 - \bar{S}/2}$ for all $\alpha(|\alpha| \leq m-1)$.

Here, C' is some constant depending only on m and n.

We assume here that we have already done the one-time work to produce MALPs that are C-equivalent to the $\Gamma(x, \ell(\mathcal{A}) - 1)$ above $C2^{-S_0}$.

*/

Explanation: Recall from Section 50 that we are able to construct CS_0-bit MALPs with constant C that are C-equivalent to the $\Gamma(x, \ell(\mathcal{A}) - 1)$ above $C2^{-S_0}$, with C being a constant depending only on m and n. By one of the properties in Section 47, these MALPS are C^2-equivalent to the $\Gamma(x, \ell(\mathcal{A}) - 1)$ above 2^{-S_0}. A straightforward application of Algorithm MALP10 from Section 43 now leads to Algorithm Find-Neighbor. (Note that $A_1(\mathcal{A}_0^-) \geq A_0 A_1(\mathcal{A}_0)$ and that we may assume $A_0 \geq C$ for an appropriate constant C depending only on m and n.) The work of the Algorithm, not including one-time work used by the algorithm, is at most a constant determined by m and n.

By using Algorithm MALP8 we can also perform the following task. Given $x \in E$, we compute a polynomial $P_1 \in \mathcal{P}$, such that $\partial^\alpha P_1(0)$ is a machine number of absolute value at most $2^{C'S_0}$ for each α $(|\alpha| \leq m-1)$, where C' is a constant depending only on m and n. We compute the $\partial^\alpha P_1(0)$ for each α $(|\alpha| \leq m-1)$. The polynomial P_1 is guaranteed to have the following property:

$$\left[\begin{array}{l}\text{Suppose } P \in \mathcal{P} \text{ and } M \geq 2^{-S_0} \text{ satisfy } P \in \Gamma(x, \ell_*, M).\\ \text{Then we can express } P_1 = P_{main} + P_{err}; \text{ with } P_{main}, P_{err} \in \mathcal{P};\\ P_{main} \in \Gamma(x, \ell_*, C'M);\\ \text{and } |\partial^\alpha P_{err}(0)| \leq 2^{C''S_0 - \bar{S}/2} \text{ for all } \alpha \ (|\alpha| \leq m-1).\end{array}\right.$$

From (1) of Section 51, and from Property 1 of Section 13, we conclude that P_1 has the following property:

$$\left[\begin{array}{l}\text{Suppose } P \in \mathcal{P} \text{ and } M \geq 2^{-S_0} \text{ satisfy } P \in \Gamma(x, \ell_*, M).\\ \text{Then } P_1 \in \Gamma(x, \ell_*, \tilde{C}M).\end{array}\right.$$

The work needed to perform the latter task is bounded by C, given one-time work $CN \log N$ in space CN.

54. Partitions of Unity

Write c, C, C', etc., for constants depending only on m, n.

Recall that, for $\mathcal{A} \subseteq \mathcal{M}$ and $Q \in CZ(\mathcal{A})$, we define

$$(1) \qquad \theta^Q_{\mathcal{A}} = \theta^Q_0 \Big/ \sum_{Q' \in CZ(\mathcal{A})} \theta^{Q'}_0 .$$

Here, θ^Q_0 is supported in $(1 + c_G/2)Q$, satisfies $\theta^Q_0 \geq 0$ everywhere, and $\theta^Q_0 \geq c$ on Q.

We have taken c_G to be a power of two, so that $(1 + c_G/2)Q$ is a union of at most C dyadic cubes, whenever Q is dyadic. Hence, we can decide with work C whether a given $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ belongs to $(1 + c_G/2)Q$ for a given dyadic cube Q. Here, we assume that $x_1, \ldots, x_n$ are machine numbers. Note that for $Q \in CZ(\mathcal{A})$,

$$(2) \qquad 1 \geq \delta_Q \geq c \cdot 2^{-S_0},$$

since $\delta_Q \leq A_2^{-1}$ for all $Q \in CZ(\mathcal{A})$, and since $|x - y| \geq c \cdot 2^{-S_0}$ for any two distinct points $x, y \in E$.

We may take

(3) $\theta_0^Q(x)=\prod_{j=1}^{n}\left[1-\left(\frac{2(x_j-x_j^Q)}{(1+c_G/2)\delta_Q}\right)^2\right]^{m+1}$ for $x=(x_1,...,x_n)\in(1+c_G/2)Q$,

(4) $\theta_0^Q(x)=0$ for $x\neq(1+c_G/2)Q$.

Here, $(x_1^Q,\dots,x_n^Q)$ is the center of Q. Note that

$$|\partial^\alpha\theta_0^Q(x)|\ \le\ C\delta_Q^{-|\alpha|}\le C\cdot 2^{CS_0},$$

thanks to (2), (3), (4). From (2), (3), (4), we see that $\partial^\alpha\theta_0^Q(x)$ may be computed up to an error at most $C\cdot 2^{CS_0-\bar{S}}$, where $|\alpha|\le m$ and $x=(x_1,\dots,x_n)$ with the x_j machine numbers of absolute value $\le 2^{S_0}$. Thus, we may compute $J_x^+(\theta_0^Q)$ up to roundoff errors.

From (4), we see that

$$J_x^+\Bigg(\sum_{Q'\in CZ(\mathcal{A})}\theta_0^{Q'}\Bigg)\ =\ \sum_{Q'\in \mathsf{Cloud}(x,\mathcal{A})}J_x^+(\theta_0^{Q'}),$$

where

$$\mathsf{Cloud}(x,\mathcal{A})=\{Q'\in CZ(\mathcal{A}):x\in(1+c_G)Q'\}.$$

Recall that we can compute $\mathsf{Cloud}\ (x,\mathcal{A})$ in time $C\log N$, given one-time work $CN\log N$ in space CN.

Fix $x=(x_1,\dots,x_n)$ with each x_j a machine number of absolute value $\le 2^{S_0}$. For the moment, we identify $\mathcal{P}^+$ with $\mathbb{R}^{D+}$ by identifying $P\in\mathcal{P}^+$ with $(\partial^\alpha P(x))_{|\alpha|\le m}$. We can write

$$\sum_{Q'\in\mathsf{Cloud}\ (x,\mathcal{A})}J_x^+(\theta_0^{Q'})\ =\ b_0+\hat{P},\ \text{with}\ \ b_0>c\ \ \text{and}\ \ \hat{P}\in\mathcal{P}^+\ \ \text{with}\ \ \hat{P}(x)=0.$$

We can compute b_0 and $\hat{P}\in\mathcal{P}^+\simeq\mathbb{R}^{D+}$ with an error at most $C\cdot 2^{CS_0-\bar{S}}$. We have $|b_0|$, $|\hat{P}|\le 2^{CS_0}$ and $b_0>c$. Note that, in the ring of m-jets at x, we have

$$\Bigg(\sum_{Q'\in\mathsf{Cloud}\ (x,\mathcal{A})}J_x^+(\theta_0^{Q'})\Bigg)^{-1}\ =\ (b_0)^{-1}\cdot\sum_{\ell=0}^{m}(-1)^\ell\left(\frac{\hat{P}}{b_0}\right)^\ell.$$

Hence,

$$\left|\Bigg(\sum_{Q'\in\mathsf{Cloud}\ (x,\mathcal{A})}J_x^+(\theta_0^{Q'})\Bigg)^{-1}\right|_{\ell^\infty}\ \le\ 2^{CS_0}\ \ \text{in}\ \mathbb{R}^{D^+},$$

and

$$\Bigg(\sum_{Q'\in\mathsf{Cloud}\ (x,\mathcal{A})}J_x^+(\theta_0^{Q'})\Bigg)^{-1}\ \in\mathbb{R}^{D+}$$

can be computed up to an error at most $2^{CS_0-\bar{S}}$ in time $C \log N$ given one-time work $CN \log N$ in space CN. Multiplying $J_x^+(\theta_0^Q)$ by

$$\Big(\sum_{Q' \in \mathsf{Cloud}\,(x,\mathcal{A})} J_x^+(\theta_0^{Q'})\Big)^{-1}$$

as jets at x, we see that we can carry out the following algorithm.

<u>Algorithm PU:</u> Given $\mathcal{A} \subseteq \mathcal{M}$, $Q \in CZ(\mathcal{A})$, and given $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, with each x_j a machine number of absolute value $\leq 2^{S_0}$, we compute machine numbers

$$\theta_{\text{approx}}^{Q,\mathcal{A},\alpha}(x) \quad (|\alpha| \leq m)$$

of absolute value $\leq 2^{CS_0}$, such that the polynomial

$$\{J_x^+(\theta_{\mathcal{A}}^Q)\}_{\text{approx}}(y) := \sum_{|\alpha| \leq m} \frac{1}{\alpha!}\,\theta_{\text{approx}}^{Q,\mathcal{A},\alpha}(x) \cdot (y - x)^\alpha$$

satisfies

$$\left|\partial^\alpha[\theta_{\mathcal{A}}^Q - \{J_x^+(\theta_{\mathcal{A}}^Q)\}_{\text{approx}}]\,(x)\right| \leq 2^{CS_0-\bar{S}} \quad \text{for } |\alpha| \leq m.$$

Here, C depends only on m, n.

The computation takes time $C \log N$, given one-time work $CN \log N$ in space CN.

55. Main Algorithm and Main Lemma

We state here the analogues of our earlier *Main Algorithm* and *Main Lemma* for our present model of computation. Here, C, C', ... denote constants determined by m, n.

<u>*The Main Algorithm*</u> `Procedure` $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$.

/* <u>Inputs are as follows:</u> $\mathcal{A}_0 \subseteq \mathcal{M}$; $Q_0 \in CZ(\mathcal{A}_0)$; $x_0 = (x_1^0, \dots, x_n^0) \in E \cap Q_0^{**}$ (hence, each x_j^0 is a machine number of absolute value $\leq 2^{S_0}$, thanks to our assumptions on E); $P_0 \in \mathcal{P}$, with each $\partial^\alpha P_0(0)$ being a given machine number of absolute value $\leq 2^{A_1(\mathcal{A}_0)S_0}$; and $x = (x_1, \dots, x_n) \in (1 + c_G)Q_0$, with each x_j being a given machine number (hence, $|x_j| \leq C2^{S_0}$, since $x_0 \in E$, and $x_0, x \in Q_0^{**}$ with $Q_0 \in CZ(\mathcal{A}_0)$; recall that Q_0 has sidelength $\leq A_2^{-1} \leq 1$).

Output is as follows: $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ is a polynomial $P^+ \in \mathcal{P}^+$. For each α ($|\alpha| \leq m$), the quantity $\partial^\alpha P^+(0)$ is a machine number of absolute value $\leq 2^{A_3(\mathcal{A}_0)S_0}$. The algorithm specifies P^+ by computing the machine numbers $\partial^\alpha P^+(0)$ for $|\alpha| \leq m$. The polynomial P^+ is to be viewed as an m-jet at x.

*/

```
Line 1   If $\mathcal{A}_0 = \mathcal{M}$, then define $f_x(\mathcal{A}_0, Q_0, x_0, P_0) := P_0$, else
Line 2   { Let $\mathcal{A}'$ be the least $\mathcal{A} \subseteq \mathcal{M}$ such that $Q_0 \in CZ(\mathcal{A})$.
Line 3    If $\mathcal{A}' < \mathcal{A}_0$, then define $f_x(\mathcal{A}_0, Q_0, x_0, P_0) := f_x(\mathcal{A}', Q_0, x_0, P_0)$,
Line 4    else
Line 5    { Produce a list $Q_1, \ldots, Q_{k_{\max}}$ of all the cubes
Line 6     $Q \in CZ(\mathcal{A}_0^-)$ such that $x \in (1 + c_G)Q$ .
Line 7     For each $k = 1, \ldots, k_{\max}$, do the following:
Line 8     { If $E \cap Q_k^{**} = \emptyset$, then set $f_k := P_0$, else
Line 9      { If $x_0 \in Q_k^*$, then set $x_k := x_0$ and $P_k := P_0$, else
Line 10      { Define $x_k :=$ Find-Representative $(Q_k)$.
Line 11         Define $P_k :=$ Find-Neighbor $(P_0, \mathcal{A}_0, x_k)$.
Line 12        } /* Now we have found $x_k$, $P_k$, for all $k$ for which
                 the "else" in Line 8 is executed */
Line 13       Define $f_k := f_x(\mathcal{A}_0^-, Q_k, x_k, P_k)$.
Line 14       } /* Now we have found $f_k$ in all cases */
Line 15     } /* End of the k-loop starting at Line 7 */
Line 16    Define $f_x(\mathcal{A}_0, Q_0, x_0, P_0) =$ (our machine approximation to)
Line 17        $\sum_{k=1}^{k_{\max}} \left\{ J_x^+ \left( \theta_{Q_k}^{\mathcal{A}_0^-} \right) \right\}_{\text{approx}} \odot_x^+ f_k$ .

           /* Here, $\left\{ J_x^+ \left( \theta_{Q_k}^{\mathcal{A}_0^-} \right) \right\}_{\text{approx}}$ is as in Algorithm PU */
Line 18    } /* Balances the curly bracket at Line 5 */
Line 19  } /* Balances the curly bracket at Line 2 */.
```

We check the following elementary result.

Lemma 1. *Assume that the inputs to the* Main Algorithm *are as in the comment "Inputs are as follows". Then the* Main Algorithm *can be executed. It takes time at most* $C'' \log N$*, given one-time work* $C'' N \log N$ *in space* $C''N$*. It produces an output* P^+ *as described in the comment "Output is as follows".*

Proof. We use induction on $\mathcal{A}_0$.

In the base case $\mathcal{A}_0 = \mathcal{M}$, inspection of Line 1 shows that the Main Algorithm executes in time at most C'', and produces as output the polynomial P_0. Since we assume that each $\partial^\alpha P_0(0)$ is a machine number of absolute value $\le 2^{A_1(\mathcal{M})S_0}$ (see the comment "Inputs are as follows"), we have $P^+ = P_0 \in \mathcal{P} \subseteq \mathcal{P}^+$, and $\partial^\alpha P^+(0)$ is a machine number of absolute value $\le 2^{A_1(\mathcal{M})S_0} \le 2^{A_3(\mathcal{M})S_0}$ for $|\alpha| \le m-1$. (See Section 17 for the inequality $A_1(\mathcal{M}) \le A_3(\mathcal{M})$. In fact, all the A_1's are smaller than all the A_3's.) For $|\alpha| = m$, we have $\partial^\alpha P^+(0) = 0$. Thus, the conclusions of Lemma 1 hold in the base case $\mathcal{A}_0 = \mathcal{M}$.

Next, suppose $\mathcal{A}_0 \neq \mathcal{M}$. Recall from Section 52 that we may execute Lines 2–3 in time $C \log N$ given one-time work $CN \log N$ in space CN (see also Section 26). We split the proof into two cases. First, suppose there exists $\mathcal{A} < \mathcal{A}_0$ such that $Q_0 \in CZ(\mathcal{A})$. Let $\mathcal{A}' \subset \mathcal{M}$ be as computed in Line 2. Then $\mathcal{A}' < \mathcal{A}_0$. By inductive hypothesis (Lemma 1 for $\mathcal{A}'$; note that $A_1(\mathcal{A}') \ge A_1(\mathcal{A}_0)$ and $A_3(\mathcal{A}') \le A_3(\mathcal{A}_0)$), we can execute Line 3 in time $C'' \log N$ given one-time work $C'' N \log N$ in space $C''N$; the algorithm terminates, and the output $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ is a polynomial P^+ as in the conclusion of Lemma 1.

Thus, Lemma 1 holds for $\mathcal{A}_0$ in the case $\mathcal{A}' < \mathcal{A}_0$. For the rest of the proof of Lemma 1, we assume that $\mathcal{A}' = \mathcal{A}_0 \neq \mathcal{M}$. Thus, our algorithm passes to the execution of Lines 5-6. Recall from Section 52, that all of our algorithms for CZ cubes work with perfect accuracy in our model of computation. Thus, as is explained in Section 29, we may execute Lines 5–6, and the loop in Lines 8,...,14 is executed at most C times.

Let us examine the k^{th} execution of the loop in Lines 8,...,14. Suppose first that $E \cap Q_k^{**} \neq \emptyset$. Since Lines 8–10 involve only the CZ algorithms, again, the analysis from Section 29 is still valid. In the case where $x_0 \in Q_k^*$, by Line 9 we have that $P_k = P_0 \in \mathcal{P}$, $\partial^\alpha P_k(0)$ is a known machine number of absolute value at most $2^{A_1(\mathcal{A}_0)S_0} \le 2^{A_1(\mathcal{A}_0^-)S_0}$ for each α ($|\alpha| \le m-1$) (recall from Section 17 that $A_1(\mathcal{A}_0) \le A_1(\mathcal{A}_0^-)$); also, the coordinates of $x_k = x_0$ are machine numbers of absolute value at most 2^{S_0}. (See the comment "Inputs are as follows" in the Main Algorithm.)

In the case where $x_0 \notin Q_k^*$, we reach the execution of Line 10. By the defining property of Find-Representative, we know that $x_k \in E \cap Q_k^{**}$. When we start executing Line 11, we have the following: $P_0 \in \mathcal{P}$, $\partial^\alpha P_0(0)$ is a known machine number of absolute value at most $2^{A_1(\mathcal{A}_0)S_0}$ for each α ($|\alpha| \leq m-1$) (see the comment "Inputs are as follows" in the Main Algorithm); also $\mathcal{A}_0 \subset \mathcal{M}$ with $\mathcal{A}_0 \neq \mathcal{M}$; and $x_k \in E$, thanks to Line 10. Thus P_0, $\mathcal{A}_0$, x_k satisfy the requirements for successful execution of Find-Neighbor. (See the comment on "Inputs" in the algorithm "Find-Neighbor" in Section 53.) Consequently, Line 11 executes successfully, in time $\leq C''$, given one-time work $C''N \log N$ in space $C''N$. (See Section 53.) Recall that the output P_k produced by Find-Neighbor in Line 11 satisfies the following: $P_k \in \mathcal{P}$; and for each α, the quantity $\partial^\alpha P_k(0)$ is a machine number of absolute value $\leq 2^{A_1(\mathcal{A}_0^-)S_0}$, which we compute in executing Line 11. (See the comment on "Outputs" in the algorithm Find-Neighbor.)

Having executed Line 11, we pass to Line 13.

Note that, in both of the cases $x_0 \in Q_k^*$ and $x_0 \notin Q_k^*$, we have shown that the following holds, upon reaching Line 13:

(†) $\mathcal{A}_0^- \subseteq \mathcal{M}$; $Q_k \in CZ(\mathcal{A}_0^-)$ (see Lines 5, 6); $x_k \in E \cap Q_k^{**}$; $P_k \in \mathcal{P}$, and for $|\alpha| \leq m-1$, $\partial^\alpha P_k(0)$ is a machine number of absolute value $\leq 2^{A_1(\mathcal{A}_0^-)S_0}$; $x \in (1+c_G)Q_k$ (see Lines 5, 6); and the coordinates of x are machine numbers (see the comment "Inputs are as follows" in the Main Algorithm).

The above remarks tell us that the inputs $\mathcal{A}_0^-$, Q_k, x_k, P_k and x are as in the comment "Inputs are as follows" in the Main Algorithm. Since also $\mathcal{A}_0^- < \mathcal{A}_0$, we may apply Lemma 1 to $f_x(\mathcal{A}_0^-, Q_k, x_k, P_k)$. Consequently, Line 13 executes in time $\leq C'' \log N$, given one-time work $C''N \log N$ in space $C''N$. Moreover, the output f_k produced by Line 13 satisfies:

(††) $f_k \in \mathcal{P}^+$; for each α ($|\alpha| \leq m$), we have computed $\partial^\alpha f_k(0)$, which is a machine number of absolute value at most $2^{A_3(\mathcal{A}_0^-)S_0}$.

We have just proven (††) for the case $E \cap Q_k^{**} \neq \emptyset$. Inspection of Line 8 shows that (††) holds also for the case $E \cap Q^{**} = \emptyset$; in this case $f_k = P_0$,

$$|\partial^\alpha f_k(0)| \leq 2^{A_1(\mathcal{A}_0)S_0} \leq 2^{A_3(\mathcal{A}_0^-)S_0}$$

and (††) follows; see the comment "Inputs are as follows" in the Main Algorithm. (We use the fact that $A_1(\mathcal{A}_0) \leq A_3(\mathcal{A}_0^-)$; see Section 17.) Hence, (††) holds for any k ($1 \leq k \leq k_{\max}$).

The above discussion shows that we execute the k-loop (Lines 7,...,15) in time $C'' \log N$, given one-time work $C''N \log N$ in space $C''N$. Thereafter, we pass to Line 16.

Next, we check that Lines 16, 17 can be executed in time $C'' \log N$, given one-time work $C'' N \log N$ in space $C'' N$; and we discuss the output $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ produced by Lines 16, 17. Each f_k satisfies (††).

To prepare for execution of the product $\odot_x^+$ in Line 17, we must compute (a machine approximation to) $(\partial^\alpha f_k(x))_{|\alpha| \le m}$, given the $(\partial^\alpha f_k(0))_{|\alpha| \le m}$ from (††). We obtain values $\{\partial^\alpha f_k(x)\}_{\mathsf{approx}}$ for $|\alpha| \le m$; each of these is a machine number of absolute value

$$\le 2^{C''' A_3(\mathcal{A}_0^-) S_0}.$$

(This follows easily from (††), together with our earlier observation that the coordinates of x are machine numbers of absolute value $\le 2^{CS_0}$.) Given the $\{\partial^\alpha f_k(x)\}_{\mathsf{approx}}$, and the output of Algorithm PU, we can compute a machine approximation to

$$\partial^\alpha \left[\left\{ J_x^+(\theta_{Q_k}^{\mathcal{A}_0^-}) \right\}_{\text{approx}} \odot_x^+ f_k \right](x) \quad \text{for } |\alpha| \le m;$$

and these machine approximations are all less than $2^{\hat{C} A_3(\mathcal{A}_0^-) S_0}$ in absolute value. Summing over $k = 1, \dots, k_{\max}$, and recalling that $k_{\max} \le C$, we obtain (for $|\alpha| \le m$) a machine number θ_α of absolute value $\le C \cdot 2^{\hat{C} A_3(\mathcal{A}_0^-) S_0}$, which serves as our approximation to

$$\partial^\alpha \left[\sum_{k=1}^{k_{\max}} \{J_x^+(\theta_{Q_k}^{\mathcal{A}_0^-})\}_{\mathsf{approx}} \odot_x^+ f_k \right](x).$$

Expanding

$$y \mapsto \sum_{|\alpha| \le m} \frac{1}{\alpha!} \theta_\alpha (y - x)^\alpha$$

in powers of y, we obtain coefficients ψ_α (with round-off errors), such that $(\psi_\alpha)_{|\alpha| \le m}$ is an approximation to

$$\left(\partial^\alpha \left[\sum_{k=1}^{k_{\max}} \{J_x^+(\theta_{Q_k}^{\mathcal{A}_0^-})\}_{\mathsf{approx}} \odot_x^+ f_k \right](0)\right)_{|\alpha| \le m}.$$

The ψ_α are machine numbers of absolute value $\le 2^{C^\# A_3(\mathcal{A}_0^-) S_0}$, thanks to our estimates for the θ_α and for the coordinates of x. Thus, we can execute Lines 16, 17. Their output is the polynomial

$$f_x(\mathcal{A}_0, Q_0, x_0, P_0) = \sum_{|\alpha| \le m} \frac{1}{\alpha!} \psi_\alpha y^\alpha,$$

specified by exhibiting the ψ_α. Since Algorithm PU takes time $\le C'' \log N$, given one-time work $C'' N \log N$ in space $C'' N$, the same is true of Lines 16, 17. (Here again, we use the fact that $k_{\max} \le C$.)

Moreover, $f_x(\mathcal{A}_0, Q_0, x_0, P_0) \in \mathcal{P}^+$, and for $|\alpha| \leq m$,

$$\partial^\alpha[f_x(\mathcal{A}_0, Q_0, x_0, P_0)](0)$$

is a machine number of absolute value

$$\leq 2^{C^\# A_3(\mathcal{A}_0^-)S_0} \leq 2^{A_0 A_3(\mathcal{A}_0^-)S_0} \leq 2^{A_3(\mathcal{A}_0)S_0},$$

which we have computed. Here, we assume that

$$A_0 \geq C^\#,$$

which is a legitimate assumption, as $C^\#$ is a constant depending only on m and n. The execution of the Main Algorithm for $(x, \mathcal{A}_0, Q_0, x_0, P_0)$ terminates after execution of Lines 16,17. We have shown that the Main Algorithm for $(x, \mathcal{A}_0, Q_0, x_0, P_0)$ runs in time $C'' \log N$, given one-time work $C'' N \log N$ in space $C'' N$.

We have established the conclusions of Lemma 1 for the inputs $(x, \mathcal{A}_0, Q_0, x_0, P_0)$. The proof of Lemma 1 is complete. ■

Note that, in the proof of Lemma 1, we needed some control on the size, e.g., of the f_k, in order to avoid overflow errors in executing Lines 16,17.

Now we are ready to state the Main Lemma.

Main Lemma for $\mathcal{A}_0$:

Let $Q_0 \in CZ(\mathcal{A}_0)$, *let* $x_0 \in E \cap Q_0^{**}$ *with* $x_0 \in E \cap Q_0^*$ *in case* $E \cap Q_0^* \neq \emptyset$, *and let* $P_0 \in \mathcal{P}$.

Suppose that, for each α $(|\alpha| \leq m-1)$, *the quantity* $\partial^\alpha P_0(0)$ *is a machine number of absolute value at most* $2^{A_1(\mathcal{A}_0)S_0}$.

Let M_0 *be a machine number, with* $2^{-S_0} \leq M_0 \leq A_1(\mathcal{A}_0)2^{+S_0}$. *Assume that* $P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0)$.

Then there exists $F \in C^m((1+c_G)Q_0)$, *with the following properties:*

(I) $J_x^+(F) - P_0 \in A_3(\mathcal{A}_0) \cdot M_0 \cdot B^+(x, \delta_{Q_0})$ *for all* $x \in (1+c_G)Q_0$.

(II) $J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A}_0) \cdot M_0)$ *for all* $x \in E \cap (1+c_G)Q_0$.

(III) *If* $x_0 \in (1+c_G)Q_0$, *then* $J_{x_0}(F) = P_0$.

(IV) *Let* $x = (x_1, \ldots, x_n) \in (1+c_G)Q_0$, *and assume each* x_j *is a machine number. Then*

$$\left|\partial^\alpha[J_x^+(F) - f_x(\mathcal{A}_0, Q_0, x_0, P_0)](0)\right| \leq 2^{A_3(\mathcal{A}_0)S_0 - \bar{S}/2} \text{ for } |\alpha| \leq m.$$

Here, $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ is the polynomial produced by the Main Algorithm for inputs $(\mathcal{A}_0, Q_0, x_0, P_0)$. Note that $\mathcal{A}_0, Q_0, x_0, P_0$ and x in (IV) are as in the comment “Inputs are as follows” in the Main Algorithm. Hence, by Lemma 1, $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ is a well-defined polynomial belonging to $\mathcal{P}^+$.

56. Proof of the Main Lemma

We write C, c, C', etc. to denote constants determined by m and n. As for the Main Lemma for "perfect arithmetic" (Section 29), we proceed by induction on $\mathcal{A}_0$, with respect to our order relation $<$ on subsets of $\mathcal{M}$.

Base Case: Suppose $\mathcal{A}_0 = \mathcal{M}$. Then we have $Q_0 \in CZ(\mathcal{M})$ and $P_0 \in \Gamma(x_0, \ell(\mathcal{M}), M_0)$, with $x_0 \in E \cap Q_0^{**}$ such that $x_0 \in E \cap Q^*$ in case $E \cap Q_0^* \neq \emptyset$. Thus, we are in the base case of the induction on $\mathcal{A}_0$ that proves the Main Lemma for perfect arithmetic. In that case, we showed that $F = P_0$ on $(1+c_G)Q_0$ satisfies (I) and (II) in the statement of our present Main Lemma. Also, (III) is obvious. Moreover, Line 1 of the Main Algorithm gives here $f_x(\mathcal{M}, Q_0, x_0, P_0) = P_0$, i.e., $\partial^\alpha[f_x(\mathcal{M}, Q_0, x_0, P_0)](0) = \partial^\alpha P_0(0)$ exactly, for $|\alpha| \leq m$. Since also $\partial^\alpha[J_x^+(F)](0) = \partial^\alpha P_0(0)$ exactly ($|\alpha| \leq m$), it follows that $\partial^\alpha[J_x^+(F) - f_x(\mathcal{M}, Q_0, x_0, P_0)](0) = 0$ exactly, for $|\alpha| \leq m$. Hence, conclusion (IV) holds also, completing the proof of the Main Lemma in the base case.

Induction Step: Fix $\mathcal{A}_0 \subset \mathcal{M}$ with $\mathcal{A}_0 \neq \mathcal{M}$, and assume that the Main Lemma for $\mathcal{A}$ holds for all $\mathcal{A} < \mathcal{A}_0$. We will prove the Main Lemma for $\mathcal{A}_0$. Let M_0, Q_0, x_0, P_0 be as in the hypotheses of the Main Lemma for $\mathcal{A}_0$. We distinguish two cases.

Trivial Case: $Q_0 \in CZ(\mathcal{A})$ for some $\mathcal{A} < \mathcal{A}_0$.

Non-Trivial Case: $Q_0 \notin CZ(\mathcal{A})$ for all $\mathcal{A} < \mathcal{A}_0$.

In the trivial case, let $\mathcal{A}'$ be minimal, subject to $Q_0 \in CZ(\mathcal{A}')$. Then $\mathcal{A}' < \mathcal{A}_0$. By inspection of Lines 1,2,3 of the Main Algorithm, we see that

(1) $f_x(\mathcal{A}_0, Q_0, x_0, P_0) = f_x(\mathcal{A}', Q_0, x_0, P_0)$ whenever $x = (x_1, \dots, x_n) \in (1+c_G)Q_0$ with each x_j being a machine number.

As in the trivial case of the Main Lemma for perfect arithmetic, we find that CM_0, Q_0, x_0, P_0 satisfy the hypotheses of the Main Lemma for $\mathcal{A}'$. Applying the Main Lemma for $\mathcal{A}'$ (which we may do, since $\mathcal{A}' < \mathcal{A}_0$ and since $A_1(\mathcal{A}') \geq A_1(\mathcal{A}_0)$), we obtain $F \in C^m((1+c_G)Q_0)$, with the following properties:

(I)′ $J_x(F) - P_0 \in A_3(\mathcal{A}') \cdot CM_0 \cdot B^+(x, \delta_{Q_0})$ for all $x \in (1+c_G)Q_0$;

(II)′ $J_x(F) \in \Gamma(x, 0, A_3(\mathcal{A}') \cdot CM_0)$ for all $x \in E \cap (1+c_G)Q_0$;

(III)′ If $x_0 \in (1+c_G)Q_0$, then $J_{x_0}(F) = P_0$; and

(IV)′ Let $x = (x_1, \dots, x_n) \in (1 + c_G)Q_0$, and assume each x_j is a machine number. Then

$$|\partial^\alpha[J_x^+(F) - f_x(\mathcal{A}', Q_0, x_0, P_0)](0)| \le 2^{A_3(\mathcal{A}')S_0 - \bar{S}/2} \text{ for } |\alpha| \le m\,.$$

Applying (1), and recalling that $CA_3(\mathcal{A}') \le A_3(\mathcal{A}_0)$ for $\mathcal{A}' < \mathcal{A}_0$, we see that (I)′,...,(IV)′ imply the desired properties (I),...,(IV) for F. This completes the inductive step in the proof of the Main Lemma, in the trivial case.

For the rest of this section, we assume we are in the non-trivial case. Let

$$\mathcal{Q} = \{Q \in CZ(\mathcal{A}_0^-) : (1 + c_G)Q_0 \cap (1 + c_G)Q \ne \emptyset\}.$$

For each $Q \in \mathcal{Q}$, we define a point $x_Q \in \mathbb{R}^n$ and a polynomial $P_Q \in \mathcal{P}$, as follows.

(2) If $E \cap Q^{**} = \emptyset$, then we set $x_Q =$ center of $Q, P_Q = P_0$.

(3) If $x_0 \in E \cap Q^*$, then we set $x_Q = x_0$, $P_Q = P_0$.

(4) If $E \cap Q^{**} \ne \emptyset$ and $x_0 \notin E \cap Q^*$, then we set

$$x_Q = \text{Find-Representative}(Q).$$

(Note that $x_Q \in E \cap Q^{**}$ with $x_Q \in E \cap Q^*$ in case $E \cap Q^* \ne \emptyset$, by the defining property of Find-Representative.)

We then let $P_Q =$ Find-Neighbor $(P_0, \mathcal{A}_0, x_Q)$. Note that $P_0, \mathcal{A}_0, x_Q$ are as required for inputs of the procedure Find-Neighbor.

Thus, in all cases, $P_Q \in \mathcal{P}$, and $\partial^\alpha P_Q(0)$ is a machine number of absolute value at most $2^{A_1(\mathcal{A}_0^-)S_0}$ for $|\alpha| \le m - 1$. (In Case (2) and Case (3), this follows from our assumptions on P_0 in the hypotheses of the Main Lemma, and the inequality $A_1(\mathcal{A}_0) \le A_1(\mathcal{A}_0^-)$. In Case (4), it follows from the defining properties of Find-Neighbor in Section 53.) Note that in Case (2), we necessarily have $\delta_Q = A_2^{-1}$ (see Lemma 4 of Section 21).

In Case (4), the defining property of Find-Neighbor also tells us the following.

(5) Suppose $P \in \mathcal{P}$ and $M > 2^{-S_0}$ satisfy

$\partial^\alpha(P - P_0)(x_Q) = 0$ for all $\alpha \in \mathcal{A}_0$, and $P \in \Gamma(x_Q, \ell(\mathcal{A}_0) - 1, M)$.

Then we can express

$P_Q = P_Q^{\text{main}} + P_Q^{\text{err}}$, with P_Q^{main}, $P_Q^{\text{err}} \in \mathcal{P}$,

$\partial^\alpha(P_Q^{\text{main}} - P_0)(x_Q) = 0$ for all $\alpha \in \mathcal{A}_0$,

and $P_Q^{\text{main}} \in \Gamma(x_Q, \ell(\mathcal{A}_0) - 1, CM)$;

and $|\partial^\alpha P_Q^{\text{err}}(0)| \le 2^{A_1(\mathcal{A}_0^-)S_0 - \bar{S}/2}$ for $|\alpha| \le m - 1$.

Since $P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0)$, and since we are in the non-trivial case, we may apply here the analysis of Section 32. By Lemma 7 from Section 32, we learn the following in Case (4).

(6) There exists $P \in \mathcal{P}$, satisfying
$\partial^\alpha(P - P_0)(x_Q) = 0$ for all $\alpha \in \mathcal{A}_0$; and $P \in \Gamma(x_Q, \ell(\mathcal{A}_0) - 1, C'M_0)$.
Moreover, any such P satisfies that $P - P_0 \in CM_0 \mathcal{B}(x_Q, A_2 \delta_{Q_0})$.

Since $M_0 \geq 2^{-S_0}$, then by (5) and (6), we conclude that in Case (4), we may write $P_Q = P_Q^{\mathrm{main}} + P_Q^{\mathrm{err}}$, with P_Q^{main}, $P_Q^{\mathrm{err}} \in \mathcal{P}$ such that

(7) $\partial^\alpha(P_Q^{\mathrm{main}} - P_0)(x_Q) = 0$ for all $\alpha \in \mathcal{A}_0$,
$P_Q^{\mathrm{main}} \in \Gamma(x_Q, \ell(\mathcal{A}_0) - 1, CM_0)$,
$P_Q^{\mathrm{main}} - P_0 \in CM_0 \mathcal{B}(x_Q, A_2 \delta_{Q_0})$; and
$|\partial^\alpha P_Q^{\mathrm{err}}(0)| \leq 2^{A_1(\mathcal{A}_0^-) S_0 - \bar{S}/2}$ for $|\alpha| \leq m - 1$.

In Case (2) and in Case (3), we simply set $P_Q^{\mathrm{main}} = P_Q$ and $P_Q^{\mathrm{err}} = 0$. Note that (7) holds also in Case (3), since in this case, $x_Q = x_0$ and by our assumptions

$$P_Q^{\mathrm{main}} = P_Q = P_0 \in \Gamma(x_Q, \ell(\mathcal{A}_0), CM_0) \subset \Gamma(x_Q, \ell(\mathcal{A}_0) - 1, C'M_0).$$

Thus (7) holds in Case (3) and Case (4). Note that (7) shows that $P_Q^{\mathrm{main}} \in \Gamma^{\#}_{\mathcal{A}_0}(x_Q, P_0, CM_0)$, in Case (3) and Case (4). (See Section 32 for the definition of $\Gamma^{\#}$.)

Next, we claim that,

(8) $P_Q^{\mathrm{main}} - P_{Q'}^{\mathrm{main}} \in CM_0 \mathcal{B}(x_Q, A_2 \delta_Q)$ whenever $(1 + c_G)Q \cap (1 + c_G)Q' \neq \emptyset$; $Q, Q' \in \mathcal{Q}$.

Indeed, (8) follows from (7) and Lemma 8 in Section 32, when both Q and Q' satisfy either (3) or (4). In case where at least one of Q, Q' satisfies (2), then (8) follows from (7). For instance, if Q satisfies (2), then $P_Q = P_0$ and $\delta_Q = A_2^{-1} \geq \delta_{Q_0}$; hence (8) either follows from (7), or else holds trivially since $P_{Q'}^{main} = P_{Q'} = P_0$.

We recall that $\delta_Q \geq c \cdot 2^{-S_0}$ for all $Q \in CZ(\mathcal{A}_0^-)$, hence for all $Q \in \mathcal{Q}$. Recall also that $|x_Q| \leq 2^{CS_0}$ for $Q \in \mathcal{Q}$ (because $x_Q \in Q^{**}$, $Q \in CZ(\mathcal{A}_0^-)$, $Q_0 \in CZ(\mathcal{A}_0)$, $(1 + c_G)Q \cap (1 + c_G)Q_0 \neq \emptyset$, $x_0 \in E \cap Q_0^{**}$, and $|x| \leq C \cdot 2^{S_0}$ for $x \in E$). Therefore from (7) we obtain that in Case (4),

(9) $|\partial^\beta P_Q^{\mathrm{err}}(x_Q)| \leq 2^{A_1(\mathcal{A}_0^-) S_0 - \bar{S}/2} \leq CM_0 \cdot (A_2 \delta_Q)^{m - |\beta|}$ for $|\beta| \leq m - 1$.

(Here, we use our hypothesis $M_0 \geq 2^{-S_0}$ from the Main Lemma, as well as (SU2) from Section 49.) Note that (9) trivially holds in Cases (2) and (3), as $P_Q^{\mathrm{err}} = 0$ in those cases.

Next, suppose $Q, Q' \in \mathcal{Q}$, with

$$(1+c_G)Q \cap (1+c_G)Q' \neq \emptyset.$$

Then $\frac{1}{2}\delta_Q \leq \delta_{Q'} \leq 2\delta_Q$ and $|x_Q - x_{Q'}| \leq C\delta_Q$. Hence, applying (8) to Q and Q', and then applying (9), we learn that

(10) $|\partial^\beta(P_Q - P_{Q'})(x_Q)| \leq C'''M_0 \cdot (A_2\delta_Q)^{m-|\beta|}$ for $|\beta| \leq m-1$.

Next, recall that $(\Gamma(x_Q, \ell(\mathcal{A}_0)-1, M))_{M>0}$ is C-equivalent above 2^{-S_0} to the blob arising from a CS_0-bit MALP with constant C'. Therefore, from Lemma 4 in Section 40, we have the following elementary result.

(11) Suppose $P \in \Gamma(x_Q, \ell(\mathcal{A}_0)-1, M), M \geq 2^{-S_0}$, and suppose that $|\partial^\alpha P'(0)| \leq 2^{-CS_0}$ (for a large enough constant C determined by m and n). Then $P+P' \in \Gamma(x_Q, \ell(\mathcal{A}_0)-1, C'M)$.

Taking $P = P_Q^{\text{main}}$ and $P' = P_Q^{\text{err}}$ in (11), recalling that $M_0 \geq 2^{-S_0}$ by our assumptions, and applying (7) and (SU2), we learn that

(12) $P_Q \in \Gamma(x_Q, \ell(\mathcal{A}_0)-1, C'M_0)$ for $Q \in \mathcal{Q}$ in Case (4).

We have (12) also for $Q \in \mathcal{Q}$ in Case (3), since then

$$P_Q = P_0 \in \Gamma(x_0, \ell(\mathcal{A}_0), M_0) = \Gamma(x_Q, \ell(\mathcal{A}_0), M_0) \subseteq \Gamma(x_Q, \ell(\mathcal{A}_0)-1, C'M_0).$$

In Case (2), there is no analogue of (12), since $x_Q \notin E$, and the $\Gamma(x_Q, \ell, M)$ are undefined.

Our basic results on the polynomials P_Q are (10) and (12).

We prepare to apply our induction hypothesis, namely the Main Lemma for $\mathcal{A}_0^-$.

Let $Q \in \mathcal{Q}$ be in Case (3) or (4). We will check that the cube Q, the point x_Q, the polynomial P_Q, and the constant $C''M_0$ (for a suitable integer constant $C'' \geq 1$ depending only on m, n), satisfy the hypotheses of the Main Lemma for $\mathcal{A}_0^-$. In fact:

- $Q \in CZ(\mathcal{A}_0^-)$, since $Q \in \mathcal{Q}$.
- $x_Q \in E \cap Q^{**}$, with $x_Q \in E \cap Q^*$ if $E \cap Q^* \neq \emptyset$. (In Case (3), this holds, since $x_Q = x_0 \in E \cap Q^*$; in Case (4), it follows from the defining property of the procedure Find-Representative.)
- $P_Q \in \mathcal{P}$, and, for $|\alpha| \leq m-1$, $\partial^\alpha P_Q(0)$ is a machine number of absolute value at most $2^{A_1(\mathcal{A}_0^-)S_0}$ (as we noted immediately after (2), (3), (4)).

- $C''M_0$ is a machine number satisfying
$$2^{-S_0} \le C''M_0 \le C'''A_1(\mathcal{A}_0)2^{S_0} \le A_1(\mathcal{A}_0^-)2^{S_0},$$
provided that $A_0 \ge C'''$. (Recall from Section 17 that $A_1(\mathcal{A}_0^-) \ge A_0A_1(\mathcal{A}_0)$.)
- $P_Q \in \Gamma(x_Q, \ell(\mathcal{A}_0^-), C''M_0)$, as we see from (12).

This completes our verification of the hypotheses of the Main Lemma for $\mathcal{A}_0^-$. Applying that lemma, we obtain a function

(13) $F_Q \in C^m((1+c_G)Q)$, for each $Q \in \mathcal{Q}$ in Case (3) or (4), satisfying the following:

(14) $J_x^+(F_Q) - P_Q \in A_3(\mathcal{A}_0^-) \cdot C''M_0 \cdot B^+(x, \delta_Q)$ for $x \in (1+c_G)Q$.

(15) $J_x(F_Q) \in \Gamma(x, 0, A_3(\mathcal{A}_0^-) \cdot C''M_0)$ for all $x \in E \cap (1+c_G)Q$.

(16) If $x_Q \in (1+c_G)Q$, then $J_{x_Q}(F_Q) = P_Q$.

(17) Let $x = (x_1, \dots, x_n) \in (1+c_G)Q$, and assume each x_j is a machine number. Then
$$|\partial^\alpha[J_x^+(F_Q) - f_x(\mathcal{A}_0^-, Q, x_Q, P_Q)](0)| \le 2^{A_3(\mathcal{A}_0^-)S_0 - \bar{S}/2} \quad \text{for } |\alpha| \le m.$$

Properties (14),..., (17) hold when $Q \in \mathcal{Q}$ falls into Case (3) or (4). In Case (2), we simply define

(18) $F_Q = P_0$.

Properties (13), (14), (16) hold also for $Q \in \mathcal{Q}$ in Case (2), since we then have $J_x^+(F_Q) = J_{x_Q}(F_Q) = P_Q = P_0$. Property (15) holds vacuously in Case (2), while property (17) makes no sense in Case (2), since $f_x(\mathcal{A}_0^-, Q, x_Q, P_Q)$ is defined only when $x_Q \in E$. (See the comment "Inputs are as follows" in the Main Algorithm.)

With $\theta^Q_{\mathcal{A}_0^-}$ as in Section 54, we define

(19) $F = \sum_{Q \in \mathcal{Q}} \theta^Q_{\mathcal{A}_0^-} F_Q$ on $(1+c_G)Q_0$.

Note that (19) makes sense, thanks to (13), since $\theta^Q_{\mathcal{A}_0^-}$ is supported in $(1+c_G)Q$. Recall that any point in $\mathbb{R}^n$ has a small neighborhood that meets at most C of the supports of the $\theta^Q_{\mathcal{A}_0^-}$ $(Q \in CZ(\mathcal{A}_0^-))$. Consequently,

(20) $F \in C^m((1+c_G)Q_0)$.

We check that the above function F satisfies properties (I),...,(IV) in the conclusion of the Main Lemma for $\mathcal{A}_0$.

Regarding (I), let $x \in (1+c_G)Q_0$, and let $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ contain x. Then

$$(21)\quad J_x^+(F-P_0) \;=\; J_x^+(F_{\widehat{Q}}-P_0) + \sum_{Q\in\mathcal{Q}} J_x^+(\theta^Q_{\mathcal{A}_0^-}) \odot_x^+ J_x^+(F_Q - F_{\widehat{Q}}),$$

since

$$\sum_{Q\in\mathcal{Q}} J_x^+(\theta^Q_{\mathcal{A}_0^-}) = 1.$$

A given $Q \in \mathcal{Q}$ makes a non-zero contribution to (21) only if $x \in (1+c_G)Q$. There are at most C such $Q \in \mathcal{Q}$, and they all satisfy $\frac{1}{2}\delta_{\widehat{Q}} \le \delta_Q \le 2\delta_{\widehat{Q}}$, and

$$(22)\quad \begin{aligned} J_x^+(F_Q - F_{\widehat{Q}}) &= P_Q - P_{\widehat{Q}} + J_x^+(F_Q - P_Q) - J_x^+(F_{\widehat{Q}} - P_{\widehat{Q}}) \\ &\in C'''A_3(\mathcal{A}_0^-)M_0 B^+(x,\delta_{\widehat{Q}}) + CM_0B^+(x, A_2\delta_{\widehat{Q}}), \end{aligned}$$

thanks to (14) and (10). It therefore follows from (21) and (14), along with our estimates for the derivatives of the $\theta^Q_{\mathcal{A}_0^-}$, that

$$(23)\quad J_x^+(F-P_0) \in [P_{\widehat{Q}} - P_0] + C^{iv}A_2^m A_3(\mathcal{A}_0^-)M_0B^+(x,\delta_{\widehat{Q}}).$$

Next, note that

$$(24)\quad P_{\widehat{Q}} - P_0 \in CM_0B^+(x_{\widehat{Q}}, A_2\delta_{Q_0}) \subseteq C'M_0B^+(x, A_2\delta_{Q_0}).$$

If $\widehat{Q}$ falls into Case (4), this follows from (7); note that $|x - x_{\widehat{Q}}| \le C\delta_{\widehat{Q}} < C'\delta_{Q_0}$ since $x \in \widehat{Q} \cap (1+c_G)Q_0$, with $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ and $Q_0 \in CZ(\mathcal{A}_0)$. (See Lemma 6 from Section 21.) If instead $\widehat{Q}$ falls into Case (2) or (3), then (24) holds trivially, since then $P_{\widehat{Q}} = P_0$.

We have $\delta_{\widehat{Q}} \le C\delta_{Q_0}$, by Lemma 6 from Section 21. Consequently, (23) and (24) together yield

$$J_x^+(F) - P_0 \in C^v A_2^m A_3(\mathcal{A}_0^-)M_0B^+(x,\delta_{Q_0}).$$

This implies conclusion (I) of the Main Lemma for $\mathcal{A}_0$, since $A_3(\mathcal{A}_0) \ge C^vA_2^mA_3(\mathcal{A}_0^-)$.

Next, we check conclusion (II) for our function F. Let $x \in E \cap (1+c_G)Q_0$, and let $\widehat{Q} \in CZ(\mathcal{A}_0^-)$ with $\widehat{Q} \ni x$. Then (19) gives

$$(25)\quad J_x(F) \;=\; J_x(F_{\widehat{Q}}) \;+\; \sum_{Q\in\mathcal{Q}} J_x(\theta^Q_{\mathcal{A}_0^-}) \odot_x J_x(F_Q - F_{\widehat{Q}}).$$

The only $Q \in \mathcal{Q}$ that contribute to (25) are those satisfying $x \in (1+c_G)Q$. There are at most C such $Q \in \mathcal{Q}$, and they satisfy $\frac{1}{2}\delta_{\widehat{Q}} \le \delta_Q \le 2\delta_{\widehat{Q}}$. Moreover, given such a Q, we can argue as follows:

Applying (15) to Q and to $\widehat{Q}$, we find that

$$(26)\quad \begin{aligned} J_x(F_Q - F_{\widehat{Q}}) &\in \Gamma(x,0,C''A_3(\mathcal{A}_0^-)M_0) - \Gamma(x,0,C''A_3(\mathcal{A}_0^-)M_0) \\ &\subseteq C'''A_3(\mathcal{A}_0^-)M_0\sigma(x,0). \end{aligned}$$

Also, (22) applies, and it gives

(27) $J_x(F_Q - F_{\widehat{Q}}) \in C^{iv} A_2^m A_3(\mathcal{A}_0^-) M_0 B(x, \delta_{\widehat{Q}})$.

From (26), (27), our estimates for the derivatives of $\theta^Q_{\mathcal{A}_0^-}$ and the Whitney t-convexity of $\sigma(x,0)$, we conclude that

(28) $J_x(\theta^Q_{\mathcal{A}_0^-}) \odot_x J_x(F_Q - F_{\widehat{Q}}) \in C^v A_2^m \cdot A_3(\mathcal{A}_0^-) M_0 \sigma(x,0)$.

Putting (15) and (28) into (25), we conclude that

(29) $J_x(F) \in \Gamma(x, 0, C^{vi} A_2^m \cdot A_3(\mathcal{A}_0^-) M_0) \subseteq \Gamma(x, 0, A_3(\mathcal{A}_0) M_0)$,

since $A_3(\mathcal{A}_0) \geq C^{vi} A_2^m \cdot A_3(\mathcal{A}_0^-)$. Conclusion (II) for our function F is immediate from (29).

Next, we check conclusion (III). Suppose $x_0 \in (1 + c_G) Q_0$. From (19),

(30) $J_{x_0}(F) = \sum_{Q \in \mathcal{Q}} J_{x_0}(\theta^Q_{\mathcal{A}_0^-}) \odot_{x_0} J_{x_0}(F_Q)$.

The only Q that contribute to (30) are those satisfying $x_0 \in (1 + c_G) Q$. Such Q fall into Case (3), and therefore satisfy $P_Q = P_0, x_Q = x_0$. In particular, such Q satisfy $x_Q = x_0 \in (1 + c_G) Q$, and therefore also $J_{x_0}(F_Q) = J_{x_Q}(F_Q) = P_Q = P_0$, by (16). Consequently, (30) implies

$$J_{x_0}(F) = \sum_{Q \in \mathcal{Q}} J_{x_0}(\theta^Q_{\mathcal{A}_0^-}) \odot_{x_0} P_0 = P_0,$$

proving conclusion (III) of the Main Lemma for $\mathcal{A}_0$.

It remains to check conclusion (IV).
Let $x = (x_1, \ldots, x_n) \in (1 + c_G) Q_0$, and assume each x_j is a machine number. Let $Q_1, \ldots, Q_{k_{\max}}$ be a list of all the $Q \in CZ(\mathcal{A}_0^-)$ for which we have $x \in (1 + c_G) Q$. This precise list of cubes is being computed in Lines 5, 6 of the Main Algorithm. We assume that our enumeration here corresponds to that in Lines 5, 6.

Then from (19) we obtain

(31) $J_x^+(F) = \sum_{k=1}^{k_{\max}} J_x^+(\theta^{Q_k}_{\mathcal{A}_0^-}) \odot_x^+ J_x^+(F_{Q_k})$.

Note that

$$J_x^+(F_{Q_k}) = P_0 \text{ if } E \cap Q_k^{**} = \emptyset \quad \text{(see (18))};$$

and if $E \cap Q_k^{**} \neq \emptyset$, then the following holds:

> If $x_0 \in Q_k^*$, then $x_{Q_k} = x_0$ and $P_{Q_k} = P_0$, else: we use Find-Representative to find $x_{Q_k} \in E \cap Q_k^{**}$, with $x_{Q_k} \in E \cap Q_k^*$ if $E \cap Q_k^* \neq \emptyset$; then we use Find-Neighbor to compute a polynomial $P_{Q_k} \in \mathcal{P}$.

Comparing the above discussion with Lines 5,..., 11 of the Main Algorithm, we find that:

(32) f_k from Line 8 agrees perfectly with $J^+_x(F_{Q_k})$ if $E \cap Q^{**}_k = \emptyset$.

(33) x_k, P_k from Lines 9,..., 12 agree perfectly with our present x_{Q_k}, P_{Q_k}, in case $E \cap Q^{**}_k \neq \emptyset$.

From (33) and (17), we see that

(34) $|\partial^\alpha[J^+_x(F_{Q_k}) - f_x(\mathcal{A}^-_0, Q_k, x_k, P_k)](0)| \leq 2^{A_3(\mathcal{A}^-_0)S_0 - \bar{S}/2}$ for $|\alpha| \leq m$, whenever $E \cap Q^{**}_k \neq \emptyset$.

Here, Q_k, x_k, P_k are as in the Main Algorithm. Comparing (34) with Line 13 in the Main Algorithm, we conclude that $|\partial^\alpha[J^+_x(F_{Q_k}) - f_k](0)| \leq 2^{A_3(\mathcal{A}^-_0)S_0 - \bar{S}/2}$ for $|\alpha| \leq m$, whenever $E \cap Q^{**}_k \neq \emptyset$.

Together with (32), this shows that, when we execute Lines 16, 17 of the Main Algorithm, we have

(35) $|\partial^\alpha[J^+_x(F_{Q_k}) - f_k](0)| \leq 2^{A_3(\mathcal{A}^-_0)S_0 - \bar{S}/2}$ for all $|\alpha| \leq m$, $1 \leq k \leq k_{\max}$.

Note also that

(36) $|\partial^\alpha[J^+_x(F_{Q_k})](0)| \leq 2^{CA_3(\mathcal{A}^-_0)S_0}$ for $|\alpha| \leq m$.

In fact, in the case $E \cap Q^{**}_k = \emptyset$, we have $J^+_x(F_{Q_k}) = P_0$ by (18), and (36) follows from our hypotheses on P_0 in the formulation of the Main Lemma (recall that $A_1(\mathcal{A}_0) \leq A_3(\mathcal{A}^-_0)$). In the case where $E \cap Q^{**}_k \neq \emptyset$, since $x \in (1 + c_G)Q_k$, (14) applies, and we have

(37) $|\partial^\alpha[F_{Q_k} - P_k](x)| \leq A_3(\mathcal{A}^-_0) \cdot C'' M_0 \cdot \delta^{m-|\alpha|}_{Q_k}$ for $|\alpha| \leq m$, $1 \leq k \leq k_{\max}$.

Since also $|x| \leq 2^{CS_0}$ (since $x \in (1 + c_G)Q_0$, with $x_0 \in Q^{**}_0 \cap E$, $\delta_{Q_0} \leq 1$), it follows from (37) that

$$|\partial^\alpha[J^+_x(F_{Q_k} - P_k)](0)| \leq A_3(\mathcal{A}^-_0)M_0 \cdot 2^{CS_0}$$

for $|\alpha| \leq m$, $1 \leq k \leq k_{\max}$. Since $M_0 \leq A_1(\mathcal{A}_0)2^{S_0} \leq 2^{A_1(\mathcal{A}^-_0)S_0}$, and since $|\partial^\alpha P_k(0)| \leq 2^{A_1(\mathcal{A}^-_0)S_0} \leq 2^{A_3(\mathcal{A}^-_0)S_0}$ (see the text right after (4)), then (36) follows. This completes our verification of (36).

Regarding the m-jets of our $\theta^{Q_k}_{\mathcal{A}^-_0}$, we recall that Algorithm PU produces $\{J^+_x(\theta^{Q_k}_{\mathcal{A}^-_0})\}_{\mathsf{approx}} \in \mathcal{P}^+$, satisfying

(38) $|\partial^\alpha[J^+_x(\theta^{Q_k}_{\mathcal{A}^-_0}) - \{J^+_x(\theta^{Q_k}_{\mathcal{A}^-_0})\}_{\mathsf{approx}}](x)| \leq 2^{CS_0 - \bar{S}}$ for $|\alpha| \leq m$, $1 \leq k \leq k_{\max}$.

Recall that Algorithm PU computes the numbers $\partial^\alpha\{J_x^+(\theta_{\mathcal{A}_0^-}^{Q_k})\}_{\mathsf{approx}}(x)$ for $|\alpha| \le m$; in particular, those numbers are machine numbers.

Since $\delta_{Q_k} \ge 2^{-CS_0}$ (because $Q_k \in CZ(\mathcal{A}_0^-)$), our estimates for derivatives of $\theta_{\mathcal{A}}^{Q}$ yield $|\partial^\alpha \theta_{\mathcal{A}_0^-}^{Q_k}(x)| \le C\delta_{Q_k}^{-|\alpha|} \le 2^{C'S_0}$ for $|\alpha| \le m$, $1 \le k \le k_{\max}$, hence

$$(39)\quad |\partial^\alpha[J_x^+(\theta_{\mathcal{A}_0^-}^{Q_k})](x)| \le 2^{CS_0} \text{ for } |\alpha| \le m,\ 1 \le k \le k_{\max}.$$

We can now use (35), (36), (38), (39) to conclude that

$$(40)\quad \left|\partial^\alpha\left[\sum_{k=1}^{k_{\max}} J_x^+(\theta_{\mathcal{A}_0^-}^{Q_k}) \odot_x^+ J_x^+(F_{Q_k}) - \sum_{k=1}^{k_{\max}} \{J_x^+(\theta_{\mathcal{A}_0^-}^{Q_k})\}_{\mathsf{approx}} \odot_x^+ f_k\right](x)\right| \le 2^{CA_3(\mathcal{A}_0^-)S_0 - \bar{S}/2} \text{ for } |\alpha| \le m.$$

(Here again, we have used the fact that $|x| \le 2^{CS_0}$, to pass from (35), (36) to corresponding estimates for derivatives at x.)

From (35), (36), (38) , (39), we have also

$$\left|\partial^\alpha[\{J_x^+(\theta_{\mathcal{A}_0^-}^{Q_k})\}_{\mathsf{approx}}](x)\right| \le 2^{CS_0} \quad\text{and}\quad |\partial^\alpha f_k(x)| \le 2^{CA_3(\mathcal{A}_0^-)S_0} \quad \text{for } |\alpha| \le m.$$

Consequently, when we execute Lines 16, 17 of the Main Algorithm, we obtain a polynomial $f_x(\mathcal{A}_0, Q_0, x_0, P_0)$ that satisfies

$$(41)\quad \left|\partial^\alpha[f_x(\mathcal{A}_0, Q_0, x_0, P_0) - \sum_{k=1}^{k_{\max}} \{J_x^+(\theta_{\mathcal{A}_0^-}^{Q_k})\}_{\mathsf{approx}} \odot_x^+ f_k](x)\right| \le 2^{CA_3(\mathcal{A}_0^-)S_0 - \bar{S}/2}$$
$$\text{for } |\alpha| \le m.$$

(Here again, we use $|x| \le 2^{CS_0}$.)

Combining (31), (40), and (41), we find that

$$\left|\partial^\alpha[J_x^+(F) - f_x(\mathcal{A}_0, Q_0, x_0, P_0)](x)\right| \le 2^{CA_3(\mathcal{A}_0^-)S_0 - \bar{S}/2} \text{ for } |\alpha| \le m\,.$$

Again using $|x| \le 2^{CS_0}$, we deduce that

$$(42)\quad \left|\partial^\alpha[J_x^+(F) - f_x(\mathcal{A}_0, Q_0, x_0, P_0)](0)\right| \le 2^{C'A_3(\mathcal{A}_0^-)S_0 - \bar{S}/2} \text{ for } |\alpha| \le m\,.$$

Since $A_3(\mathcal{A}_0) \ge C'A_3(\mathcal{A}_0^-)$, we conclude from (42) that (IV) of the Main Lemma for $\mathcal{A}_0$ holds.

Thus, we have shown that our $F \in C^m((1 + c_G)Q_0)$ satisfies all the properties (I),...,(IV).

The proof of the Main Lemma is complete. ■

57. Applications of the main lemma

In this section, we prove an analogue of Theorem 4 that relates to our model of computation. We assume here the model of computation from Section 38, as well as (SU1), (SU2) from Section 49. In this section, c, C, C', etc. stand for constants depending only on m and n.

Theorem 9. *Suppose we are given the following data:*

- *A finite set $E \subset \mathbb{R}^n$, such that*
  - $*$ *For $x \in E$, we have that $x = (x_1, ..., x_n)$ where each x_i is a machine number with $|x_i| \leq 2^{S_0}$.*
  - $*$ *For $x, y \in E$ such that $x \neq y$ we have $|x - y| \geq 2^{-S_0}$.*
- *For each $x \in E$, two real numbers $f(x)$ and $\sigma(x)$, such that*
  - $*$ *$f(x)$ is a machine number that satisfies $|f(x)| \leq 2^{S_0}$.*
  - $*$ *$\sigma(x)$ is a machine number that satisfies $2^{-S_0} \leq \sigma(x) \leq 2^{S_0}$.*

Assume that $\#(E) = N$. Then, there exists $F \in C^m(\mathbb{R}^n)$ with the following properties:

(A) *If $\tilde{F} \in C^m(\mathbb{R}^n)$ and $2^{-S_0} \leq M \leq 2^{S_0}$ satisfy*

$$\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq M \text{ and } |\tilde{F}(x) - f(x)| \leq M\sigma(x) \text{ for } x \in E,$$

then

$$\| F \|_{C^m(\mathbb{R}^n)} \leq CM \text{ and } |F(x) - f(x)| \leq CM\sigma(x) \text{ for } x \in E.$$

(B) *There is an algorithm, in our model of computation, that receives the given data, performs one-time work, and then responds to queries.*

A query consists of a point $x = (x_1, ..., x_n) \in \mathbb{R}^n$, such that x_i is a machine number with $|x_i| \leq 2^{S_0}$ for all $1 \leq i \leq n$. The response to the query is the family of coefficients $\left(\partial^\alpha \bar{P}_x(0)\right)_{|\alpha| \leq m}$ of a polynomial $\bar{P}_x$ that satisfies

$$|\partial^\alpha \left(J_x^+(F) - \bar{P}_x\right)(0)| \leq 2^{-\bar{S}/4} \quad \text{for all } |\alpha| \leq m.$$

The one-time work takes $CN \log N$ operations and CN storage. The work to answer a query is $C \log N$.

Here, C is a constant depending only on m and n.

We will make use of the Main Lemma for $\emptyset$ from Section 55. Recall that $CZ(\emptyset)$ consists of all dyadic cubes of sidelength A_2^{-1}. The following lemma is an immediate consequence of the Main Lemma for $\emptyset$ from Section 55, and of Lemma 1 from Section 55.

Lemma 1. *Suppose that* $Q_0 \subset \mathbb{R}^n$ *is a dyadic cube of sidelength* A_2^{-1}, $x_0 \in E \cap Q_0^*$ *and* $2^{-S_0} \leq M_0 \leq A_1(\emptyset)2^{+S_0}$. *Let* $P_0 \in \mathcal{P}$ *be such that*

(1) $\partial^\alpha P_0(0)$ *is a machine number that satisfies* $|\partial^\alpha P_0(0)| \leq 2^{A_1(\emptyset)S_0}$ *for all* $|\alpha| \leq m-1$.

(2) $P_0 \in \Gamma(x_0, \ell_*, M_0)$.

Then, there exists

$$F \in C^m((1+c_G)Q_0)$$

with the following properties:

(3) $|\partial^\beta(F-P_0)(x)| \leq CA_3(\emptyset)M_0$ *for all* $x \in (1+c_G)Q_0, |\beta| \leq m$.

(4) $J_x(F) \in \Gamma(x, 0, CA_3(\emptyset)M_0)$ *for all* $x \in E \cap (1+c_G)Q_0$.

(5) *Let* $x = (x_1, \ldots, x_n) \in (1+c_G)Q_0$ *be such that* x_i *is a machine number for all* $1 \leq i \leq n$. *Then,*

$$\left|\partial^\alpha\left[J_x^+(F) - f_x(\emptyset, Q_0, x_0, P_0)\right](0)\right| \leq 2^{A_3(\emptyset)S_0 - \bar{S}/2} \textit{for all } |\alpha| \leq m,$$

where $f_x(\emptyset, Q_0, x_0, P_0)$ *is computed by the* Main Algorithm *in Section* 55.

(6) *Let* $x = (x_1, \ldots, x_n) \in (1+c_G)Q_0$ *be such that* x_i *is a machine number for all* $1 \leq i \leq n$. *Then the numbers*

$$\partial^\alpha f_x(\emptyset, Q_0, x_0, P_0)(0),$$

for $|\alpha| \leq m$, *are all machine numbers in the range* $[-2^{A_3(\emptyset)S_0}, 2^{A_3(\emptyset)S_0}]$.

Here, $C > 0$ *is a constant depending only on* m *and* n.

Let us describe the algorithm promised in Theorem 9. We perform all the one-time work described in Sections 48, 50, 51, 52. In addition, in the one-time work, we subdivide $\mathbb{R}^n$ into dyadic cubes of sidelength A_2^{-1}. Let Ω_0 be the set all dyadic cubes Q of sidelength A_2^{-1}, such that $E \cap Q^* \neq \emptyset$. As in Section 34, we have $\#(\Omega_0) \leq CN$. We compute the list Ω_0, and then sort the cubes in Ω_0 with respect to the lexicographic order on the centers of the cubes. We store the sorted list in memory, during the one-time work. This requires $CN \log N$ operations and CN storage.

For each $Q \in \Omega_0$, we compute a representative $x_Q :=$ Find-Representative(Q). Then $x_Q \in E \cap Q^*$, by the defining property of Find-Representative from Section 25. We store the CN representatives in memory.

Next, for each $Q \in \Omega_0$ we will compute a polynomial $P_Q \in \mathcal{P}$ having the following properties:

(7) $\partial^\alpha(P_Q)(0)$ is a machine number in the range $[-2^{A_1(\emptyset)S_0}, 2^{A_1(\emptyset)S_0}]$ for all $|\alpha| \leq m-1$.

(8) Suppose $P \in \mathcal{P}$ and $M \geq 2^{-S_0}$ satisfy $P \in \Gamma(x_Q, \ell_*, M)$.
Then $P_Q \in \Gamma(x_Q, \ell_*, CM)$.

We may compute the machine numbers $(\partial^\alpha(P_Q)(0))_{|\alpha|\leq m-1}$ thanks to the discussion at the end of Section 53. (We assume, as we may, that $A_1(\emptyset)$ exceeds the constant C' from that discussion.) We compute and store all the polynomials P_Q in memory, during the one-time work. The total time required for the computation of the points x_Q and the polynomials P_Q does not exceed $CN \log N$, and the storage required is no more than CN.

This completes the description of the one-time work of our algorithm. The resources being spent for the one-time work are bounded by $CN \log N$ computer operations and CN storage, for C depending only on m and n.

We move to describe the query algorithm. Suppose we are given a point $x \in \mathbb{R}^n$, whose coordinates are machine numbers with absolute values that do not exceed 2^{S_0}. We set $\Omega_0(x) = \{Q \in \Omega_0 : x \in (1+c_G)Q\}$. Note that $\#(\Omega_0(x)) \leq C$. Given the one-time work, we may compute $\Omega_0(x)$ using at most $C \log N$ operations, by applying binary searches on the sorted list Ω_0 (details in Section 34).

Once $\Omega_0(x)$ is obtained, the algorithm computes the polynomial $\bar{P}_x$, which is our machine approximation to

$$(9) \qquad \sum_{Q \in \Omega_0(x)} J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ f_x(\emptyset, Q, x_Q, P_Q).$$

Let us elaborate on the computation of the approximation to (9). For each $Q \in \Omega_0(x)$ denote $f_Q = f_x(\emptyset, Q, x_Q, P_Q)$. Note that from (7), we see that x, Q, x_Q, P_Q satisfy the conditions in the comment "Inputs are as follows" in Section 55. Hence, f_Q is well-defined, and we may apply the Main Algorithm from Section 55 to compute $(\partial^\alpha f_Q(0))_{|\alpha|\leq m}$. Next, in order to carry out the $\odot_x^+$-operation, we compute the machine approximation to $(\partial^\alpha f_Q(x))_{|\alpha|\leq m}$. According to (6), we have that $|\partial^\alpha f_Q(0)| \leq 2^{A_3(\emptyset)S_0}$ for all $|\alpha| \leq m$, and therefore also our machine approximations $(\partial^\alpha f_Q(x))_{\text{approx}}$ are never larger than $2^{CA_3(\emptyset)S_0}$ in absolute value. Using Algorithm PU from Section 54 we may compute machine numbers that approximate

$$(10) \quad \partial^\alpha\left(J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ f_Q\right)(x) \quad \text{for all} \quad |\alpha| \leq m.$$

The machine approximations are all smaller than $2^{CA_3(\emptyset)S_0}$ in absolute value, and they differ from the actual quantities in (10) by no more than $2^{C'A_3(\emptyset)S_0-\bar{S}/2}$. Summing over $Q \in \Omega_0(x)$, we compute numbers that are machine approximations to

$$(11)\quad \partial^\alpha\Bigg(\sum_{Q\in\Omega_0(x)} J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ f_Q\Bigg)(x) \text{ for all } |\alpha| \le m.$$

Since $\#(\Omega_0(x)) \le C$, our machine approximations are all bounded by $2^{C'A_3(\emptyset)S_0}$, and differ from (11) by at most $2^{\check{C}A_3(\emptyset)S_0-\bar{S}/2}$, according to (SU1) and (SU2). Finally, we compute from our approximation to (11) additional machine numbers, which are our approximations to

$$(12)\quad \partial^\alpha\Bigg(\sum_{Q\in\Omega_0(x)} J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ f_Q\Bigg)(0) \text{ for all } |\alpha| \le m.$$

Since $|x| \le 2^{CS_0}$, then our machine approximations to (12) are smaller than $2^{\hat{C}A_3(\emptyset)S_0}$, and differ from (12) by no more than $2^{\hat{C}A_3(\emptyset)S_0-\bar{S}/2}$. Thus we are able to compute an approximation to the polynomial in (9), as we computed good approximations to all of its derivatives at zero. This completes the description of the query algorithm. The amount of work needed to carry out the query is bounded by $C\log N$, given the one-time work.

Remark. It is equally easy to produce, in Theorem 9, an approximation to the derivatives of the polynomial $J_x^+(F)$ at the point x, rather than at zero. We just observe that $\bar{S} \ge CS_0$ and $|x| \le 2^{CS_0}$, hence the roundoff errors caused by translating from 0 to x do not hurt.

It remains to prove that the polynomials in (9) satisfy the conclusions of Theorem 9. This is essentially the content of the following lemma.

Lemma 2. *Let E, f, σ be as in Theorem 9. Then, there exists $F \in C^m(\mathbb{R}^n)$ for which the following holds: Suppose $2^{-S_0} \le M \le (A_1(\emptyset))^{1/2}2^{S_0}$ satisfies that*

$$(13)\quad \Gamma(x, \ell_*, M) \ne \emptyset \textit{ for all } x \in E.$$

Then,

$$(14)\quad |F(x) - f(x)| \le C' A_2^m A_3(\emptyset) M\sigma(x) \textit{ for all } x \in E,$$

$$(15)\quad \| F \|_{C^m(\mathbb{R}^n)} \le C' A_2^m A_3(\emptyset) M,$$

(16) *For any $x \in \mathbb{R}^n$ whose coordinates are machine numbers whose absolute value is smaller than 2^{S_0}, we have*

$$|\partial^\alpha\left(J_x^+(F) - \bar{P}_x\right)(0)| \le 2^{C'A_3(\emptyset)S_0-\bar{S}/2} \quad \textit{for all } |\alpha| \le m,$$

where $\bar{P}_x$ is defined in (9).

Here, $C' > 0$ denotes a constant depending only on m and n.

Proof. Recall that for any cube $Q \in \Omega_0$, we have defined a point $x_Q \in Q^* \cap E$, and a polynomial $P_Q \in \mathcal{P}$ such that $\partial^\alpha P_Q(0)$ is a machine number whose absolute value does not exceed $2^{A_1(\emptyset)S_0}$, for all $|\alpha| \leq m-1$ (see (7)). Fix $Q \in \Omega_0$. By our assumption (13), we have

(17) $\Gamma(x_Q, \ell_*, M) \neq \emptyset.$

According to (8) and (17), we know that

(18) $P_Q \in \Gamma(x_Q, \ell_*, C'M) \subset CMB(x_Q, 1)$

where the last inclusion follows from Property 4 from Section 13, and the definition of $\Gamma(x_Q, 0, M)$. We will now invoke Lemma 1, for the quantity $C'M$, the cube Q, the point $x_Q \in E \cap Q^*$ and the polynomial P_Q, based on (18). By the conclusion of Lemma 1, there exists $F_Q \in C^m((1+c_G)Q)$, with the following properties:

(19) $|\partial^\beta(F_Q - P_Q)(x)| \leq CA_3(\emptyset)M$ for all $|\beta| \leq m, x \in (1+c_G)Q$.

(20) $J_x(F_Q) \in \Gamma(x, 0, CA_3(\emptyset)M)$ for all $x \in E \cap (1+c_G)Q$,

(21) Let $x \in (1+c_G)Q_k$ be a point whose coordinates are machine numbers. Then,

$$|\partial^\alpha \left(J_x^+(F_Q) - f_x(\emptyset, Q, x_Q, P_Q)\right)(0)| \leq 2^{A_3(\emptyset)S_0 - \bar{S}/2}.$$

(22) Let $x \in (1+c_G)Q$ be a point whose coordinates are machine numbers. Then the quantities $\partial^\alpha f_x(\emptyset, Q, x_Q, P_Q)(0)$, for $|\alpha| \leq m$, are all machine numbers in the range $[-2^{A_3(\emptyset)S_0}, 2^{A_3(\emptyset)S_0}]$.

We define a function $F : \mathbb{R}^n \to \mathbb{R}$ by setting

(23) $F(x) = \sum_{Q \in \Omega_0} \theta_Q^\emptyset(x) F_Q(x).$

Since $\mathrm{Supp}(\theta_Q^\emptyset) \subset (1 + c_G/2)Q$ and $F_Q \in C^m((1+c_G)Q)$, then F is a well-defined $C^m(\mathbb{R}^n)$-function. For any $x \in \mathbb{R}^n$, we have that $x \in \mathrm{Supp}(\theta_Q^\emptyset)$ only for $Q \in \Omega_0(x)$. Therefore (23) implies that

(24) $J_x^+(F) = \sum_{Q \in \Omega_0(x)} J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ J_x^+(F_Q).$

For any $Q \in \Omega_0$ we have $\delta_Q = A_2^{-1}$, and by the explanation in Section 54,

(25) $|\partial^\beta(\theta_Q^\emptyset)(x)| \leq CA_2^m$ for all $|\beta| \leq m$ and $x \in \mathbb{R}^n$.

Since $\#(\Omega_0(x)) \leq C$, then (18), (19), (24), (25) imply that

(26) $|\partial^\beta F(x)| < CA_2^m A_3(\emptyset)M$ for all $|\beta| \leq m, x \in \mathbb{R}^n$.

Thus, (15) is proven.

Recall the definition (9) of $\bar{P}_x$, and also the discussion following that definition. We have that, for all $|\alpha| \leq m$,

$$(27)\quad \left|\partial^\alpha\Bigg(\bar{P}_x - \sum_{Q\in\Omega_0(x)} J_x^+\left(\theta_Q^\emptyset\right) \odot_x^+ f_x(\emptyset, Q, x_Q, P_Q)\Bigg)(0)\right| \leq 2^{CA_3(\emptyset)S_0 - \bar{S}/2}$$

By (21), (24), (25) and (27) we conclude that

$$(28)\quad \left|\partial^\alpha\left(J_x^+(F) - \bar{P}_x\right)(0)\right| \leq 2^{\hat{C}A_3(\emptyset)S_0-\bar{S}/2} \text{ for all } |\alpha| \leq m.$$

This proves (16). Next, we prove (14). Fix $x \in E$ and a cube $Q \in \Omega_0(x)$. Then, since the θ's are a partition of unity,

$$(29)\quad J_x(F) = J_x(F_Q) + M \sum_{Q_\nu\in\Omega_0(x)} J_x\left(\theta_{Q_\nu}^\emptyset \cdot \frac{F_{Q_\nu} - F_Q}{M}\right).$$

According to (20) and to Property 1 from Section 13, we know that

$$J_x\left(\frac{F_{Q_\nu} - F_Q}{M}\right) \in C'A_3(\emptyset)\sigma(x,0).$$

By (18) and (19), we have that

$$J_x\left(\frac{F_{Q_\nu} - F_Q}{M}\right) \in C'A_3(\emptyset)B(x,1).$$

Next, we invoke the Whitney t-Convexity of $\sigma(x,0)$, according to Property 3 from Section 13. By the Whitney t-convexity and (25),

$$(30)\quad J_x\left(\theta_{Q_\nu}^\emptyset \cdot \tfrac{F_{Q_\nu}-F_Q}{M}\right) = J_x\left(\theta_{Q_\nu}^\emptyset\right) \odot_x J_x\left(\tfrac{F_{Q_\nu}-F_Q}{M}\right) \in C\,A_2^m A_3(\emptyset)\sigma(x,0).$$

Recall that $\#(\Omega_0(x)) < C$. From (29), (30) and (20), we conclude that

$$(31)\quad J_x(F) \in \Gamma(x,0,CA_3(\emptyset)M) + C'A_2^m A_3(\emptyset)M\sigma(x,0) \subset \Gamma(x,0,\check{C}A_2^m A_3(\emptyset)M).$$

Now (14) follows from (31). The lemma is thus proven. ■

Proof of Theorem 9. Let $2^{-S_0} \leq M \leq 2^{S_0}$ and suppose that $\tilde{F} \in C^m(\mathbb{R}^n)$ satisfies

$$(32)\quad \|\, \tilde{F}\, \|_{C^m(\mathbb{R}^n)} \leq M \text{ and } |\tilde{F}(x) - f(x)| \leq M\sigma(x) \text{ for all } x \in E.$$

According to Property 0 from Section 13,

$$(33)\quad \Gamma(x,\ell_*,CM) \neq \emptyset \text{ for all } x \in E.$$

Assume, as we may, that $(A_1(\emptyset))^{1/2} \geq C$ for C being the constant from (33). Now, (A) and (B) follow from Lemma 2, and from the description of the algorithm above, since A_2 and $A_3(\emptyset)$ are constants depending only on m and n. The theorem is thus proven. ■

Recall that $\bar{S}$ is the precision of our model of computation, as presented in Section 38. Recall from Section 38 that we are interested only in the order of magnitude of $\bar{S}$; it is possible to simulate a model of computation with $C\bar{S}$-bit machine numbers, using a model of computation that is based on $\bar{S}$-bit machine numbers. The work to simulate a single $C\bar{S}$-bit operation, using a model of computation able to work only with $\bar{S}$-bit numbers, is bounded by C' that depends solely on C.

Recall from Section 49, that our only assumptions on $\bar{S}$ and S_0 are that $\bar{S} \geq CS_0$ and $S_0 > C$, for some constant C depending only on m and n (since $A_0, p_\#$ depend only on m and n). As we are interested only in the order of magnitude of $\bar{S}$, it is possible to deduce a variant of Theorem 9, in which S_0, the precision in which the input is given, equals $\bar{S}$, the precision of our model of computation. We conclude the following theorem.

Theorem 10. *Assume the model of computation from Section* 38*. Suppose we are given the following data:*

- *A finite set* $E \subset \mathbb{R}^n$*, such that for any* $x = (x_1, ..., x_n) \in E$*, we have that each* x_i *is a machine number.*
- *For each* $x \in E$*, two real numbers* $f(x)$ *and* $\sigma(x)$*, such that* $f(x), \sigma(x)$ *are machine numbers, and* $\sigma(x) > 0$*.*

Assume that $\#(E) = N$*. Then, there exists* $F \in C^m(\mathbb{R}^n)$ *with the following properties*:

(A) *If* $\tilde{F} \in C^m(\mathbb{R}^n)$ *and* $M > 0$ *is a machine number such that*

$$\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq M \quad \textit{and} \quad |\tilde{F}(x) - f(x)| \leq M\sigma(x) \quad \textit{for } x \in E,$$

then

$$\| F \|_{C^m(\mathbb{R}^n)} \leq CM \quad \textit{and} \quad |F(x) - f(x)| \leq CM\sigma(x) \quad \textit{for } x \in E.$$

(B) *There is an algorithm, in our model of computation, that receives the given data, performs one-time work, and then responds to queries.*

A query consists of a point $x = (x_1, ..., x_n) \in \mathbb{R}^n$*, such that* x_i *is a machine number for all* $1 \leq i \leq n$*. The response to the query is a family of machine numbers* $(q_\alpha)_{|\alpha| \leq m}$ *such that*

$$|\partial^\alpha F(x) - q_\alpha| \leq 2^{-\bar{S}} \textit{ for all } |\alpha| \leq m.$$

(*It might happen that* $|\partial^\alpha F(x)| > 2^{\bar{S}}$ *for some* α*, and consequently,* $\partial^\alpha F(x)$ *cannot be approximated by a machine number. In that case, we output machine numbers* $m_0, ..., m_k$*, with* $k \leq C$*, such that* $\left|\partial^\alpha F(x) - \sum_{i=0}^{k} 2^{i\bar{S}} m_i\right| \leq 2^{-\bar{S}}$.)

The one-time work takes $CN \log N$ *operations and* CN *storage. The work to answer a query is* $C \log N$.

Here, C *is a constant depending only on* m *and* n.

Indeed, Theorem 10 follows immediately from Theorem 9 and the discussion in Section 38, once we observe that for a machine number a, the condition $a > 0$ is equivalent to $a \geq 2^{-\bar{S}}$.

References

[1] Arya, S., Mount, D., Netanyahu, N., Silverman, R. and Wu, A.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* **45** (1998), no. 6, 891–923.

[2] Bierstone, E., Milman, P. and Pawłucki, W.: Differentiable functions defined on closed sets. A problem of Whitney. *Invent. Math.* **151** (2003), no. 2, 329–352.

[3] Bierstone, E., Milman, P. and Pawłucki, W.: Higher-order tangents and Fefferman's paper on Whitney's extension problem. *Ann. of Math. (2)* **164** (2006), no. 1, 361–370.

[4] Brudnyi, A. and Brudnyi, Y.: Metric spaces with linear extensions preserving Lipschitz condition. *Amer. J. Math.* **129** (2007), no. 1, 217–314.

[5] Brudnyi, Y.: On an extension theorem. *Funk. Anal. i Prilzhen.* **4** (1970), 97–98; English transl. in *Func. Anal. Appl.* **4** (1970), 252–253.

[6] Brudnyi, Y. and Shvartsman, P.: The traces of differentiable functions to closed subsets of $\mathbb{R}^n$. In *Function Spaces (Poznán, 1989)*, 206–210. Teubner-Texte Math. **120**. Teubner, Stuttgart, 1991.

[7] Brudnyi, Y. and Shvartsman, P.: A linear extension operator for a space of smooth functions defined on closed subsets of R^n. *Dokl. Akad. Nauk SSSR* **280** (1985), 268-270. English transl. in *Soviet Math. Dokl.* **31** (1985), no. 1, 48–51.

[8] Brudnyi, Y. and Shvartsman, P.: Generalizations of Whitney's extension theorem. *Int. Math. Research Notices* **3** (1994), 129–139.

[9] Brudnyi, Y. and Shvartsman, P.: The Whitney problem of existence of a linear extension operator. *J. Geom. Anal.* **7** (1997), no. 4, 515–574.

[10] Brudnyi, Y. and Shvartsman, P.: Whitney's extension problem for multivariate $C^{1,w}$ functions. *Trans. Amer. Math. Soc.* **353** (2001), no. 6, 2487–2512.

[11] Callahan, P. B. and Kosaraju, S. R.: A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. ACM* **42** (1995), no. 1, 67–90.

[12] Fefferman, C.: Interpolation and extrapolation of smooth functions by linear operators. *Rev. Mat. Iberoamericana* **21** (2005), no. 1, 313–348.

[13] Fefferman, C.: A sharp form of Whitney's extension theorem. *Ann. of Math. (2)* **161** (2005), 509–577.

[14] Fefferman, C.: Whitney's extension problem for C^m. *Ann. of Math. (2)* **164** (2006), no. 1, 313–359.

[15] Fefferman, C.: Whitney's extension problem in certain function spaces. (preprint).

[16] Fefferman, C.: A generalized sharp Whitney theorem for jets. *Rev. Mat. Iberoamericana* **21** (2005), no. 2, 577–688.

[17] Fefferman, C.: Extension of $C^{m,\omega}$ smooth functions by linear operators. *Rev. Mat. Iberoamericana* **25** (2009), no. 1, 1–48.

[18] Fefferman, C.: C^m extension by linear operators *Ann. of Math. (2)* **166** (2007), no. 3, 779–835.

[19] Fefferman, C. and Klartag, B.: Fitting a C^m-smooth function to data I. *Annals of Math. (2)*, to appear.

[20] Fefferman, C.: Fitting a C^m-smooth function to data III. (preprint).

[21] Glaeser, G.: Étude de quelques algèbres tayloriennes. *J. Analyse Math.* **6** (1958), 1–124.

[22] Har-Peled, S. and Mendel, M.: Fast construction of nets in low-dimensional metrics, and their applications. *SIAM J. Comput.* **35** (2006), no. 5, 1148–1184.

[23] Hartmanis, J. and Simon, J.: On the power of multiplication in random-access machines. Proc. 15th Annu. IEEE Sympos. *Switching Automata Theory* (1974), 13–23.

[24] Kahan, S. and Snoeyinkm, J.: On the bit complexity of minimum link paths: Superquadratic algorithms for problem solvable in linear time. *Comput. Geom.* **12** (1999), no. 1, 33–44.

[25] Knuth, D.: *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3$^{\text{rd}}$ edition. Addison-Wesley, 1997.

[26] Malgrange, B.: *Ideals of Differentiable Functions.* Oxford University Press, 1966.

[27] Preparata, F. P. and Shamos, M. I.: *Computational Geometry: An introduction*, 2$^{\text{nd}}$ edition. Texts and Monographs in Computer Science. Springer-Verlag, New York, 1985.

[28] Schönhage, A.: On the power of random access machines. In *Proc. 6th. Internat. Colloq. Automata Lang. Program.*, 520–529. Lecture Notes Comput. Sci, Vol. 71. Springer-Verlag, 1979.

[29] Shvartsman, P.: Lipschitz selections of multivalued mappings and traces of the Zygmund class of functions to an arbitrary compact. *Dokl. Acad. Nauk SSSR* **276** (1984), 559-562; English transl. in *Soviet Math. Dokl.* **29** (1984), 565–568.

[30] Shvartsman, P.: On traces of functions of Zygmund classes. *Sibirskyi Mathem. J.* **28** **N5** (1987), 203–215; English transl. in *Siberian Math. J.* **28** (1987), 853–863.

[31] Shvartsman, P.: Lipschitz selections of set-valued functions and Helly's theorem. *J. Geom. Anal.* **12** (2002), no. 2, 289–324.

[32] Stein, E. M.: *Singular Integrals and Differentiability Properties of Functions.* Princeton Univ. Press, 1970.

[33] Tarski, A.: *A Decision Method for Elementary Algebra and Geometry.* U. of California Press, 1951.

[34] 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 Annals of the History of Computing* **15** (1993), no. 4, 27–75.

[35] Whitney, H.: Analytic extensions of differentiable functions defined in closed sets. *Trans. Amer. Math. Soc.* **36** (1934), 63–89.

[36] Whitney, H.: Differentiable functions defined in closed sets I. *Trans. Amer. Math. Soc.* **36** (1934), 369–389.

[37] Whitney, H.: Functions differentiable on the boundaries of regions. *Ann. of Math.* **35** (1934), 482–485.

[38] Zobin, N.: Whitney's problem on extendability of functions and an intrinsic metric. *Adv. Math.* **133** (1998), no. 1, 96–132.

[39] Zobin, N.: Extension of smooth functions from finitely connected planar domains. *J. Geom. Anal.* **9** (1999), no. 3, 489–509.

Recibido: 26 de mayo de 2006

Charles Fefferman
Department of Mathematics
Princeton University, Fine Hall
Washington Road
Princeton, New Jersey 08544, USA
cf@math.princeton.edu

Bo'az Klartag
School of Mathematics
Institute for Advanced Study
Einstein Drive
Princeton, New Jersey 08540, USA
klartag@ias.edu

First author is partially supported by NSF grant #DMS-0245242. Second author is supported by the Clay Mathematics Institute and by NSF grant #DMS-0456590.