Rev. Mat. Iberoamericana **25** (2009), no. 1, 275–421

The C^m Norm of a Function with Prescribed Jets II

Charles Fefferman

Abstract

We give algorithms to compute a function F on $\mathbb{R}^n$, having prescribed Taylor polynomials (or taking prescribed values) at N given points, with the C^m-norm of F close to least possible.

Contents

2000 Mathematics Subject Classification: 65D05, 65D17.
Keywords: Interpolation, Whitney extension theorem, algorithm.

0. Introduction

Fix $m, n \geq 1$. We compute the least possible (infimum) C^m-norm of a function $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ having prescribed $m^{\underline{th}}$ order Taylor polynomials at N given points. Also, given $\epsilon > 0$, we compute such an F, whose C^m-norm is within ϵ percent of least possible. Our computation consists of an algorithm, to be implemented on an (idealized) digital computer. Given ϵ and N as above, our computation uses at most $\exp(C/\epsilon)N \log N$ computer operations, where C is a "controlled constant" (see below).

We also present preliminary results on the more difficult problem of computing the least possible (infimum) C^m-norm of a function F taking prescribed values at N given points. Again, given $\epsilon > 0$, we compute such an F, whose C^m-norm is within ϵ percent of the least possible. This time, our computation uses $\exp(C/\epsilon)N^5(\log N)^2$ operations. Surely this result is not optimal; we look forward to future improvements.

In previous work [22, 23], Fefferman-Klartag computed functions F having prescribed values, or prescribed Taylor polynomials, at N given points, with the C^m-norm of F having the least possible "order of magnitude". Our goal here is to gain precision, by passing from "orders of magnitude" to errors of at most ϵ percent, for an arbitrarily small given ϵ.

To state our results precisely, we must say exactly what we mean by "the C^m-norm", an "idealized digital computer", and "computing a function". Let us start with the C^m-norm.

If $\Omega \subset \mathbb{R}^n$ is open, then $C^m(\Omega)$ denotes the space of functions $F : \Omega \to \mathbb{R}$, whose derivatives through order m are continuous and bounded on Ω. For $F \in C^m(\Omega)$ and $x \in \Omega$, we write $J_x(F)$ (the "jet" of F at x) to denote the $m^{\underline{th}}$ order Taylor polynomial of F at x. Thus, $J_x(F)$ belongs to $\mathcal{P}$, the vector space of all (real) $m^{\underline{th}}$ degree polynomials on $\mathbb{R}^n$.

There are many equivalent norms on $C^m(\Omega)$, e.g.,

$$\| F \| = \sup_{x \in \Omega} \max_{|\alpha| \leq m} |\partial^\alpha F(x)|, \tag{1}$$

or

$$\| F \| = \sup_{x \in \Omega} \left(\sum_{|\alpha| \leq m} \frac{|\alpha|!}{\alpha!} |\partial^\alpha F(x)|^2 \right)^{1/2}. \tag{2}$$

To fix a particular norm on C^m, we assume throughout this paper that, for each $x \in \mathbb{R}^n$, we are given a norm $| \cdot |_x$ on the vector space $\mathcal{P}$. These norms are subject to restrictions, to be spelled out in Section 3 below. We then define

$$\| F \|_{C^m(\Omega)} = \sup_{x \in \Omega} |J_x(F)|_x. \tag{3}$$

For instance, the norms (1) and (2) arise by taking

$$|P|_x = \max_{|\alpha| \leq m} |\partial^\alpha P(x)| \tag{4}$$

or

$$|P|_x = \left(\sum_{|\alpha| \leq m} \frac{|\alpha|!}{\alpha!} |\partial^\alpha P(x)|^2 \right)^{1/2}, \tag{5}$$

respectively, for $P \in \mathcal{P}$, $x \in \mathbb{R}^n$.

Throughout this paper, we assume that the C^m-norm is given by (3).

In Section 3 below, we introduce the notion of a "controlled constant". In this introduction, it is enough to note that a controlled constant may depend only on m, n and the choice of the family of norms $| \cdot |_x$ ($x \in \mathbb{R}^n$) in (3). Throughout this paper, we write c, C, C', etc. to denote controlled constants. These letters may denote different controlled constants in different occurrences.

If $X, Y \geq 0$ are real numbers computed from input data (such as prescribed Taylor polynomials at N given points), then we say that X and Y have "the same order of magnitude", provided we have $cX \leq Y \leq CX$ for controlled constants c and C. To "compute the order of magnitude of X" is to compute some $Y \geq 0$ such that X and Y have the same order of magnitude.

Next, we discuss our "idealized digital computer". Our computer has standard Von Neumann architecture (see, e.g., [39]), but we assume here that the computer can store, and perform elementary operations, on exact real numbers, without roundoff error.

The elementary operations include addition, multiplication, comparison (deciding whether two given numbers x and y satisfy $x < y$), exponentiation and logarithms, and the "greatest integer" function. Each of these elementary operations takes one unit of "work". See Section 2 below, for a more careful discussion of this model of computation and its pitfalls.

Our computer will need to acquire information on the family of norms $|\cdot|_x$ on $\mathcal{P}$, which we used in (3) to define the C^m-norm. We assume that our computer has access to an Oracle. Given $P \in \mathcal{P}$, $x \in \mathbb{R}^n$, and $\epsilon > 0$, the Oracle returns a real number $\mathcal{N}_\epsilon(P, x)$, guaranteed to satisfy $(1+\epsilon)^{-1}\mathcal{N}_\epsilon(P,x) \leq |P|_x \leq (1+\epsilon)\mathcal{N}_\epsilon(P,x)$.

Each time our computer obtains an answer $\mathcal{N}_\epsilon(P, x)$ from the Oracle, we are charged $\exp(C/\epsilon)$ units of "work".

Note that our assumptions regarding the Oracle are quite conservative. For instance, for the norms given by (4) or (5), we can compute $|P|_x$ exactly in at most C operations, for any given $P \in \mathcal{P}$, $x \in \mathbb{R}^n$.

This concludes our introductory remarks on the "idealized digital computer".

We still need to explain what we mean by "computing a function". We have in mind the following dialogue with the computer. First, we enter the input data for our problem (e.g., a number $\epsilon > 0$, points $x_1, \ldots, x_N \in \mathbb{R}^n$, and polynomials $P_1, \ldots, P_N \in \mathcal{P}$; we will be "computing a function" $F \in C^m(\mathbb{R}^n)$, such that $J_{x_i}(F) = P_i$ for each $i = 1, \ldots, N$). The computer then executes an algorithm, performing L_0 elementary operations (the "one-time work"). When it has finished the one-time work, the machine signals that it is ready to accept further input. We may then address "queries" to the computer. A "query" consists of a point $\underline{x} \in \mathbb{R}^n$, and the computer responds to each query $\underline{x}$ by executing an algorithm (the "query work", entailing L_1 elementary operations) and returning a polynomial $P_{\underline{x}} \in \mathcal{P}$. We say that our algorithms "compute the function" $F \in C^m(\mathbb{R}^n)$, provided we have $P_{\underline{x}} = J_{\underline{x}}(F)$ for any query point $\underline{x} \in \mathbb{R}^n$. Here, we demand that F be uniquely determined by the input data. We do not allow "adaptive algorithms" in which query work performed in computing $P_{\underline{x}}$ is stored and used subsequently to compute $P_{\underline{y}}$ for another query point $\underline{y}$. We allow the computer to access the Oracle during the one-time work, but not at query time. The work charged by the Oracle is part of the one-time work L_0.

The computer resources used in "computing a function" F are the one-time work L_0; the query work L_1; and the "space" or "storage", i.e., the number of memory cells in the random-access memory (RAM).

We will be working with algorithms that compute a function F as well as other data (e.g., a real number). We regard any work done in computing the "other data" to be part of the one-time work.

This concludes our explanation of "computing a function".

It will be convenient to introduce the following definitions and notation. A "Whitney field" is a family $\vec{P} = (P^x)_{x \in E}$ of polynomials $P^x \in \mathcal{P}$, indexed by the points x in a finite set $E \subset \mathbb{R}^n$. We say that $\vec{P} = (P^x)_{x \in E}$ is a Whitney field "on E", and we write $\mathcal{Wh}(E)$ for the vector space of all Whitney fields on E. If $\vec{P}$ is a Whitney field on E, and if $S \subset E$, then we define the "restriction" $\vec{P}|_S \in \mathcal{Wh}(S)$ in an obvious way.

If $\vec{P} = (P^x)_{x \in E}$ is a Whitney field, $\Omega \supset E$ is an open set, and $F \in C^m(\Omega)$, then we say that "F agrees with $\vec{P}$", or "F is an extending function for $\vec{P}$", provided $J_x(F) = P^x$ for each $x \in E$.

We define a C^m norm on Whitney fields. If $\vec{P} \in \mathcal{Wh}(E)$, and if $\Omega \supset E$ is an open set, then we define

$$\| \vec{P} \|_{C^m(\Omega)} = \inf \left\{ \| F \|_{C^m(\Omega)} \colon F \in C^m(\Omega)\,,\ F \text{ agrees with } \vec{P} \right\}.$$

Elementary examples show that this infimum need not be a minimum. For a constant $A > 1$, we say that an extending function $F \in C^m(\mathbb{R}^n)$ for a Whitney field $\vec{P}$ is "A-optimal", provided $\| F \|_{C^m(\mathbb{R}^n)} \leq A \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

Similarly, if $f : E \longrightarrow \mathbb{R}$ and $F \in C^m(\Omega)$, with $E \subset \mathbb{R}^n$ finite and $\Omega \supset E$ open, then we say that "F agrees with f", or "F is an extending function for f", provided we have $F = f$ on E. We define a C^m-norm on functions $f : E \longrightarrow \mathbb{R}$, by setting

$$\| f \|_{C^m(\Omega)} = \inf \left\{ \| F \|_{C^m(\Omega)} \colon F \in C^m(\Omega)\,,\ F \text{ agrees with } f \right\}.$$

Again, this infimum needn't be a minimum. For a constant $A > 1$, we say that an extending function $F \in C^m(\mathbb{R}^n)$ for $f : E \longrightarrow \mathbb{R}$ is "A-optimal", provided

$$\| F \|_{C^m(\mathbb{R}^n)} \leq A \| f \|_{C^m(\mathbb{R}^n)}\,.$$

Note that an A-optimal extending function is always defined on the full $\mathbb{R}^n$.

We are now ready to state the main results of this paper. We write $\#(E)$ for the number of elements of a finite set E.

Theorem 1. *Given $0 < \epsilon < 1/2$, and given a Whitney field $\vec{P} \in \mathcal{Wh}(E)$, with $\#(E) = N \geq 1$, an algorithm to be specified below computes a $(1+\epsilon)$-optimal extending function for $\vec{P}$, as well as a number $\mathcal{N}_\epsilon(\vec{P})$, such that*

$$(1 + \epsilon)^{-1} \mathcal{N}_\epsilon(\vec{P}) \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq (1+\epsilon)\, \mathcal{N}_\epsilon(\vec{P})\,.$$

Our algorithm uses one-time work at most $\exp(C/\epsilon) N \log(N+1)$, query work at most $C \log(N/\epsilon)$, and storage at most $\exp(C/\epsilon) N$.

Theorem 2. *Given* $0 < \epsilon < 1/2$*, and given a function* $f : E \longrightarrow \mathbb{R}$*, with* $E \subset \mathbb{R}^n$ *and* $\#(E) = N \geq 1$*, an algorithm to be specified below computes a* $(1+\epsilon)$*-optimal extending function for* f*, as well a number* $\mathcal{N}_\epsilon(f)$*, such that*

$$(1+\epsilon)^{-1}\,\mathcal{N}_\epsilon(f) \leq \| f \|_{C^m(\mathbb{R}^n)} \leq (1+\epsilon)\,\mathcal{N}_\epsilon(f)\,.$$

Our algorithm uses one-time work at most $\exp(C/\epsilon)\,N^5(\log(N+1))^2$*, query work at most* $C\log(N/\epsilon)$*, and storage at most* $\exp(C/\epsilon)N^2$*.*

We would prefer a power of $1/\epsilon$ in place of $\exp(C/\epsilon)$ in Theorem 1, but the N-dependence seems optimal.

The algorithm promised in Theorem 2 is presumably far from optimal.

To set the stage for discussion of the proofs of Theorems 1 and 2, we first recall the previous work of Fefferman-Klartag [22, 23]. There, we proved the following results.

Theorem 3. (Easy) *Given a Whitney field* $\vec{P} \in \mathcal{Wh}(E)$*, with* $\#(E) = N \geq 1$*, an algorithm presented in* [22, 23] *computes a* C*-optimal extending function for* $\vec{P}$*, and a number* $\mathcal{N}(\vec{P})$ *such that*

$$c\mathcal{N}(\vec{P}) \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq C\mathcal{N}(\vec{P}).$$

Our algorithm uses one-time work at most $CN\log(N+1)$*, query work at most* $C\log(N+1)$*, and storage at most* CN*.*

Theorem 4. (Hard) *Given a function* $f : E \longrightarrow \mathbb{R}$*, with* $E \subset \mathbb{R}^n$ *and* $\#(E) = N \geq 1$*, an algorithm presented in* [22, 23] *computes a* C*-optimal extending function for* f*, and a number* $\mathcal{N}(f)$ *such that*

$$c\mathcal{N}(f) \leq \| f \|_{C^m(\mathbb{R}^n)} \leq C\mathcal{N}(f)\,.$$

Our algorithm uses one-time work at most $CN\log(N+1)$*, query work at most* $C\log(N+1)$*, and storage at most* CN*.*

The proofs of Theorems 3 and 4 are based on "finiteness principles". A "finiteness principle" for $f : E \longrightarrow \mathbb{R}$ asserts that

$$\| f \|_{C^m(\mathbb{R}^n)} \leq C \cdot \max\{\| (f|_S) \|_{C^m(\mathbb{R}^n)} \colon S \subseteq E\,, \#(S) \leq k^\#\}, \tag{6}$$

with $k^\#$ depending only on m and n.

For Whitney fields $\vec{P} \in \mathcal{Wh}(E)$, a finiteness principle asserts that

$$\| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq C \cdot \max\{\| (\vec{P}|_S) \|_{C^m(\mathbb{R}^n)} \colon S \subseteq E\,, \#(S) \leq k^\#\}\,. \tag{7}$$

In fact, (7), with $k^\# = 2$, is immediate from the classical Whitney extension theorem, which we now recall in the case of finite sets.

Theorem 5. (Whitney) *Let* $\vec{P} = (P^x)_{x \in E}$ *be a Whitney field, and let* $M \geq 0$ *be the smallest real number such that:*

$$|\partial^\alpha P^x(x)| \leq M \quad \textit{for } |\alpha| \leq m \textit{ and } x \in E\textit{; and}$$
$$|\partial^\alpha (P^x - P^y)(x)| \leq M\,|x-y|^{m-|\alpha|} \textit{ for } |\alpha| \leq m-1\,,\, x, y \in E\,.$$

Then

$$cM \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq CM\,.$$

See, e.g., [31, 38, 41].

Estimate (6) lies deeper. It was conjectured by Y. Brudnyi and P. Shvartsman [6],...,[10], and proven by them [8] in the case $m = 2$, with an optimal constant $k^\#$. (The case $m = 1$ is trivial.) The general case was proven in [14]. Related results on "Whitney's extension problem" and its variants appear in Whitney [41, 42, 43], Glaeser [25], Y. Brudnyi and P. Shvartsman [5],[6],...,[10], [35],...,[37]; E. Bierstone, P. Milman and W. Pawłucki [2, 3]; N. Zobin [44, 45] ; and C. Fefferman and B. Klartag [13],...,[20], [22, 23]. It would be interesting to find an easy proof of Theorem 4 or estimate (6).

Estimate (6) allows us to compute the order of magnitude of $\| f \|_{C^m(\mathbb{R}^n)}$, as in Theorem 4, because the order of magnitude of $\| (f|_S) \|_{C^m(\mathbb{R}^n)}$ for $\#(S) \leq k^\#$ may be easily computed by linear algebra. (See [14] for details.) Hence, in effect, we may take $\mathcal{N}(f)$ in Theorem 4 to be the right-hand side of (6).

As for Theorem 3, we can take $\mathcal{N}(\vec{P})$ to be the number M in Theorem 5 (which is comparable to the right-hand side of (7), with $k^\# = 2$). A glance at Theorem 5 suggests that it takes work N^2 to compute the number M. In fact, the work to compute the order of magnitude of M can be cut down to $N \log N$, by using the "Well-Separated Pairs Decomposition", and the "Balanced Box Decomposition Tree" from computer science. (See Callahan-Kosaraju [11] and Arya et al [1].) These ideas from computer science are clearly related to our problems, since, e.g., they allow an efficient computation of the Lipschitz norm of a function $f : E \longrightarrow \mathbb{R}$, up to a factor $(1+\epsilon)$. See Har-Peled and Mendel [26]. These results from computer science play a key rôle in our work, here and in [22, 23].

The proofs of Theorem 5, and of (6), (7), are constructive. Thus, once we know how to compute the order of magnitude of $\| f \|_{C^m(\mathbb{R}^n)}$ or $\| \vec{P} \|_{C^m(\mathbb{R}^n)}$, we can also compute C-optimal extending functions.

This concludes our discussion of Theorems 3 and 4.

To continue our preparation for the proofs of Theorems 1 and 2, we next discuss a result from our previous paper [21].

There, we gave a version of Whitney's theorem (Theorem 5), in which the C^m-norm of the extending function is controlled up to a factor $(1+\epsilon)$. To state the result, we introduce the notion of an "ϵ-testing set":

Let $\epsilon > 0$ be given. A finite set $S \subset \mathbb{R}^n$ is an "ϵ-testing set", provided there exists an open ball $B(x_0, r) \subset \mathbb{R}^n$, such that

(8) $S \subset B(x_0, r)$, and

(9) $|y - y'| > c\epsilon e^{-2/\epsilon} r$ for any two distinct points $y, y' \in S$.

An ϵ-testing set has simple geometry; roughly speaking, it has only one relevant lengthscale. Note that any ϵ-testing set S satisfies

(10) $\#(S) \leq \exp(C/\epsilon)$,

and that any singleton or pair, $S = \{x\}$ or $S = \{x, y\}$, is an ϵ-testing set.

The main result of [21] is as follows.

Theorem 6. *Let $\vec{P} = (P^x)_{x \in E}$ be a Whitney field. Then, for $0 < \epsilon < c$, we have*

(11) $$\| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \max\{\| (\vec{P}|_S) \|_{C^m(\mathbb{R}^n)} : S \subseteq E \text{ is an } \epsilon\text{-testing set}\}$$

In view of (10), Theorem 6 is a "$(1+\epsilon)$-finiteness principle", with $k^\# = k^\#(\epsilon) = \exp(C/\epsilon)$.

We comment on the proof of Theorem 6. Recall that Whitney's classical proof of Theorem 5 is based on a "Whitney partition of unity", adapted to a decomposition of $\mathbb{R}^n$ into "Whitney cubes". Our proof of Theorem 6 patches together $(1+\epsilon)$-optimal extending functions for $\vec{P}|_S$ (for suitable ϵ-testing sets S), by means of a "gentle partition of unity". (See Sections 5 and 6 below.)

The proof of Theorem 6 is constructive. We refine it here, bringing in computer-science ideas from [11], to prove the following result, analogous to the main result in [20].

Theorem 7. *Let $E \subset \mathbb{R}^n$ be given, with $\#(E) = N \geq 1$. Let $0 < \epsilon < c$. Then there exists a list $S_1, S_2, \ldots, S_L$ of ϵ-testing sets contained in E, with the following properties.*

(A) $L \leq \frac{C}{\epsilon} N$.

(B) $\| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \max\{\| (\vec{P}|_{S_\ell}) \|_{C^m(\mathbb{R}^n)} : \ell = 1, \ldots, L\}$ *for any* $\vec{P} \in \mathcal{Wh}(E)$.

(C) *The list $S_1, \ldots, S_L$ can be computed from ϵ and E, using work at most $\exp(C/\epsilon) N \log N$, and storage at most $\exp(C/\epsilon) N$.*

Moreover, given a $(1+\epsilon)$-optimal extending function F_ℓ for $\vec{P}|_{S_\ell}$, for each $\ell = 1, \ldots, L$, the proof of Theorem 7 constructs a $(1+C\epsilon)$-optimal extending function for $\vec{P}$. (We patch together the F_ℓ, using a gentle partition of unity.)

We are now ready to discuss the proof of Theorem 1.
Once we have established Theorem 7, our task in proving Theorem 1 reduces to the following extension problem.

Problem 1: *Given $0 < \epsilon < c$, and given a Whitney field $\vec{P}$ on an ϵ-testing set, compute a $(1+C\epsilon)$-optimal extending function for $\vec{P}$, and compute a number $\mathcal{N}_\epsilon(\vec{P})$, such that*

$$(12) \quad (1+C\epsilon)^{-1}\mathcal{N}_\epsilon(\vec{P}) \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)} \leq (1+C\epsilon)\mathcal{N}_\epsilon(\vec{P}).$$

This is unfortunately not trivial, even for a Whitney field on a single point. (Recall that the analogous computation in the setting of Theorems 3 and 4 required nothing more than trivial linear algebra.)

To explain our solution to Problem 1, we confine our discussion to the computation of a number $\mathcal{N}_\epsilon(\vec{P})$ satisfying (12). Once we can do this, we can also compute a $(1+C\epsilon)$-optimal extending function, since our methods are constructive.

To simplify further, we replace Problem 1 by the following easier version.

Problem 2: *Let $0 < \epsilon < c$, and let $S \subset B(x_0, r)$ be an ϵ-testing set, as in* (8), (9). *Assume $r < \epsilon^{-1}$. Given a Whitney field $\vec{P} \in Wh(S)$, compute a number $\mathcal{N}_\epsilon(\vec{P})$ such that*

$$(1+C\epsilon)^{-1} \| \vec{P} \|_{C^m(B(x_0,r))} \leq \mathcal{N}_\epsilon(\vec{P}) \leq (1+C\epsilon) \| \vec{P} \|_{C^m(B(x_0,2r))}.$$

Problem 2 is easier than Problem 1, since it is enough to construct a nearly optimal extending function on the ball $B(x_0, r)$, as opposed to the whole $\mathbb{R}^n$. We will see, in Sections 14...20 below, that the solution of Problem 2 is in fact one step in our solution of Problem 1. We omit details here, and confine our discussion to Problem 2. Our introductory discussion of Problem 2 is oversimplified; see Section 14 for correct statements.

A crucial step in our solution of Problem 2 is the following

Smoothing Lemma: *Let $S \subset B(x_0, r)$ be an ϵ-testing set, as in* (8), (9) *where $0 < \epsilon < c$. Then, for any function $F \in C^m(B(x_0, 2r))$, there exists a function $\tilde{F} \in C^{m+1}(B(x_0, r))$, with the following properties:*

(A) $J_y(\tilde{F}) = J_y(F)$ *for all* $y \in S$.

(B) $\| \tilde{F} \|_{C^m(B(x_0,r))} \leq (1+C\epsilon) \| F \|_{C^m(B(x_0,2r))}$.

(C) $|\partial^\alpha \tilde{F}(x)| \leq C \exp\left(\frac{4m}{\epsilon}\right) r^{-1} \| F \|_{C^m(B(x_0,2r))}$ *for* $|\alpha| = m+1, x \in B(x_0, r)$.

Thus, $\tilde{F}$ inherits the good properties of F, and its $(m+1)^{\underline{rst}}$ derivatives are under control.

Let us first discuss the proof of the Smoothing Lemma, and then see how it applies to Problem 2.

It is natural to try to prove the Smoothing Lemma by convolving F with an approximate identity. This produces a function $\tilde{F}_0$ that satisfies (B) and (C) but not (A). In fact, for $y \in S$, we expect that $J_y(\tilde{F}_0)$ will be nowhere near $J_y(F)$, since we have no control over the modulus of continuity of the $m^{\underline{th}}$ derivatives of F. Our problem is to "correct" $\tilde{F}_0$, to achieve (A), without spoiling (B) and (C).

To do so, we let $\tilde{P}^y = J_y(F) \in \mathcal{P}$, for each $y \in S$. The function $\tilde{P}^y$ satisfies (A), (B), (C) locally on a tiny ball B^y about y; but it makes no sense to use $\tilde{P}^y$ as an extending function outside B^y. We patch together our functions $\tilde{F}_0$ and $\tilde{P}^y$ (all $y \in S$), by using a gentle partition of unity, to produce a function $\tilde{F}$ that satisfies (A), (B) and (C).

This concludes our summary of the proof of the Smoothing Lemma.

We prepare to apply the Smoothing Lemma to Problem 2. To do so, we extend our testing set S to a "fine net", i.e., a finite set S^+ with the following properties.

(13) $S \subset S^+ \subset B(x_0, r)$.

(14) For any $x \in B(x_0, r)$, there exists $y \in S^+$ such that $|x-y| \leq \exp\left(-\frac{6m}{\epsilon}\right) r$.

(15) $\#(S^+) \leq \exp(C/\epsilon)$.

It is trivial to construct such an S^+.

To solve Problem 2 using the Smoothing Lemma, we now pose a

Linear Programming Problem: Find a real number M and polynomials $P^+_y \in \mathcal{P}$ for all $y \in S^+$, satisfying the following constraints, with M as small as possible.

The Constraints:

(16) $|P^+_y|_y \leq M$ for each $y \in S^+$.

(17) $|\partial^\alpha(P^+_y - P^+_{y'})(y)| \leq Cr^{-1}\exp\left(\frac{4m}{\epsilon}\right)|y-y'|^{m+1-|\alpha|}M$ for $|\alpha| \leq m$, $y, y' \in S^+$.

(18) $P^+_y = P^y$ for each $y \in S$, where $\vec{P} = (P^y)_{y\in E}$ is the given Whitney field in Problem 2.

Using the Oracle, we can replace (16) by a family of linear constraints, without significantly changing the problem. Thus, the above minimization problem is indeed a linear programming problem. This linear programming problem involves at most $\exp(C/\epsilon)$ linear constraints in a vector space of dimension at most $\exp(C/\epsilon)$.

Let $(M_0, (P^0_y)_{y\in S^+})$ be a solution of the above linear programming problem. Using the Smoothing Lemma, we will show that

$$(19)\quad (1+C\epsilon)^{-1}\ \|\ \vec{P}\ \|_{C^m(B(x_0,r))} \le M_0 \le (1+C\epsilon)\cdot\ \|\ \vec{P}\ \|_{C^m(B(x_0,2r))}.$$

Once we know (19), we can just take $\mathcal{N}_\epsilon(\vec{P}) = M_0$, and we will have solved Problem 2.

To see (19), we argue as follows. By definition of the C^m-norm of a Whitney field, there exists $F \in C^m(B(x_0, 2r))$, such that

$$(20)\quad J_y(F) = P^y \text{ for each } y \in S, \text{ and}$$

$$(21)\quad \|\ F\ \|_{C^m(B(x_0,2r))} \le (1+\epsilon)\cdot\ \|\ \vec{P}\ \|_{C^m(B(x_0,2r))}.$$

Applying the Smoothing Lemma, we obtain a function $\tilde{F} \in C^m(B(x_0, r))$, satisfying

$$(22)\quad J_y(\tilde{F}) = J_y(F) \text{ for each } y \in S,$$

$$(23)\quad \|\ \tilde{F}\ \|_{C^m(B(x_0,r))} \le (1+C\epsilon)\cdot\ \|\ \vec{P}\ \|_{C^m(B(x_0,2r))}, \text{ and}$$

$$(24)\quad |\partial^\alpha \tilde{F}(x)| \le C\exp\left(\tfrac{4m}{\epsilon}\right) r^{-1}\cdot\ \|\ \vec{P}\ \|_{C^m(B(x_0,2r))} \text{ for } |\alpha| = m+1,\ x \in B(x_0, r).$$

Let us set

$$(25)\quad M = (1+C\epsilon)\cdot\ \|\ \vec{P}\ \|_{C^m(B(x_0,2r))} \text{ and}$$

$$(26)\quad P^+_y = J_y(\tilde{F}) \text{ for all } y \in S^+.$$

Then the constraints (16), (17), (18) hold. In fact, (16) follows from (23), (25) and (26), by definition (3) of the C^m-norm.

Moreover, (17) follows from (24), (25) and (26) and Taylor's theorem; while (18) is immediate from (20), (22) and (26).

Thus, the above $(M, (P^+_y)_{y\in S^+})$ satisfies the constraints (16), (17), (18), whereas $(M^0, (P^0_y)_{y\in S^+})$ is a minimizer. Consequently, $M_0 \le M$. That is,

$$M_0 \le (1+C\epsilon)\cdot\ \|\ \vec{P}\ \|_{C^m(B(x_0,2r))},$$

which is half of the desired estimate (19).

To establish the other half of (19), we recall that $(M_0, (P^0_y)_{y\in S^+})$ satisfies the constraints (16), (17), (18). Thus,

$$(27)\quad |P^0_y|_y \le M_0 \text{ for } y \in S^+;$$

$$(28)\quad |\partial^\alpha(P^0_y - P^0_{y'})(y)| \le Cr^{-1}\exp\left(\tfrac{4m}{\epsilon}\right)|y-y'|^{m+1-|\alpha|}M_0 \text{ for } |\alpha| \le m,\ y, y' \in S^+;$$

and

(29) $P^0_y = P^y$ for each $y \in S$.

From (27) we obtain the estimates

(30) $|\partial^\alpha P^0_y(y)| \leq CM_0$ for $|\alpha| \leq m$, $y \in S^+$.

(This step uses the "Bounded Distortion Property", which is one of the assumptions made in Section 3 on the family of norms $|\cdot|_x$ in (3).)

In the statement of Problem 2, we assume that $r < \epsilon^{-1}$ and $\epsilon < c$. Hence, (30) trivially implies

(31) $|\partial^\alpha P^0_y(y)| \leq Cr^{-1} \exp\left(\frac{4m}{\epsilon}\right) M_0$ for $|\alpha| \leq m$, $y \in S^+$.

Thanks to (28), (31) and the classical Whitney extension theorem (Theorem 5 with $m+1$ in place of m), there exists a function $F \in C^{m+1}(\mathbb{R}^n)$, such that

(32) $J_y(F) = P^0_y$ for all $y \in S^+$,

and

(33) $|\partial^\alpha F(x)| \leq Cr^{-1} \exp\left(\frac{4m}{\epsilon}\right) M_0$ for $|\alpha| = m+1$ and $x \in \mathbb{R}^n$.

In view of (29) and (32), we have

(34) $J_y(F) = P^y$ for $y \in S$.

We will show that

(35) $\| F \|_{C^m(B(x_0,r))} \leq (1 + C\epsilon)M_0$.

Once we establish (35), we obtain from (34), (35) and the definition of the C^m-norm of a Whitney field that

$$\| \vec{P} \|_{C^m(B(x_0,r))} \leq (1 + C\epsilon)M_0,$$

which is the remaining half of our desired estimate (19).

Thus, to prove (19) and thereby solve Problem 2, it remains only to prove (35). By definition (3), estimate (35) is equivalent to the estimate

(36) $|J_x(F)|_x \leq (1 + C\epsilon)M_0$ for all $x \in B(x_0, r)$.

So our task is to prove (36). To do so, we start with the points of the fine net S^+. From (27) and (32), we see at once that

(37) $|J_y(F)|_y \leq M_0$ for all $y \in S^+$.

Moreover, whenever $|x - y| \leq \exp\left(-\frac{6m}{\epsilon}\right) r$ and $|\alpha| \leq m$, estimate (33) and Taylor's theorem give

(38) $$\begin{aligned} |\partial^\alpha [J_x(F) - J_y(F)](x)| &\leq Cr^{-1} \exp\left(\tfrac{4m}{\epsilon}\right) |x - y|^{m+1-|\alpha|} M_0 \\ &\leq Cr^{-1} \exp\left(\tfrac{4m}{\epsilon}\right) [\exp\left(-\tfrac{6m}{\epsilon}\right) r]^{m+1-|\alpha|} M_0 \leq \epsilon M_0, \end{aligned}$$

since $r < \epsilon^{-1}$ in Problem 2.

Thus, the jets $J_x(F)$ and $J_y(F)$ are "close". From (38) and the properties of the norms $|\cdot|_x$ assumed in Section 3, we conclude that

$$(39)\quad |J_x(F)|_x \le (1+C\epsilon)|J_y(F)|_y + C\epsilon M_0, \text{ whenever } |x-y| \le \exp\left(-\tfrac{6m}{\epsilon}\right) r.$$

The desired estimate (36) now follows at once from (37) and (39), together with property (14) of the fine net S^+.

This completes our summary of the solution of Problem 2, as well as our discussion of the proof of Theorem 1. We pass to the proof of Theorem 2.

A fundamental difference between Theorems 1 and 2 is that the natural "finiteness principle" analogous to Theorem 6 is false in the context of Theorem 2. In fact, the following negative result is proven in Fefferman-Klartag [24], for the case of $C^2(\mathbb{R}^2)$, equipped with the norm (1) or (2).

Theorem 8. *There exists a universal constant $\epsilon_0 > 0$, for which the following holds.*

Given any $k^\#$, there exists a function $f : E \longrightarrow \mathbb{R}$ on a finite set $E \subset \mathbb{R}^2$, such that

$$\max\left\{ \| (f|_S) \|_{C^2(\mathbb{R}^2)} : S \subseteq E, \#(S) \le k^\# \right\} \le 1, \textit{ but } \| f \|_{C^2(\mathbb{R}^2)} \ge 1 + \epsilon_0.$$

Thus, an efficient algorithm to compute $\| f \|_{C^m(\mathbb{R}^n)}$ up to a factor $(1+\epsilon)$ will require new ideas. Our Theorem 2 provides a (presumably) inefficient solution to this problem.

To explain the ideas in the proof of Theorem 2, we again confine ourselves to the computation of the norm, leaving the construction of a nearly optional extending function to Sections 26...29 below.

Let us say that a Whitney field $\vec{P} = (P^x)_{x\in E}$ "agrees" with a function $f : E \longrightarrow \mathbb{R}$, provided $(P^x)(x) = f(x)$ for each $x \in E$. Immediately from the relevant definitions, we then have

$$\| f \|_{C^m(\mathbb{R}^n)} = \inf\left\{ \| \vec{P} \|_{C^m(\mathbb{R}^n)} : \vec{P} \in Wh(E),\ \vec{P} \text{ agrees with } f \right\}.$$

Together with Theorem 1, this yields

$$(40)\quad (1 + C\epsilon)^{-1} \| f \|_{C^m(\mathbb{R}^n)} \le \inf\left\{ \mathcal{N}_\epsilon(\vec{P}) : \vec{P} \in Wh(E),\ \vec{P} \text{ agrees with } f \right\} \le (1 + C\epsilon) \| f \|_{C^m(\mathbb{R}^n)},$$

with $\mathcal{N}_\epsilon(\vec{P})$ as in Theorem 1.

To exploit (40), we review the proof of Theorem 1 to see how $\mathcal{N}_\epsilon(\vec{P})$ depends on $\vec{P} \in Wh(E)$. We find that there exist a finite-dimensional vector space W, and (real-valued) linear functionals $\lambda_1, \dots, \lambda_K$ on $Wh(E) \oplus W$, such that

$$(41)\quad \mathcal{N}_\epsilon(\vec{P}) = \min_{w\in W} \max_{k=1,\dots,K} |\lambda_k(\vec{P}, w)|.$$

Moreover, W and $\lambda_1, \dots, \lambda_K$ may be computed efficiently from ϵ and E.

Combining (40) and (41), we now see that

$$(42)\quad (1+C\epsilon)^{-1}\,\|f\|_{C^m(\mathbb{R}^n)} \le$$
$$\le \min\Big\{\max_{k=1,\dots,K} |\lambda_k(\vec{P},w)| : (\vec{P},w)\in \mathcal{Wh}(E)\oplus W,\ \vec{P}\text{ agrees with } f\Big\}$$
$$\le (1+C\epsilon)\,\|f\|_{C^m(\mathbb{R}^n)}\ .$$

Consequently, we may take $\mathcal{N}_\epsilon(f)$ to be the "minimax" in (42), and we have

$$(1+C\epsilon)^{-1}\,\mathcal{N}_\epsilon(f) \le \|f\|_{C^m(\mathbb{R}^n)} \le (1+C\epsilon)\,\mathcal{N}_\epsilon(f)\,,$$

as in Theorem 2. The computation of $\mathcal{N}_\epsilon(f)$ is a linear programming problem, involving at most $\exp(C/\epsilon)\cdot N$ constraints, in a vector space of dimension at most $\exp(C/\epsilon)\cdot N$.

We solve this linear programming problem using the "ellipsoid method" from computer science. (See [29].) The one-time work and storage given in Theorem 2 arise from the ellipsoid method. More efficient linear programming algorithms will surely sharpen Theorem 2, but are unlikely to yield an efficient computation of $\|f\|_{C^m(\mathbb{R}^n)}$ up to a factor $(1+\epsilon)$ without further ideas.

Note that the computation for Theorem 1 entails solving $\sim N/\epsilon$ "little" linear programming problems, each having "size" $\sim \exp(C/\epsilon)$. On the other hand, our computation for Theorem 2 involves a single "big" linear programming problem, of "size" $\sim \exp(C/\epsilon)\cdot N$. Of course, the "big problem" requires much more work than all the "little problems" combined.

This completes our introductory remarks on the proof of Theorem 2. Again, we stress that our discussion of the proofs of Theorems 1 and 2 is oversimplified. (For instance, the proof of Theorem 7 actually comes after that of Theorem 1.) We refer the reader to Sections 1...29 below, for the real story.

Let us briefly explain what happens in each of the sections to follow.

Section 1 gives notation and elementary definitions.

Section 2 specifies our model of computation in more detail than in this introduction.

Section 3 gives our assumptions on the norms $|\cdot|_x$ in (3), defines "controlled constants", and shows how to replace (16) by a family of linear constraints, using the Oracle.

Section 4 recalls well-known results from computer science.

Section 5 recalls the basic lemma from [21] on "gentle partitions of unity", and gives a few simple corollaries.

Section 6 states the "Main Patching Lemma", which is a constructive version of Theorem 6. Given a list of useful inputs, the Main Patching Lemma constructs a $(1 + C\epsilon)$-optimal extending function for a given Whitney field $\vec{P}$.

The task of computing the "useful inputs" for the Main Patching Lemma occupies Sections 8...20 below.

Section 7 proves the Main Patching Lemma, along the lines of [21].

Section 8 uses the Well-Separated-Pairs Decomposition to compute efficiently the order of magnitude of the number M in Theorem 5.

Section 9 computes the "regularized distance", a smooth function comparable to the distance to a given finite set $E \subset \mathbb{R}^n$. See [31, 38, 41] for this notion.

Section 10 gives routine algorithms to compute some cutoff functions appearing in the Main Patching Lemma.

Section 11 computes the list of testing sets used by the Main Patching Lemma.

Section 12 states and proves the Smoothing Lemma, which we have stated (not quite correctly) in this introduction.

Section 13 implements our discussion of (27)...(39) above, as an algorithm.

Section 14 (roughly speaking) applies the Smoothing Lemma to solve Problem 2 (local extension from a Whitney field on a testing set), as explained earlier in this introduction.

Sections 15...20 pass from our solution of Problem 2 (local extension) to a solution of Problem 1 (global extension). Section 15 treats the case of a Whitney field on a single point, while Sections 16...20 combine our results from Sections 14 and 15 to treat the general case. More precisely, we subdivide the general case into several subcases, depending essentially on the diameter of the given ϵ-testing set. Sections 16...19 treat the various subcases, and Section 20 puts them together.

Sections 21...24 present the proof of Theorem 1. The algorithm promised in Theorem 1 uses the results of Sections 8 through 20, in order to apply the Main Patching Lemma.

Section 25 proves Theorem 7, and also records some observations on the algorithm in Theorem 1. These observations will be used in the proof of Theorem 2.

Sections 26...29 give the proof of Theorem 2. In Section 26, we introduce "minimax functions", similar to the minimax in (41) above. We

prove in Sections 27 and 28 that the quantity $\mathcal{N}_\epsilon(\vec{P})$ in Theorem 1 is a minimax function of $\vec{P}$, thus establishing (41). Finally, Section 29 exploits (41), as explained in our introduction, to compute an almost-optimal Whitney field $\vec{P} \in \mathsf{Wh}(E)$ agreeing with a given function $f : E \longrightarrow \mathbb{R}$. This reduces Theorem 2 to Theorem 1.

In the sections below, we present algorithms in the following format.

Algorithm [Name]: *Given [data] satisfying [assumptions], we compute [stuff] such that [some properties hold].*

The algorithm requires work at most [W] and storage at most [S].

Explanation: [Here, we specify the algorithm and prove that it performs as asserted.]

If the above [assumptions] are not satisfied, then we make no claim as to what our algorithm does, or even whether it terminates. Our algorithm need not check whether the [assumptions] are satisfied.

Many of our algorithms compute a function $F \in C^m(\mathbb{R}^n)$, and possibly other data as well (e.g., a number $\mathcal{N}_\epsilon(\vec{P})$ or a ball $B(x_0, r)$). In this case, instead of saying that our algorithm "requires work at most [W] and storage at most [S]", we say the following:

The algorithm requires one-time work at most $[L_0]$*, query work at most* $[L_1]$*, and storage at most* $[S]$.

Recall that any work used to compute answers other than the function F is counted as part of the one-time work.

For a few algorithms, which are either well-known or discussed in [23], we provide a reference to the literature in place of an Explanation.

Acknowledgments. It is a pleasure to thank Bo'az Klartag for many valuable conversations, some of which led to major improvements in this paper. I am grateful also to A. Naor, A. Razborov, and A. Wigderson, for pointing me to the relevant ideas and literature from computer science. I thank A. Tsao, for bringing to my attention the practical problem of passing a smooth surface through N given points. From E. Bierstone, Y. Brudnyi, B. Klartag, P. Milman, W. Pawłucki, P. Shvartsman, and N. Zobin I learned a lot about "Whitney's extension problems".

As always, Gerree Pecht spoils me with her rapid, accurate LaTeXing of my long manuscript.

We conclude this introduction with a trivial remark about our labeling of equations. Let k, k', ℓ be integers, and suppose we are in Section k'. Then we write $(k.\ell)$ to denote equation (ℓ) in Section k. In the case $k' = k$, we simply write (ℓ).

Let us get to work.

1. Notation and Preliminaries

Recall that we have fixed $m, n \geq 1$, and that $\mathcal{P}$ denotes the vector space of all (real) polynomials of degree at most m on $\mathbb{R}^n$.

For $\Omega \subset \mathbb{R}^n$ open, we write $C^m_{loc}(\Omega)$ for the space of real-valued locally C^m functions on Ω, and we write $C^m(\Omega)$ for the space of all $F \in C^m_{loc}(\Omega)$ such that F and its derivatives through $m^{\underline{th}}$ order are bounded on Ω. We define $C^{m+1}_{loc}(\Omega)$ and $C^{m+1}(\Omega)$ similarly. For $F \in C^m_{loc}(\Omega)$ and $x \in \Omega$, we write $J_x(F)$ for the $m^{\underline{th}}$ order Taylor polynomial of F at x. Thus, $J_x(F)$ belongs to $\mathcal{P}$.

For $P_1, P_2 \in \mathcal{P}$ and $x \in \mathbb{R}^n$, we write $P_1 \odot_x P_2$ to denote the product of P_1 and P_2 as m-jets at x; i.e., $P_1 \odot_x P_2$ is the unique polynomial in $\mathcal{P}$ that satisfies

$$\partial^\alpha([P_1 \odot_x P_2] - P_1P_2)(x) = 0 \text{ for } |\alpha| \leq m.$$

If $F_1, F_2 \in C^m_{loc}(\Omega)$ and $x \in \Omega$, then $J_x(F_1F_2) = J_x(F_1) \odot_x J_x(F_2)$.

For $F \in C^m_{loc}(\Omega)$, we write $\mathrm{supp}_\Omega F$ to denote the set of all points $x \in \Omega$ such that F is not identically zero on any neighborhood of x. When $\Omega = \mathbb{R}^n$, we write $\mathrm{supp}\, F$ for $\mathrm{supp}_\Omega F$.

Recall that a Whitney field on a finite set $S \subset \mathbb{R}^n$ is a family of polynomials

$$\vec{P} = (P^y)_{y \in S} \tag{1}$$

indexed by S, with each $P^y \in \mathcal{P}$. If $S \subset \Omega$ (with Ω open), and if $F \in C^m_{loc}(\Omega)$, then we say that F *agrees with* $\vec{P}$ as in (1) if we have $J_y(F) = P^y$ for all $y \in S$.

If $\vec{P}$ as in (1) is a Whitney field on S, and if $S' \subset S$ is a subset, then we write $\vec{P}|_{S'}$ for the Whitney field $(P^y)_{y \in S'}$, and we call $\vec{P}|_{S'}$ the "restriction" of $\vec{P}$ to S'.

If F agrees with $\vec{P}|_{S'}$, then we say that F "*agrees with* $\vec{P}$ *on* S'". If $S' = \{y_0\}$ is a singleton, then we say that F "agrees with $\vec{P}$ at y_0".

We write $\mathrm{Wh}(S)$ for the vector space of Whitney fields on a given set $S \subset \mathbb{R}^n$.

Let $x \in \mathbb{R}^n$ and $r > 0$. Then $B(x, r)$ denotes the open ball in $\mathbb{R}^n$ with center x and radius r.

A dyadic cube in $\mathbb{R}^n$ is a Cartesian product of the form

$$Q = [2^\ell m_1, 2^\ell(m_1 + 1)) \times \cdots \times [2^\ell m_n, 2^\ell(m_n + 1))$$

where $\ell, m_1, \ldots, m_n$ are integers. More generally, a "cube" in $\mathbb{R}^n$ is a Cartesian product of half-open intervals

$$Q = [a_1, b_1) \times [a_2, b_2) \times \cdots \times [a_n, b_n) \subset \mathbb{R}^n,$$

with $b_1 - a_1 = b_2 - a_2 = \cdots = b_n - a_n$. We write δ_Q to denote the sidelength of a cube Q, and we write $\mathrm{cent}(Q)$ to denote the center of Q.

We define Q^* to be the unique cube satisfying $\mathsf{cent}(Q^*) = \mathsf{cent}(Q)$ and $\delta_{Q^*} = 3\delta_Q$. Also, we write Q^{**} to denote $(Q^*)^*$, and Q^{***} to denote $(Q^{**})^*$. Note that if Q is a dyadic cube, then Q^*, Q^{**} and Q^{***} may be trivially partitioned into dyadic cubes .

For any set $E \subset \mathbb{R}^n$, we write E^{int} and E^{cl} for the interior and closure of E, respectively.

For any set S, we write $\#(S)$ to denote the number of elements of S. If S is infinite, then we define $\#(S) = +\infty$. We write ϕ to denote the empty set.

For $E \subset \mathbb{R}^n$, we write $\mathsf{diam}(E)$ to denote the diameter of E, i.e., $\sup\{|x - x'| : x, x' \in E\}$. Also, for $E, E' \subset \mathbb{R}^n$, we write $\mathsf{dist}(E, E')$ to denote $\inf\{|x-x'| : x \in E,\ x' \in E'\}$. If $E = \phi$, then $\mathsf{diam}(E) = 0$, and if $E = \phi$ or $E' = \phi$, then $\mathsf{dist}(E, E') = +\infty$. We write $\mathsf{dist}(x, E')$ to denote $\inf\{|x - x'| : x' \in E'\}$. If $E' = \phi$, then $\mathsf{dist}(x, E') = +\infty$.

2. The Model of Computation

Our model of computation (called "real RAM" in the computer science literature) differs from a standard Von Neumann computer [39] in that we assume that the computer can store, retrieve from memory, and manipulate exact real numbers , without roundoff errors. Our computer includes several registers, and finitely many memory cells labeled by integers 1,2,..., Memory-Size. Each memory cell and each register is capable of storing either an integer or a real number. The machine can perform the following operations, each of which costs one unit of "work".

- Retrieve the contents of memory cell k, and write it to one of the registers. ($1 \le k \le$ Memory-Size).
- Write the contents of a register to memory cell k ($1 \le k \le$ Memory-Size).
- Write the number 0 or 1 to one of the registers.
- Given two real numbers x and y appearing in the registers, compute $x + y$, $x - y$, xy, and (if $y \neq 0$) x/y. Also, decide whether $x < y$, $x > y$, or $x = y$.
- Given a real number x appearing in a register, compute $\mathsf{exp}(x)$; and, if $x > 0$, compute $\log(x)$.
- Given a real number x appearing in a register, compute $\lfloor x \rfloor$, the greatest integer $\le x$.
- Retrieve a real number entered by the user, and place it in a register.
- Output to the user a real number or integer appearing as the contents of a register.

We assume that the above operations can be carried out perfectly, without roundoff error.

Also, as mentioned in the introduction, we assume that our computer is capable of communicating with an Oracle, which will be used to convey to the computer (an approximation to) $|P|_x$ for a given $P \in \mathcal{P}$ and $x \in \mathbb{R}^n$. Thus, we assume that the computer can do the following:

- Output the contents of a register to the Oracle.
- Wait for the Oracle to communicate a real number S; once the Oracle speaks, the number S is placed in one of the registers of our machine.

To output a number to the Oracle costs us one unit of "work". However, when the Oracle speaks to our machine, then she decides how many units of "work" to charge us. For more about the Oracle, see Section 3.

The flow of control in our computer proceeds as for a standard Von Neumann machine [39]. This concludes our brief description of the model of computation used in this paper.

It is well-known to computer scientists that the above "real RAM" model of computation (even without an Oracle) leads to some suspiciously efficient algorithms that make crucial use of the complete absence of roundoff errors. See, e.g., [34]. This issue can be dealt with in various ways. In Fefferman-Klartag [23], we made a rigorous analysis of the effect of roundoff errors on the algorithms presented there. We believe that the algorithms in this paper can be analyzed in a somewhat similar spirit, but we have not put in the hard work required to decide the issue. The best we can say now is that it is natural to guess that our algorithms will survive in the presence of small enough roundoff errors.

3. C^m Norm

Suppose we are given a norm $|\cdot|_x$ on the vector space $\mathcal{P}$, for each $x \in \mathbb{R}^n$. We assume that these norms satisfy the following conditions, for certain given constants $\bar{c}_0, \bar{C}_0, \bar{C}_1$.

Bounded Distortion Property: We have

$$\bar{c}_0 \cdot \max_{|\alpha| \le m} |\partial^\alpha P(x)| \le |P|_x \le \bar{C}_0 \cdot \max_{|\alpha| \le m} |\partial^\alpha P(x)|$$

for all $x \in \mathbb{R}^n$ and $P \in \mathcal{P}$.

Approximate Translation-Invariance Property: Let $P \in \mathcal{P}$ and $x, y \in \mathbb{R}^n$ be given. Define a polynomial $P_y \in \mathcal{P}$ by setting $P_y(z) = P(z-y)$ for all $z \in \mathbb{R}^n$. Then we have

$$|P_y|_{x+y} \le \exp(\bar{C}_1|y|) \cdot |P|_x .$$

Throughout this paper, we assume that we are given the family of norms $|\cdot|_x$, and the constants $\bar{c}_0, \bar{C}_0, \bar{C}_1$, such that the above properties hold. We say that a constant is *controlled* if it is determined by $m, n, \bar{c}_0, \bar{C}_0$ and $\bar{C}_1$.

We write c, C, C', etc., to denote controlled constants. Unless otherwise specified, these constants may change from one occurrence to the next. We recall the following definitions from the introduction.

For $F \in C^m(\Omega)$, we define $\| F \|_{C^m(\Omega)} = \sup_{x\in\Omega} |J_x(F)|_x$.
Also, for any Whitney field $\vec{P}$, we define $\| \vec{P} \|_{C^m(\mathbb{R}^n)}$ to denote the infimum of $\| F \|_{C^m(\mathbb{R}^n)}$ over all $F \in C^m(\mathbb{R}^n)$ that agree with $\vec{P}$.

We need to specify the norms $|\cdot|_x$ to our computer. Therefore, we assume we have access to an Oracle, as follows. We query the Oracle by specifying a point $x \in \mathbb{R}^n$, a polynomial $P \in \mathcal{P}$, and a number ϵ, with $0 < \epsilon < 1$. The Oracle responds by producing a number $\mathcal{N}_\epsilon(P, x)$, such that

$$(1 + \epsilon)^{-1} \mathcal{N}_\epsilon(P, x) \le |P|_x \le (1 + \epsilon)\, \mathcal{N}_\epsilon(P, x).$$

We are charged $\exp(C/\epsilon)$ units of work for each query (P, x, ϵ) addressed to the Oracle.

We may suppose that $\mathcal{N}_\epsilon(P, x) = \mathcal{N}_\epsilon(-P, x)$, for any P, x, ϵ. (In fact, we may replace $\mathcal{N}_\epsilon(P, x)$ by $\max\{\mathcal{N}_\epsilon(P, x), \mathcal{N}_\epsilon(-P, x)\}$ without harm.)

The following algorithm gives us a close approximation to the unit ball for the norm $|\cdot|_x$.

Algorithm 3.1. ("Find-Unit-Ball").

Given a number ϵ, with $0 < \epsilon < 1$; and given a point $x \in \mathbb{R}^n$; we compute a finite set $O(\epsilon, x)$ of linear functionals on $\mathcal{P}$, with the following properties.

(O_0) *For any $\lambda \in O(\epsilon, x)$, we have also $-\lambda \in O(\epsilon, x)$.*

(O_1) $(1+\epsilon)^{-1}|P|_x \le \max\{\lambda(P) : \lambda \in O(\epsilon, x)\} \le (1+\epsilon)|P|_x$ *for all* $P \in \mathcal{P}$.

(O_2) $\#(O(\epsilon, x)) \le \exp(C/\epsilon)$.

Moreover,

(O_3) *The work and storage used for the computation are at most* $\exp(C/\epsilon)$.

Explanation: We introduce some definitions and prove a few elementary facts, then we give the algorithm and prove (O_0)...(O_3). Fix $x \in \mathbb{R}^n$. For $P \in \mathcal{P}$, let $|P| = \max_{|\alpha|\le m} |\partial^\alpha P(x)|$. Thus,

(1) $c|P| \le |P|_x \le C|P|$ for $P \in \mathcal{P}$, by the Bounded Distortion Property.

Let $\mathcal{P}^*$ be the dual vector space to $\mathcal{P}$, and let $|\cdot|^*$ and $|\cdot|^*_x$ be the norms dual to $|\cdot|, |\cdot|_x$, respectively. Thus (1) implies

(2) $c|\lambda|^* \le |\lambda|^*_x \le C|\lambda|^*$ for $\lambda \in \mathcal{P}^*$.

Let A, $\epsilon > 0$ be real numbers. Assume that

(3) A exceeds a large enough controlled constant, and

(4) $0 < \epsilon < A^{-2}$.

Later, we will pick A to be a controlled constant, large enough to satisfy (3). For now, however, we just assume (3) and (4).

We introduce two finite sets $\Gamma \subset \mathcal{P}$ and $\Lambda^* \subset \mathcal{P}^*$, with the following properties.

(5) Every $P_0 \in \Gamma$ satisfies $|P_0| \leq 2A$.

(6) For any $P \in \mathcal{P}$ with $|P| \leq A$, there exists $P_0 \in \Gamma$ such that $|P - P_0| \leq \epsilon$.

(7) Every $\lambda_0 \in \Lambda^*$ satisfies $|\lambda_0|^* \leq 2A$.

(8) For any $\lambda \in \mathcal{P}^*$ with $|\lambda|^* \leq A$, there exists $\lambda_0 \in \Lambda^*$ such that $|\lambda - \lambda_0|^* \leq \epsilon$.

(We can easily construct such Λ^* and Γ. For instance, we can take Γ to be the ball of radius $2A$ about 0 for the norm $|\cdot|$, intersected with a fine enough cubic lattice in $\mathcal{P}$. We can define Λ^* similarly.)
Now, we define sets $\Gamma(\epsilon) \subseteq \Gamma$ and $O^*(\epsilon, x) \subseteq \Lambda^*$, as follows:

(9) $\Gamma(\epsilon) = \{P_0 \in \Gamma : \mathcal{N}_\epsilon(P_0, x) \leq 1 + A\epsilon\}$, where $\mathcal{N}_\epsilon(P_0, x)$ is the number computed by the Oracle for the query P_0, x, ϵ; and

(10) $O^*(\epsilon, x) = \{\lambda_0 \in \Lambda^* : \lambda_0(P_0) \leq 1 + 2A\epsilon$ for all $P_0 \in \Gamma(\epsilon)\}$.

We will prove two elementary propositions regarding $O^*(\epsilon, x)$.

Proposition 1. *Let $\lambda_0 \in O^*(\epsilon, x)$, and let $P \in \mathcal{P}$. Then*

$$\lambda_0(P) \leq (1 + 10A\epsilon) \cdot |P|_x.$$

Proof. We may assume that $|P|_x = 1$. Then $|P| \leq C$ by (1). Hence, (3) and (6) show that there exists $P_0 \in \Gamma$ such that $|P - P_0| \leq \epsilon$. Fix such a P_0. By (1), we have $|P - P_0|_x \leq C\epsilon$. Hence, $|P_0|_x \leq |P|_x + |P_0 - P|_x \leq 1 + C\epsilon$. The defining property of $\mathcal{N}_\epsilon(P, x)$ then yields $\mathcal{N}_\epsilon(P_0, x) \leq (1 + C'\epsilon) \leq 1 + A\epsilon$ by (3).

Thus, $P_0 \in \Gamma$ and $\mathcal{N}_\epsilon(P_0, x) \leq 1 + A\epsilon$. By definition (9), we have $P_0 \in \Gamma(\epsilon)$. Since also $\lambda_0 \in O^*(\epsilon, x)$, it follows from (10) that

(11) $\lambda_0(P_0) \leq 1 + 2A\epsilon$.

Moreover, we have $\lambda_0 \in O^*(\epsilon, x) \subseteq \Lambda^*$, hence $|\lambda_0|^* \leq 2A$ by (7). Consequently,

(12) $|\lambda_0(P - P_0)| \leq |\lambda_0|^* \cdot |P - P_0| \leq 2A\epsilon$.

From (11) and (12), we obtain $\lambda_0(P) \leq 1 + 4A\epsilon$, proving Proposition 1. ■

Proposition 2. *Let $P \in \mathcal{P}$. Then there exists $\lambda_0 \in O^*(\epsilon, x)$ such that $\lambda_0(P) \geq (1 - C\epsilon) \cdot |P|_x$.*

Proof. There exists $\lambda \in \mathcal{P}^*$ such that

(13) $|\lambda|^*_x = 1$ and

(14) $\lambda(P) = |P|_x$.

By (13) and (2), we have $|\lambda|^* \leq C$. Hence, by (3) and (8), there exists $\lambda_0 \in \Lambda^*$ such that

(15) $|\lambda - \lambda_0|^* \leq \epsilon$.

By (2), we therefore have

(16) $|\lambda - \lambda_0|^*_x \leq C\epsilon$.

From (13) and (16), we obtain

(17) $|\lambda_0|^*_x \leq 1 + C\epsilon$.

Let $P_0 \in \Gamma(\epsilon)$. Then $\mathcal{N}_\epsilon(P_0, x) \leq 1 + A\epsilon$ by (9); hence, $|P_0|_x \leq (1 + C\epsilon) \cdot (1 + A\epsilon)$ by defining property of $\mathcal{N}_\epsilon(P, x)$. Consequently, we have

$$\lambda_0(P_0) \leq |\lambda_0|^*_x \cdot |P_0|_x \leq (1 + C\epsilon) \cdot [(1 + C\epsilon) \cdot (1 + A\epsilon)] \quad \text{(see (17))}.$$

Thanks to (3) and (4), it follows that $\lambda(P_0) \leq 1 + 2A\epsilon$.

Thus, $\lambda_0 \in \Lambda^*$, and $\lambda_0(P_0) \leq 1 + 2A\epsilon$ for all $P_0 \in \Gamma(\epsilon)$. By definition (10), we have

(18) $\lambda_0 \in O^*(\epsilon, x)$.

Moreover, (14) and (16) yield the inequality

(19) $\lambda_0(P) = \lambda(P) - (\lambda - \lambda_0)(P) \geq |P|_x - |\lambda - \lambda_0|^*_x \cdot |P|_x \geq (1 - C\epsilon) \cdot |P|_x$.

The conclusion of Proposition 2 is immediate from (18) and (19). ■

Let us now take A to be a controlled constant, large enough that (3) holds. We have also (4), provided $\epsilon > 0$ is less than a small enough controlled constant.

Since A is now a controlled constant, the two Propositions above tell us that

(20) $(1 - C\epsilon) \cdot |P|_x \leq \max\{\lambda(P) : \lambda \in O^*(\epsilon, x)\} \leq (1 + C\epsilon) \cdot |P|_x$ for $P \in \mathcal{P}$.

Estimate (20) holds if ϵ is less than a small enough controlled constant.

Now we can explain how to carry out Algorithm 3.1. First suppose that ϵ is less than a small enough controlled constant. With A as above, we compute sets Λ^* and Γ satisfying (5)...(8). This elementary computation takes work and storage at most $C\epsilon^{-D}$, where $D = \dim \mathcal{P}$. Moreover, we can take our Λ^* and Γ to satisfy also

$$\#(\Lambda^*), \#(\Gamma) \leq C\epsilon^{-D}. \tag{21}$$

(Details of the computation of Γ and Λ^* are left to the reader.) We then compute the set $\Gamma(\epsilon)$ from (9). This requires $\#(\Gamma)$ queries to the Oracle, to learn the numbers $\mathcal{N}_\epsilon(P_0, x)$ for all $P_0 \in \Gamma$. Thus, the work to compute $\Gamma(\epsilon)$ is at most $\exp(C/\epsilon)$, thanks to (21) and our assumption on the work to query the Oracle.

Having computed $\Gamma(\epsilon)$, we can then compute $O^*(\epsilon, x)$ from (10). The work needed for this step is at most $C \cdot \#(\Lambda^*) \cdot \#(\Gamma) \leq C \cdot \epsilon^{-2D}$, while the storage needed is at most $C \cdot \epsilon^{-D}$.

Thus, we have computed $O^*(\epsilon, x)$ using total work and storage at most $\exp(C/\epsilon)$. The set $O^*(\epsilon, x)$ satisfies (20), and also

$$\#(O^*(\epsilon, x)) \leq \#(\Lambda^*) \leq C\epsilon^{-D} < \exp(C/\epsilon). \tag{22}$$

We have achieved (20) and (22), using work and storage at most $\exp(C/\epsilon)$, provided ϵ is less than a small enough controlled constant.

We now drop our assumption that ϵ is less than a small enough controlled constant. We assume merely that $0 < \epsilon < 1$.

To compute $O(\epsilon, x)$ satisfying $(O_1), (O_2), (O_3)$ in Algorithm 3.1, it is enough to replace ϵ by $\epsilon' = c\epsilon$ for a small enough controlled constant c; we then compute $O^*(\epsilon', x)$, and we just set $O(\epsilon, x) = O^*(\epsilon', x)$. Properties $(O_1), (O_2), (O_3)$ are immediate from the corresponding properties of $O^*(\epsilon', x)$.

By taking the sets Λ^* and Γ in (5)...(8) to be symmetric about the origin, we obtain from the above construction that $\lambda \in O(\epsilon, x)$ if and only if $-\lambda \in O(\epsilon, x)$, for any given functional λ on $\mathcal{P}$. Thus, property (O_0) holds as well. This concludes our explanation of Algorithm 3.1.

We close this section by noting an elementary consequence of the Bounded Distortion and Approximate-Translation Invariance properties.

Lemma 1. *Let $P \in \mathcal{P}$, $x \in \mathbb{R}^n$, $\tau \in B(0, 1)$. If $|P|_x \leq 1$, then*

$$|P|_{x+\tau} \leq 1 + C|\tau|.$$

Proof. Define the translate $P_\tau \in \mathcal{P}$, by setting $P_\tau(z) = P(z - \tau)$ for all $z \in \mathbb{R}^n$. We have

$$|P_\tau|_{x+\tau} \leq 1 + C|\tau|, \tag{23}$$

by the Approximate-Translation Invariance property.

On the other hand, by the Bounded Distortion Property, we have $|\partial^\alpha P(x)| \leq C$ for $|\alpha| \leq m$, which implies the estimate

$$|\partial^\alpha (P - P_\tau)(x + \tau)| \leq C|\tau| \quad \text{for } |\alpha| \leq m\,.$$

Another application of the Bounded Distortion Property now gives

$$|P - P_\tau|_{x+\tau} \leq C|\tau|. \tag{24}$$

The desired conclusion is immediate from (23) and (24). ■

4. Background from Computer Science

In this section, we recall some standard results from computer science. We start with the "Ellipsoid Algorithm" for linear programming.

Let us work in $\mathbb{R}^D$. We write ν, ν', ν_0, etc., for vectors in $\mathbb{R}^D$, and we write $\lambda, \lambda', \lambda_\ell$, etc., to denote linear functionals on $\mathbb{R}^D$. We denote a positive-definite quadratic form on $\mathbb{R}^D$ by $q(\cdot)$. An "ellipsoid" is a subset $E \subset \mathbb{R}^D$, given by

$$E = \{\nu \in \mathbb{R}^D : q(\nu - \nu_0) \leq 1\} \tag{1}$$

for a positive-definite quadratic form q and a vector ν_0.

A "linear constraint" in $\mathbb{R}^D$ is an inequality of the form $\lambda(\nu) \geq b$ (with $\lambda \in (\mathbb{R}^D)^*$ and $b \in \mathbb{R}$ given), for an unknown vector $\nu \in \mathbb{R}^D$. Given a list of linear constraints $\lambda_\ell(\nu) \geq b_\ell$ for $\ell = 1, \ldots, L$, the "feasible region" is defined as

$$K = \{\nu \in \mathbb{R}^D : \lambda_\ell(\nu) \geq b_\ell \text{ for } \ell = 1, \ldots, L\}.$$

Now suppose we are given a list of linear constraints, and a linear functional $\hat{\lambda} \in (\mathbb{R}^D)^*$. We would like to find a vector $\nu \in K$, with $\hat{\lambda}(\nu)$ as small as possible, or nearly so. Of course, this makes sense only if K is non-empty and $\hat{\lambda}$ is bounded below on K.

For the "ellipsoid algorithm", we assume that K is roughly comparable to a given ellipsoid E, in the sense that

$$K \subseteq E, \text{ and } \mathrm{vol}\, K \geq \lambda^D \,\mathrm{vol}\, E \text{ for given real number } \lambda > 0. \tag{2}$$

(In (2), λ does not denote a linear functional. We trust that no confusion will result.)

The idea of the ellipsoid method is that either the center of E belongs to K, or else one can trivially produce another ellipsoid E', such that $K \subset E'$ and $\mathrm{vol}\,(E') \leq (1 - c/D)\,\mathrm{vol}\,(E)$ for a universal constant c.

This allows us to perform the following algorithm.

Ellipsoid Algorithm: *Given positive real numbers* ϵ, λ, *with* $0 < \epsilon < 1/2$ *and* $0 < \lambda < 1/2$*; and given an ellipsoid* E *with center* ν_0*; and given a list of* L *linear constraints on* $\mathbb{R}^D$*, whose feasible region* K *satisfies*

(a) $K \subseteq E$ *and*

(b) $\operatorname{vol} K \geq \lambda^D \operatorname{vol} E$;

and given a linear functional $\hat{\lambda} \in (\mathbb{R}^D)^*$*; we compute a vector* $\nu_1 \in K$ *such that*

$$\hat{\lambda}(\nu_1) \leq \min\{\hat{\lambda}(\nu) : \nu \in K\} + \epsilon \cdot \max\{|\hat{\lambda}(\nu - \nu_0)| : \nu \in E\}.$$

The computation requires work at most $CD^4 L \log\left(\frac{1}{\lambda}\right) \log\left(\frac{1}{\epsilon}\right)$*, and storage at most* $C \cdot (D + L)^2$.

The following special case will be used below. We write

$$\mathcal{B}_D = \{(\nu_1, \ldots, \nu_D) \in \mathbb{R}^D : |\nu_i| \leq 1 \text{ for } i = 1, \ldots, D\}.$$

Special Ellipsoid Algorithm: *Given positive real numbers* $\epsilon, \wedge, \lambda$, *assumed to satisfy* $0 < \epsilon < 1/2$ *and* $2\lambda < \wedge$*; and given a list of* L *linear constraints in* $\mathbb{R}^D$*, whose feasible region* K *is assumed to satisfy*

(a) $K \subseteq \wedge \mathcal{B}_D$ *and*

(b) $K \supseteq \lambda \mathcal{B}_D + \tilde{\nu}$ *for an (unknown) vector* $\tilde{\nu} \in \mathbb{R}^D$;

we compute a vector $(\nu_1^0, \ldots, \nu_D^0) \in K$*, such that*

$$\nu_1^0 \leq \min\{\nu_1 : (\nu_1, \ldots, \nu_D) \in K\} + \epsilon \wedge .$$

The work required is at most $CD^4 L \log\left(\frac{\wedge D}{\lambda}\right) \log\left(\frac{D}{\epsilon}\right)$*, and the storage required is at most* $C \cdot (D + L)^2$.

See [29]. Also, see [28, 32] as a sample of the extensive literature on linear programming.

Next, we give some standard results from computational geometry. We start this discussion with the Well-Separated Pairs Decomposition due to Callahan and Kosaraju [11]. The results below are significantly weaker than those of [11]. We state here only what we will need below.

Let $E \subset \mathbb{R}^n$, with $\#(E) = N < \infty$; and let $0 < \kappa < 1$ be given. (We assume $N \geq 2$.) We write $A(\kappa)$, $A'(\kappa)$, etc., to denote constants determined by κ and n. Then there exists a finite list of Cartesian products, $E'_\nu \times E''_\nu$, $\nu = 1, \ldots, \nu_{\max}$, with the following properties:

- Each E'_ν and E''_ν is a non-empty subset of E.
- $\{(x', x'') \in E \times E : x' \neq x''\}$ is the disjoint union of the $E'_\nu \times E''_\nu$, $\nu = 1, \ldots, \nu_{\max}$.

- $\mathsf{diam}(E'_\nu)$, $\mathsf{diam}(E''_\nu) \leq \kappa \cdot \mathsf{dist}(E'_\nu, E''_\nu)$ for each ν.
- The number of $E'_\nu \times E''_\nu$ is $\nu_{\max} \leq A(\kappa)N$.
- There is an algorithm, called **Algorithm WSPD**, that accepts as input (E, κ), and produces as output a list of "representatives" $(x'_\nu, x''_\nu) \in E'_\nu \times E''_\nu$, $\nu = 1, \ldots, \nu_{\max}$.
  Algorithm WSPD consumes work at most $A'(\kappa)N \log N$, and storage at most $A''(\kappa)N$.

Later, we will take κ to be a small enough controlled constant. The quantities $A(\kappa)$, $A'(\kappa)$, $A''(\kappa)$ will then also be controlled constants.

We pass to the "Balanced Box Decomposition Tree" or "BBD Tree", due to Arya, Mount, Netanyahu, Silverman and Wu [1]. Again, we state only what we will need below, which is significantly weaker than the full results in [1].

BBD Tree Algorithm: *There is an algorithm with the following properties.*

- *The input for the algorithm consists of a non-empty finite set* $E \subset \mathbb{R}^n$, *with* $\#(E) = N$.
- *The algorithm performs one-time work, and then responds to queries as follows.*
- *A query consists of a point* $x \in \mathbb{R}^n$ *or a dyadic cube* Q *in* $\mathbb{R}^n$.
- *The response to a query* $x \in \mathbb{R}^n$ *is a point* $y \in E$ *such that* $|x - y| \leq 2\,\mathsf{dist}(x, E)$.
- *The response to a query* Q *is a "representative"* $x_Q \in E \cap Q$, *or a message indicating that* $E \cap Q$ *is empty.*
- *The one-time work is at most* $CN \log(N+1)$ *using storage at most* CN.
- *The work to answer a query is at most* $C \log(N + 1)$.

See also Sections 23 and 27 in [23], where the query algorithm for Q is spelled out in detail.

As an obvious consequence of the above, we have the following algorithm.

Algorithm Find-Representative: *Given a non-empty set* $E \subset \mathbb{R}^n$, *with* $\#(E) = N < \infty$, *we can perform one-time work* $CN \log(N+1)$ *in space* CN, *after which we can respond to queries as follows:*

- *A query consists of a dyadic cube* $Q \subset \mathbb{R}^n$.
- *The response to a query* Q *is a "representative"* $x_Q \in E \cap Q^{**}$, *or a message indicating that* $E \cap Q^{**}$ *is empty.*
- *The work to answer a query is at most* $C \log(N + 1)$.

In fact, we have only to partition Q^{**} into dyadic cubes $\{Q_\nu\}$, and apply the BBD Tree query algorithm to each Q_ν.

There is an analogue of Algorithm Find-Representative, with Q^{**} replaced by Q^*. We again give this algorithm the name "Find-Representative". Similarly, we may take Q^{***} in place of Q^{**}.

5. Gentle Partitions of Unity

We recall a result from our previous paper [21], and give two corollaries and a simple variant.

Lemma GPU: *Let $\{U_\nu\}$ be an open cover of an open set $\Omega \subset \mathbb{R}^n$, and let $\delta(x) > 0$ be defined for all $x \in \Omega$. Suppose that, for each ν, we are given functions $F_\nu \in C^m(U_\nu)$ and $\chi_\nu \in C^m(\Omega)$. Let $\epsilon, A_0, A_1, A_2 > 0$ and $M \geq 0$ be real numbers. Assume that the following conditions are satisfied.*

(GPU1) *Any given $x \in \Omega$ belongs to $\mathrm{supp}_\Omega \chi_\nu$ for at most A_0 distinct ν.*

(GPU2) $\sum_\nu \chi_\nu = 1$ *on* Ω.

(GPU3) $\chi_\nu \geq 0$ *on* Ω *for each* ν.

(GPU4) $\mathrm{supp}_\Omega \chi_\nu \subset U_\nu$ *for each* ν.

(GPU5) $|\partial^\alpha \chi_\nu(x)| \leq A_1 \epsilon \cdot (\delta(x))^{-|\alpha|}$ *for* $0 < |\alpha| \leq m$, $x \in \Omega$, *and for each* ν.

(GPU6) $|J_x(F_\nu)|_x \leq M$ *for* $x \in \mathrm{supp}_\Omega \chi_\nu$ *(any* ν*).*

(GPU7) $|\partial^\alpha(F_\nu - F_\mu)(x)| \leq A_2 M \cdot (\delta(x))^{m-|\alpha|}$ *for* $|\alpha| \leq m-1$, $x \in \mathrm{supp}_\Omega \chi_\nu \cap \mathrm{supp}_\Omega \chi_\mu$ *(any* μ, ν*).*

Then the function $F = \sum_\nu \chi_\nu F_\nu$ *belongs to* $C^m(\Omega)$, *and*

$$\| F \|_{C^m(\Omega)} \leq (1 + A'\epsilon)M\,,$$

where A' depends only on A_0, A_1, A_2, m, n, and on $\bar{c}_0, \bar{C}_0$ in the Bounded Distortion Property. *(See Section* 3.*)*

The above differs from "Lemma GPU" in [21] only in the most trivial details.

Corollary 1. *Let $y_0 \in \mathbb{R}^n$; let $\tilde{A} > 0$, $M \geq 0$, $\delta > 0$, $0 < \epsilon < 1$ be real numbers; and let $\theta_0 \in C^m(\mathbb{R}^n)$. Assume that $0 \leq \theta_0 \leq 1$ on $\mathbb{R}^n$; $\theta_0(x) = 1$ for $|x - y_0| \leq e^{1/(16\epsilon)}\,\delta$; $\mathrm{supp}\theta_0 \subset B(y_0, e^{1/(8\epsilon)}\,\delta)$; and*

$$|\partial^\alpha \theta_0(x)| \leq \tilde{A}\epsilon\,|x - y_0|^{-|\alpha|} \text{ for } 0 < |\alpha| \leq m\,,\ x \in \mathbb{R}^n \setminus \{y_0\}\,.$$

Also, let $F_0 \in C^m(B(y_0, e^{1/(8\epsilon)}\delta))$, $F_1 \in C^m(\mathbb{R}^n)$. *Assume that*

$$\| F_0 \|_{C^m(B(y_0, e^{1/(8\epsilon)}\delta))} \leq M,$$
$$\| F_1 \|_{C^m(\mathbb{R}^n)} \leq M, \quad \textit{and}$$
$$J_{y_0}(F_0) = J_{y_0}(F_1).$$

Then the function $F = \theta_0 F_0 + (1-\theta_0)F_1$ *belongs to* $C^m(\mathbb{R}^n)$, *and*

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + A'\epsilon)M,$$

with A' *depending only on* $\tilde{A}, m, n$ *and the constants* $\bar{c}_0, \bar{C}_0$ *in the* Bounded Distortion Property.

Proof. We write A', A'', etc., to denote constants determined by $\tilde{A}$, m, n, $\bar{c}_0$, $\bar{C}_0$ as in the conclusion of our Corollary.

We apply Lemma GPU with $\Omega = \mathbb{R}^n, U_0 = B(y_0, e^{1/(8\epsilon)}\delta)$, $U_1 = \mathbb{R}^n$, $\chi_0 = \theta_0$, $\chi_1 = 1 - \theta_0$, $\delta(x) = \delta + |x - y_0|$, F_0 and F_1 as in the hypotheses of Corollary 1, $A_0 = 2$, $A_1 = 2^m\tilde{A}$, and A_2= large enough constant of the form A'.

All the hypotheses of Lemma GPU are immediate here, except for (GPU5) and (GPU7). To check these, we argue as follows.

For $0 < |\alpha| \leq m$ and $|x - y_0| \geq \delta$, we have

$$|\partial^\alpha \chi_0(x)| = |\partial^\alpha \chi_1(x)| = |\partial^\alpha \theta_0(x)| \leq \tilde{A}\epsilon |x - y_0|^{-|\alpha|}$$
$$\leq (2^m\tilde{A})\epsilon \cdot (\delta + |x - y_0|)^{-|\alpha|}.$$

On the other hand, for $0 < |\alpha| \leq m$ and $|x - y_0| < \delta$, we have

$$|\partial^\alpha \chi_0(x)| = |\partial^\alpha \chi_1(x)| = 0, \quad \text{since } \theta_0 = 1 \text{ on } B(y_0, \delta).$$

(GPU5) is immediate from the above observations.

To check (GPU7), we first note that

$$|\partial^\alpha (F_0 - J_{y_0}(F_0))(x)| \leq A'M\,|x - y_0|^{m-|\alpha|}$$

for $|\alpha| \leq m - 1$, $x \in B(y_0, e^{1/(8\epsilon)}\delta)$; and

$$|\partial^\alpha (F_1 - J_{y_0}(F_1))(x)| \leq A'M\,|x - y_0|^{m-|\alpha|}$$

for $|\alpha| \leq m - 1$, $x \in B(y_0, e^{1/(8\epsilon)}\delta)$, thanks to our hypotheses on the C^m norms of F_0 and F_1 together with Taylor's theorem and the Bounded Distortion Property. Combining these last two estimates, and recalling that $J_{y_0}(F_0) = J_{y_0}(F_1)$, we find that

$$|\partial^\alpha (F_0 - F_1)(x)| \leq A''M|x - y_0|^{m-|\alpha|} \quad \text{for } |\alpha| \leq m - 1,\ x \in B(y_0, e^{1/8\epsilon}\delta),$$

from which (GPU7) follows at once. Thus Lemma GPU applies, and it yields the conclusion of Corollary 1. ∎

Corollary 2. *Let $0 < \epsilon_0 < 1$ and $M \geq 0$ be real numbers, let $Q_0 \subset \mathbb{R}^n$ be a cube, let $\theta_0 \in C^m(\mathbb{R}^n)$, and let $F_0 \in C^m(Q_0^{\mathrm{int}})$.*

Assume that $0 \leq \theta_0 \leq 1$ on $\mathbb{R}^n$;

$\operatorname{supp} \theta_0 \subset Q_0^{\mathrm{int}}$;
$|\partial^\alpha \theta_0(x)| \leq \epsilon_0$ *for* $0 < |\alpha| \leq m$, $x \in \mathbb{R}^n$; *and*
$\| F_0 \|_{C^m(Q_0^{\mathrm{int}})} \leq M$.

Then

$F = \theta_0 \cdot F_0$ *belongs to* $C^m(\mathbb{R}^n)$, *and*
$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon_0)M$.

Proof. Immediate from Lemma GPU, with $\Omega = \mathbb{R}^n$, $U_0 = Q_0^{\mathrm{int}}$, $U_1 = \mathbb{R}^n$, $\chi_0 = \theta_0$, $\chi_1 = 1 - \theta_0$, $\delta(x) = 1$ for all $x \in \mathbb{R}^n$, F_0 as given, $F_1 = 0$, $\epsilon = \epsilon_0$, $A_0 = 2$, $A_1 = 1$, $A_2 = 1/\bar{c}_0$, with $\bar{c}_0$ from the Bounded Distortion Property. ■

The next result is an easier variant of Lemma GPU.

Lemma LGPU: *Let $0 < \epsilon < 1$, $A \geq 1$, $M \geq 0$ be real numbers. Let $\hat{Q}_\nu$ be pairwise disjoint dyadic cubes of sidelength $\hat{\delta}$, where $A^{-1}\epsilon^{-1} \leq \hat{\delta} \leq \epsilon^{-1}$. Let $\theta_\nu \in C^m(\mathbb{R}^n)$, and assume that $\theta_\nu \geq 0$ on $\mathbb{R}^n$ (each ν), $\sum_\nu \theta_\nu \leq 1$ on $\mathbb{R}^n$, $\operatorname{supp} \theta_\nu \subset (\hat{Q}_\nu^*)^{\mathrm{int}}$ (each ν), and $|\partial^\alpha \theta_\nu(x)| \leq A\epsilon$ for $0 < |\alpha| \leq m$, $x \in \mathbb{R}^n$ (each ν).*

Let $F_\nu \in C^m((\hat{Q}_\nu^*)^{\mathrm{int}})$, with $\| F_\nu \|_{C^m((\hat{Q}_\nu^*)^{\mathrm{int}})} \leq M$ (each ν).

Then $F = \sum_\nu \theta_\nu F_\nu$ belongs to $C^m(\mathbb{R}^n)$, and

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + A'\epsilon)M,$$

where A' depends only on A, m, n and $\bar{c}_0$, $\bar{C}_0$ in the Bounded Distortion Property.

Proof. Obviously, $F \in C^m(\mathbb{R}^n)$. Our task is to estimate $\| F \|_{C^m(\mathbb{R}^n)}$. We write A', A'', etc., to denote constants determined by $A, m, n, \bar{c}_0, \bar{C}_0$.

Fix $x \in \mathbb{R}^n$, and note that

$$(1) \qquad J_x(F) = \sum_\nu \theta_\nu(x) J_x(F_\nu) + \sum_\nu \mathcal{E}_\nu, \text{ where}$$

$$(2) \qquad \mathcal{E}_\nu = J_x(\theta_\nu F_\nu) - \theta_\nu(x) J_x(F_\nu) \in \mathcal{P}.$$

By hypothesis, we have $|J_x(F_\nu)|_x \leq M$ for each ν; and $\sum_\nu \theta_\nu(x) \leq 1$ with each $\theta_\nu(x) \geq 0$. Since $|\cdot|_x$ is a norm, it follows that

$$(3) \qquad \Big| \sum_\nu \theta_\nu(x) J_x(F_\nu) \Big|_x \leq M.$$

We turn to the $\mathcal{E}_\nu$. From (2) we have

$$(4) \qquad \partial^\alpha \mathcal{E}_\nu(x) = \sum_{\substack{\beta+\gamma=\alpha \\ |\beta|\neq 0}} \tfrac{\alpha!}{\beta!\gamma!}\, \partial^\beta \theta_\nu(x) \cdot \partial^\gamma F_\nu(x).$$

For $0 < |\beta| \leq m$, we have $|\partial^\beta \theta_\nu(x)| \leq A\epsilon$; and, for $|\gamma| \leq m$, we have $|\partial^\gamma F_\nu(x)| \leq A'M$, thanks to our hypothesis on $\| F_\nu \|_{C^m}$ and the Bounded Distortion Property. Putting these remarks into (4), we find that

$$(5) \qquad |\partial^\alpha \mathcal{E}_\nu(x)| \leq A''\epsilon M \text{ for } |\alpha| \leq m \text{ (each } \nu).$$

Since x belongs to at most A''' distinct $\hat{Q}^*_\nu$, and since $\operatorname{supp} \theta_\nu \subset \hat{Q}^*_\nu$, we see from (2) that $\mathcal{E}_\nu$ is nonzero for at most A''' distinct ν. Hence, (5) implies

$$\left| \partial^\alpha \left[\sum_\nu \mathcal{E}_\nu \right](x) \right| \leq A'\epsilon M \quad \text{for } |\alpha| \leq m.$$

Another application of the Bounded Distortion Property now yields

$$(6) \qquad |\sum_\nu \mathcal{E}_\nu|_x \leq A''\,\epsilon M.$$

From (1), (3), (6), we see that $|J_x(F)|_x \leq (1 + A''\epsilon)M$. Since $x \in \mathbb{R}^n$ was arbitrary, it follows that $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + A''\epsilon)M$, proving Lemma LGPU. ■

6. The Main Patching Lemma

In this section and the next, we adapt to our purposes the arguments from Section 5 in [21]. Here, we give the set-up and state a result. The next section proves that result.

Our set-up is as follows. We are given real numbers $\epsilon > 0$ and $M \geq 0$, and positive constants $A_0, \ldots, A_7$. In addition, we suppose we are given the following objects.

Whitney Field: We suppose we are given a Whitney field $\vec{P} = (P^y)_{y \in E}$ on a finite set $E \subset \mathbb{R}^n$.
We assume that

$$(1) \qquad |\partial^\alpha (P^x - P^y)(x)| \leq A_0 M \cdot |x - y|^{m-|\alpha|} \text{ for } |\alpha| \leq m,\ x, y \in E,\ x \neq y.$$

Regularized Distance: We suppose we are given a function $\delta(\cdot)$ in $C^m_{\text{loc}}(\mathbb{R}^n \setminus E)$.
We assume that

$$(2) \qquad A_1^{-1} \operatorname{dist}(x, E) \leq \delta(x) \leq A_1 \operatorname{dist}(x, E) \text{ for all } x \in \mathbb{R}^n \setminus E, \text{ and}$$

$$(3) \qquad |\partial^\alpha \delta(x)| \leq A_2 \cdot (\delta(x))^{1-|\alpha|} \text{ for } |\alpha| \leq m,\ x \in \mathbb{R}^n \setminus E.$$

Partitions of Unity: We suppose we are given a C^m partition of unity

(4) $\quad 1 = \sum_{-\infty < \ell < \infty} \chi_\ell(t)$ on $\mathbb{R}$.

We assume for each ℓ that

(5) $\quad \chi_\ell(t) \geq 0$ for $t \in \mathbb{R}$; $\operatorname{supp} \chi_\ell \subset (\ell - 1, \ell + 1)$; $|\left(\frac{d}{dt}\right)^k \chi_\ell(t)| \leq A_3$ for $k \leq m$, $t \in \mathbb{R}$.

For each $\ell \in \mathbb{Z}$, we suppose we are given a C^m partition of unity

(6) $\quad 1 = \sum_\nu \theta_\nu^\ell$ on $\mathbb{R}^n$.

We assume, for each ℓ, ν, that

(7) $\quad \theta_\nu^\ell \geq 0$ on $\mathbb{R}^n$, and θ_ν^ℓ is supported in the interior of a cube Q_ν^ℓ of sidelength δ_ℓ, where

(8) $\quad A_4^{-1}\epsilon^{-1} \exp((\ell+1)/\epsilon) \leq \delta_\ell \leq A_4\epsilon^{-1} \exp((\ell+1)/\epsilon)$.

Also, for each ℓ, ν, we assume that

(9) $\quad |\partial^\alpha \theta_\nu^\ell(x)| \leq A_5 \cdot [\epsilon^{-1} \exp((\ell+1)/\epsilon)]^{-|\alpha|}$ for $|\alpha| \leq m$, $x \in \mathbb{R}^n$.

Regarding the cubes Q_ν^ℓ, we assume that

(10) For fixed $\ell \in \mathbb{Z}$ and $x \in \mathbb{R}^n$, there are at most A_6 distinct ν for which $x \in Q_\nu^\ell$.

Testing Sets: For each ℓ, ν, we suppose we are given a set

(11) $\quad S_\nu^\ell \subset E \cap (Q_\nu^\ell)^*$.

We assume that

(12) $\quad \operatorname{dist}(y, S_\nu^\ell) \leq A_7 \cdot \exp((\ell-1)/\epsilon)$ for all $y \in E \cap (Q_\nu^\ell)^*$.

Local Extending Functions: For each ℓ, ν, we suppose we are given a function $F_\nu^\ell \in C^m(\mathbb{R}^n)$. We assume that

(13) $\quad F_\nu^\ell$ agrees with $\vec{P}$ on S_ν^ℓ, and

(14) $\quad \| F_\nu^\ell \|_{C^m(\mathbb{R}^n)} \leq M$.

For each $x \in E$, we suppose we are given a function $F^x \in C^m(\mathbb{R}^n)$. We assume that

(15) $\quad J_x(F^x) = P^x$ for $x \in E$, and

(16) $\quad$ For each ℓ and ν, if $S_\nu^\ell = \{x\}$, then $F_\nu^\ell = F^x$.

We call a constant "weakly controlled" if it is determined by $A_0, \dots, A_7$ above, together with m, n and $\bar{c}_0, \bar{C}_0, \bar{C}_1$ from Section 3. We write A_1, A', A'', etc., to denote weakly controlled constants.

We patch together the F^ℓ_ν into a single function $\tilde{F}$ on $\mathbb{R}^n$, by setting

$$\tilde{F}(x) = P^x(x) \text{ for } x \in E, \text{ and} \tag{17}$$

$$\tilde{F}(x) = \sum_{\ell,\nu} \chi_\ell(\epsilon \log \delta(x)) \cdot \theta^\ell_\nu(x) \cdot F^\ell_\nu(x) \text{ for } x \in \mathbb{R}^n \smallsetminus E. \tag{18}$$

Under the above assumptions, we have the following result.

Main Patching Lemma: *Suppose ϵ is less than a small enough weakly controlled constant. Then $\tilde{F}$ belongs to $C^m(\mathbb{R}^n)$, agrees with $\vec{P}$, and satisfies $\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq (1 + A\epsilon)M$ for a weakly controlled constant A.*

The proof of this lemma will be given in the next section.

7. Proof of the Main Patching Lemma

In this section, we prove the Main Patching Lemma, using ideas from Section 5 of [21]. We adopt the convention that, for $k \geq 1$, the label $\langle k \rangle$ refers to equation (k) in Section 6 of our present paper. Also, we retain the assumptions and conventions of Section 6. In particular, A, A', A'', etc., denote "weakly controlled constants".

We adapt the arguments in Section 5 of [21], using the hypotheses of the Main Patching Lemma in place of equations (6), (8)...(13), (18)...(22), and (34), (35) in Section 5 of [21].

Let

$$\Omega = \mathbb{R}^n \smallsetminus E. \tag{1}$$

For each ℓ, ν, define

$$\chi^\ell_\nu(x) = \theta^\ell_\nu(x) \cdot \chi_\ell(\epsilon \log \delta(x)) \text{ for } x \in \Omega. \tag{2}$$

From $\langle 4 \rangle \dots \langle 7 \rangle$ and $\langle 10 \rangle$, we see that

(3) Any given $x \in \Omega$ belongs to $\mathrm{supp}_\Omega \chi^\ell_\nu$ for at most A distinct (ℓ, ν);

$$\sum_{\ell,\nu} \chi^\ell_\nu = 1 \text{ on } \Omega; \text{ and} \tag{4}$$

$$\chi^\ell_\nu \geq 0 \text{ on } \Omega. \tag{5}$$

Setting

$$U^\ell_\nu = \Omega \text{ for each } \ell, \nu, \text{ we have} \tag{6}$$

$$\mathrm{supp}_\Omega \chi^\ell_\nu \subset U^\ell_\nu, \tag{7}$$

by definition of supp_Ω.

We prepare to estimate the derivatives of χ_ν^ℓ.

As in Section 5 of [21], we first note that $\partial^\alpha[\chi_\ell(\epsilon \log \delta(\cdot))](x)$ is a sum of terms of the form

$$\prod_{\nu=1}^{r} \partial^{\beta_\nu}[\epsilon \log \delta(\cdot)](x) \cdot \chi_\ell^{(r)}(\epsilon \log \delta(x)),$$

where $\chi_\ell^{(r)}$ denotes the r^{th} derivative of χ_ℓ, and where $\beta_1 + \cdots + \beta_r = \alpha$ and each β_ν is non-zero. Moreover, each $\partial^{\beta_\nu}[\epsilon \log \delta(x)]$ is a sum of terms of the form

$$\epsilon \frac{\partial^{\gamma_1} \delta(x) \cdots \partial^{\gamma_{s_\nu}} \delta(x)}{(\delta(x))^{s_\nu}},$$

with $\gamma_1 + \cdots + \gamma_{s_\nu} = \beta_\nu$. Consequently, for $0 < |\alpha| \leq m$, the quantity $\partial^\alpha[\chi_\ell(\epsilon \log \delta(\cdot))](x)$ is a sum of terms

$$\epsilon^r \frac{\partial^{\gamma_1} \delta(x) \cdots \partial^{\gamma_s} \delta(x)}{(\delta(x))^s} \cdot \chi_\ell^{(r)}(\epsilon \log \delta(x)),$$

with $1 \leq r \leq m$ and $\gamma_1 + \cdots + \gamma_s = \alpha$. Thanks to $\langle 3 \rangle$ and $\langle 5 \rangle$, it follows that

(8) $\quad |\partial^\alpha[\chi_\ell(\epsilon \log \delta(\cdot))](x)| \leq A\epsilon \cdot (\delta(x))^{-|\alpha|}$ for $0 < |\alpha| \leq m$, $x \in \Omega$.

We note also that

(9) $\quad \exp((\ell - 1)/\epsilon) < \delta(x) < \exp((\ell + 1)/\epsilon)$ for $x \in \mathrm{supp}_\Omega \chi_\ell(\epsilon \log \delta(\cdot))$,

as we see at once from $\langle 5 \rangle$. Hence, for

$$x \in \mathrm{supp}_\Omega \chi_\nu^\ell = \mathrm{supp}_\Omega(\theta_\nu^\ell \cdot \chi_\ell(\epsilon \log \delta(\cdot))),$$

we have

(10) $\quad |\partial^\alpha \theta_\nu^\ell(x)| \leq A\epsilon^{|\alpha|} (\delta(x))^{-|\alpha|}$ for $|\alpha| \leq m$ (see $\langle 9 \rangle$).

Since $0 \leq \chi_\ell(\epsilon \log \delta(x)) \leq 1$ (see $\langle 4 \rangle$, $\langle 5 \rangle$), it follows from (8), (10) and our definition (2) of χ_ν^ℓ that

(11) $\quad |\partial^\alpha \chi_\nu^\ell(x)| \leq A\epsilon \cdot (\delta(x))^{-|\alpha|}$ for $0 < |\alpha| \leq m$, $x \in \Omega$, each ℓ, ν.

Thus, we have succeeded in estimating the derivatives of χ_ν^ℓ.

We next turn to the extending functions F_ν^ℓ in $\langle 13 \rangle$, $\langle 14 \rangle$. Since $F_\nu^\ell \in C^m(\mathbb{R}^n)$, we have in particular that

(12) $\quad F_\nu^\ell \in C^m(U_\nu^\ell)$.

From $\langle 14 \rangle$, we have

(13) $\quad |J_x(F_\nu^\ell)|_x \leq M$ for all $x \in \mathrm{supp}_\Omega \chi_\nu^\ell$ (any ℓ, ν).

We prepare to estimate the derivatives of $F^\ell_\nu - F^{\ell'}_{\nu'}$ at a point

(14) $x \in \mathrm{supp}_\Omega \chi^\ell_\nu \cap \mathrm{supp}_\Omega \chi^{\ell'}_{\nu'}$.

For x as in (14), we have $x \in \mathrm{supp}_\Omega \theta^\ell_\nu \subset Q^\ell_\nu$; and also $x \in \mathrm{supp}\chi_\ell(\epsilon \log \delta(\cdot))$, hence $\delta(x)$ is estimated by (9). Comparing (9) with $\langle 7\rangle$, $\langle 8\rangle$, we see that $\delta(x) \le A\epsilon$ sidelength (Q^ℓ_ν); hence $\langle 2\rangle$ yields a point

(15) $\bar{y} \in E$

such that

(16) $|x - \bar{y}| \le A\delta(x) \le A'\epsilon$ sidelength (Q^ℓ_ν).

Since $x \in Q^\ell_\nu$ and ϵ is less than a small enough weakly controlled constant, we learn from (15), (16) that $\bar{y} \in E \cap (Q^\ell_\nu)^*$. Consequently, $\langle 12\rangle$ applies and we obtain a point

(17) $y \in S^\ell_\nu$,

such that

(18) $|y - \bar{y}| \le A \cdot \exp((\ell - 1)/\epsilon) \le A\delta(x)$, thanks to (9).

From (16), (17), (18), we conclude that

(19) There exists $y \in S^\ell_\nu$, such that $|x - y| \le A\delta(x)$.

Similarly,

(20) There exists $y' \in S^{\ell'}_{\nu'}$, such that $|x - y'| \le A\delta(x)$.

Fix y, y' as in (19), (20), and note that

(21) $|y - y'| \le A\delta(x)$.

From $\langle 13\rangle$ and (19), we have

(22) $J_y(F^\ell_\nu) = P^y$,

and from $\langle 13\rangle$ and (20), we have

(23) $J_{y'}(F^{\ell'}_{\nu'}) = P^{y'}$.

From $\langle 14\rangle$, the Bounded Distortion Property, and Taylor's theorem, we obtain the estimate

$$|\partial^\alpha(F^\ell_\nu - J_y(F^\ell_\nu))(x)| \le AM|x - y|^{m-|\alpha|} \quad \text{for } |\alpha| \le m.$$

In view of (19) and (22), this implies that

(24) $|\partial^\alpha(F^\ell_\nu - P^y)(x)| \le AM(\delta(x))^{m-|\alpha|}$ for $|\alpha| \le m$.

Similarly,

(25) $|\partial^\alpha(F^{\ell'}_{\nu'} - P^{y'})(x)| \le AM(\delta(x))^{m-|\alpha|}$ for $|\alpha| \le m$.

Next, invoking $\langle 1 \rangle$, we see that

$$|\partial^\alpha (P^y - P^{y'})(y)| \leq AM|y - y'|^{m-|\alpha|} \quad \text{for } |\alpha| \leq m, \text{ if } y \neq y'.$$

Since $|y - y'|, |x - y| \leq A\delta(x)$ (see (19) and (21)), it follows that

$$|\partial^\alpha (P^y - P^{y'})(y)| \leq AM(\delta(x))^{m-|\alpha|} \quad \text{for } |\alpha| \leq m,$$

and hence

(26) $|\partial^\alpha (P^y - P^{y'})(x)| \leq AM(\delta(x))^{m-|\alpha|}$ for $|\alpha| \leq m$, since $P^y - P^{y'} \in \mathcal{P}$.

From (24), (25), (26), we obtain our desired estimate,

(27) $|\partial^\alpha (F^\ell_\nu - F^{\ell'}_{\nu'})(x)| \leq AM(\delta(x))^{m-|\alpha|}$ for $|\alpha| \leq m$, $x \in \mathrm{supp}_\Omega \chi^\ell_\nu \cap \mathrm{supp}_\Omega \chi^{\ell'}_{\nu'}$ (any (ℓ, ν), (ℓ', ν')).

We can now apply Lemma GPU to the open set Ω, the open cover $\{U^\ell_\nu\}$, the function $\delta(x)$, the partition of unity $\{\chi^\ell_\nu\}$, and the functions $F^\ell_\nu \in C^m(U^\ell_\nu)$.

The hypotheses of Lemma GPU are immediate from our present results (3), (4), (5), (7), (11), (13), (27).

Thus, we learn from Lemma GPU that the function $\tilde{F}$, defined in $\langle 17 \rangle$, $\langle 18 \rangle$, satisfies

(28) $\tilde{F} \in C^m(\mathbb{R}^n \setminus E)$, and $\| \tilde{F} \|_{C^m(\mathbb{R}^n \setminus E)} \leq (1 + A\epsilon)M$.

Next, we investigate how $\tilde{F}$ behaves on $\mathbb{R}^n \setminus E$ near a point of E. Fix

(29) $x_0 \in E$,

let $\triangle$ be a small enough positive number to be fixed later, and suppose

(30) $x \in \mathbb{R}^n \setminus E$, with

(31) $|x - x_0| < \triangle$.

Recall that E is finite; hence, we may suppose that

(32) $2\triangle < \mathrm{dist}(x_0, E \setminus \{x_0\})$.

For $y \in E \setminus \{x_0\}$, we then have

$$|x - y| \geq |x_0 - y| - |x_0 - x| > 2\triangle - \triangle = \triangle > |x - x_0|.$$

Consequently,

(33) $|x - x_0| = \mathrm{dist}(x, E)$.

Let (ℓ, ν) be such that

(34) $\quad \mathrm{supp}_\Omega \chi_\nu^\ell \ni x.$

Then $x \in \mathrm{supp}_\Omega \chi_\ell(\epsilon \log \delta(\cdot))$; hence (9) applies. From (9) and $\langle 2 \rangle$, we obtain

(35) $\quad A^{-1} \exp((\ell - 1)/\epsilon) < |x - x_0| < A \exp((\ell + 1)/\epsilon).$

Also, (34) yields

(36) $\quad x \in \mathrm{supp}_\Omega \theta_\nu^\ell \subset Q_\nu^\ell.$

Recall from $\langle 7 \rangle$, $\langle 8 \rangle$ that sidelength $(Q_\nu^\ell) = \delta_\ell < A\epsilon^{-1} \exp((\ell + 1)/\epsilon)$; together with (35) and (31), this yields

(37) $\quad \text{sidelength } (Q_\nu^\ell) < A\epsilon^{-1} \exp(2/\epsilon) \cdot |x - x_0| \leq A\epsilon^{-1} \exp(2/\epsilon) \cdot \triangle.$

On the other hand, $\langle 7 \rangle$ and $\langle 8 \rangle$ yield also

(38) $\quad \text{sidelength } (Q_\nu^\ell) = \delta_\ell > A^{-1}\epsilon^{-1} \exp((\ell + 1)/\epsilon) > A^{-1}\epsilon^{-1}|x - x_0|,$

thanks to (35). Since ϵ is less than a small enough weakly controlled constant, it follows from (29), (36) and (38) that

(39) $\quad x_0 \in (Q_\nu^\ell)^* \cap E.$

We next show that $E \cap (Q_\nu^\ell)^*$ consists of the single point x_0. In fact, if $y \in E \cap (Q_\nu^\ell)^*$ with $y \neq x_0$, then from (37) and (39) we would have

(40) $\quad |y - x_0| \leq A \cdot \text{ sidelength } (Q_\nu^\ell) \leq A\epsilon^{-1} \exp(2/\epsilon) \cdot \triangle.$

If $\triangle$ is small enough, then (40) cannot hold for two distinct points y, x_0 in the finite set E. We assume that $\triangle$ is small enough that (32) holds and (40) is impossible. We then obtain $E \cap (Q_\nu^\ell)^* = \{x_0\}$.

Together with $\langle 11 \rangle$ and $\langle 12 \rangle$, this shows that $S_\nu^\ell = \{x_0\}$. Therefore, from $\langle 16 \rangle$, we obtain the equality

(41) $\quad F_\nu^\ell = F^{x_0}.$

We have proven (41) for every (ℓ, ν) satisfying (34).
Consequently, for x_0, x as in (29), (30), (31), the definition $\langle 18 \rangle$ and (2) yield

$$\tilde{F}(x) = \sum_{\ell,\nu} \chi_\nu^\ell(x) \cdot F_\nu^\ell(x) = \sum_{\ell,\nu} \chi_\nu^\ell(x) \cdot F^{x_0}(x).$$

Recalling (4), we conclude that

(42) $\quad \tilde{F}(x) = F^{x_0}(x)$ for all $x \in \mathbb{R}^n \setminus E$ such that $|x - x_0| < \triangle.$

Here, x_0 is an arbitrary point of E, and $\triangle$ is a small enough positive number. We note that

(43) $F^{x_0} \in C^m(\mathbb{R}^n)$, and $\| F^{x_0} \|_{C^m(\mathbb{R}^n)} \leq M$, for $x_0 \in E$,

thanks to (41) and ⟨14⟩.
Also, we recall from ⟨15⟩ that

(44) $J_{x_0}(F^{x_0}) = P^{x_0}$ for any $x_0 \in E$.

We now define a function F on $\mathbb{R}^n$, by setting

(45) $F = \tilde{F}$ on $\mathbb{R}^n \setminus E$, and $F = F^{x_0}$ on $B(x_0, \triangle)$ for each $x_0 \in E$.

Note that (45) provides a consistent definition of a function F, since the balls $B(x_0, \triangle)$ ($x_0 \in E$) are pairwise disjoint for $\triangle$ small enough, and thanks to (42). Since $\tilde{F} \in C^m(\mathbb{R}^n \setminus E)$ and $F^{x_0} \in C^m(B(x_0, \triangle))$ for each $x_0 \in E$, it follows from (45) that

(46) $F \in C^m(\mathbb{R}^n)$.

(We can now pick $\triangle$ to be any positive number small enough that the above arguments work.) Moreover, (44) and (45) show that

(47) $J_{x_0}(F) = P^{x_0}$ for all $x_0 \in E$.

We estimate the C^m norm of F, by checking that

(48) $|J_x(F)|_x \leq (1 + A\epsilon)M$ for all $x \in \mathbb{R}^n$.

Indeed, for $x \in \mathbb{R}^n \setminus E_0$, (48) follows from (28) and (45); while, for $x \in E_0$, (48) follows from (43) and (45). Thus, (48) holds in all cases, and consequently,

(49) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + A\epsilon)M$.

We note that, in fact,

(50) $F = \tilde{F}$ on $\mathbb{R}^n$.

In fact, (50) holds on $\mathbb{R}^n \setminus E$, thanks to (45). For $x_0 \in E$, we have

$$F(x_0) = P^{x_0}(x_0) = \tilde{F}(x_0)$$

by (47) and ⟨17⟩. Thus (50) holds also on E.

In view of (50), our results (46), (47), (49) show that $\tilde{F} \in C^m(\mathbb{R}^n)$, $\tilde{F}$ agrees with $\vec{P}$, and $\| \tilde{F} \|_{C^m(\mathbb{R}^n)} \leq (1 + A\epsilon)M$. These are the conclusions of the Main Patching Lemma.

The proof of that Lemma is complete. ■

8. Comparing Polynomials at Representative Points

Let $E \subset \mathbb{R}^n$ be a finite set, and let $0 < \kappa < 1$ be a real number. Recall the representatives (x'_ν, x''_ν) $(1 \leq \nu \leq \nu_{\max})$ arising from the Well-Separated Pairs Decomposition from Section 4, for the parameter κ. As an application of the Well-Separated Pairs Decomposition, we recall the following lemma from [20]. (See also Har-Peled and Mendel [26].)

Lemma 8.1. *Suppose we are given an* $(m-1)^{\underline{\text{st}}}$ *degree polynomial* $\bar{P}^x$ *on* $\mathbb{R}^n$*, for each* $x \in E$*. Let* $M \geq 0$*, and assume that*

$$|\partial^\alpha(\bar{P}^{x'_\nu} - \bar{P}^{x''_\nu})(x'_\nu)| \leq M \cdot |x'_\nu - x''_\nu|^{m-|\alpha|} \quad \text{for } |\alpha| \leq m-1,\ 1 \leq \nu \leq \nu_{\max}.$$

Assume also that κ *is less than a small enough constant determined by* m, n*. Then, for any* $x, y \in E$*, we have*

$$|\partial^\alpha(\bar{P}^x - \bar{P}^y)(x)| \leq CM \cdot |x-y|^{m-|\alpha|} \quad \text{for } |\alpha| \leq m-1,$$

with C *depending only on* m *and* n*.*

In order to check efficiently that estimate (1) in Section 6 is satisfied, we will use the following variant of Lemma 8.1.

Lemma 8.2. *Let* $\vec{P} = (P^x)_{x \in E}$ *be a Whitney field on* E*, and let* $M \geq 0$*. Assume that* κ *is less than a small enough constant determined by* m, n*. Assume also that*

(1) $|\partial^\alpha P^y(y)| \leq M$ *for* $|\alpha| = m$*,* $y \in E$*; and that*

(2) $|\partial^\alpha(P^{x'_\nu} - P^{x''_\nu})(x'_\nu)| \leq M \cdot |x'_\nu - x''_\nu|^{m-|\alpha|}$ *for* $|\alpha| \leq m-1$*,* $1 \leq \nu \leq \nu_{\max}$*.*

Then

(3) $|\partial^\alpha(P^x - P^y)(x)| \leq CM \cdot |x-y|^{m-|\alpha|}$ *for* $|\alpha| \leq m$*,* $x, y \in E$*,* $x \neq y$*,*

with C *depending only on* m, n*.*

Proof. We reduce matters to Lemma 8.1. We write c, C, C', etc. (in this proof) to denote constants depending only on m, n. Let

$$P^x(z) = \sum_{|\alpha| \leq m} \lambda^x_\alpha \cdot (z-x)^\alpha \quad \text{for } x \in E,\ z \in \mathbb{R}^n.$$

Then (1) yields

(4) $|\lambda^x_\alpha| \leq CM$ for $|\alpha| = m$.

Define

$$\bar{P}^x(z) = \sum_{|\alpha| \leq m-1} \lambda^x_\alpha \cdot (z-x)^\alpha \quad \text{for } x \in E,\ z \in \mathbb{R}^n.$$

Thus, each $\bar{P}^x$ is an $(m-1)^{\underline{rst}}$ degree polynomial on $\mathbb{R}^n$. Also, for $x, y \in E$ and $|\alpha| \leq m-1$, we have

$$\partial^\alpha \bar{P}^y(x) = \partial^\alpha P^y(x) - \sum_{|\gamma| = m-|\alpha|} c(\alpha,\gamma)\, \lambda^y_{\alpha+\gamma} \cdot (x-y)^\gamma,$$

which implies that

(5) $|\partial^\alpha \bar{P}^y(x) - \partial^\alpha P^y(x)| \leq CM|x-y|^{m-|\alpha|}$ thanks to (4).

In view of (5), our hypothesis (2) implies the estimate

$$|\partial^\alpha(\bar{P}^{x'_\nu} - \bar{P}^{x''_\nu})(x'_\nu)| \leq CM|x'_\nu - x''_\nu|^{m-|\alpha|} \quad \text{for } |\alpha| \leq m-1,\ 1 \leq \nu \leq \nu_{\max}.$$

Therefore, Lemma 8.1 tells us that

(6) $|\partial^\alpha(\bar{P}^x - \bar{P}^y)(x)| \leq C'M|x-y|^{m-|\alpha|}$ for $|\alpha| \leq m-1$, $x, y \in E$, $x \neq y$.

The conclusion (3) of Lemma 9.2 follows from (5) and (6), for $|\alpha| \leq m-1$. For $|\alpha| = m$, conclusion (3) follows from (1), since $\partial^\alpha P^x$ and $\partial^\alpha P^y$ are constant polynomials. The proof of the lemma is complete. ■

9. Computing a Regularized Distance

Suppose $E \subset \mathbb{R}^n$ and $\#(E) = N < \infty$, with $N \geq 2$. In this section we show how to compute a function $\delta(x) > 0$, defined on $\mathbb{R}^n \setminus E$, and satisfying estimates (2) and (3) in Section 6. The idea goes back to Whitney [41]. To give an efficient algorithm, we bring in the BBD Tree from Section 4.

We define $\mathcal{Q}$ to be the set of all dyadic cubes $Q \subset \mathbb{R}^n$, such that

(1) $E \cap Q^{***} = \phi$,

but (1) fails for any dyadic cube strictly containing Q. For fixed $x \in \mathbb{R}^n \setminus E$, one checks easily that any sufficiently small dyadic cube containing x satisfies (1), while any sufficiently large dyadic cube containing x fails to satisfy (1). Consequently, $x \in Q$ for some $Q \in \mathcal{Q}$. Thus $\mathcal{Q}$ is a covering of $\mathbb{R}^n \setminus E$. Also, the cubes in $\mathcal{Q}$ are pairwise disjoint, since any two distinct dyadic cubes Q, Q' satisfy one of the three conditions $Q \cap Q' = \phi$, $Q \subset Q'$, $Q' \subset Q$. Moreover, each $Q \in \mathcal{Q}$ is obviously contained in $\mathbb{R}^n \setminus E$. The above remarks show that

(2) The cubes of $\mathcal{Q}$ form a partition of $\mathbb{R}^n \setminus E$.

Next, suppose $Q, Q' \in \mathcal{Q}$, and $Q^* \cap (Q')^* \neq \phi$. Then

(3) $\frac{1}{2}\delta_Q \leq \delta_{Q'} \leq 2\delta_Q$.

In fact, suppose (3) fails. Since Q, Q' are dyadic cubes, their sidelengths are powers of two. Therefore, δ_Q and $\delta_{Q'}$ must differ by at least a factor of 4. Without loss of generality, we may assume that $\delta_Q \leq \frac{1}{4}\delta_{Q'}$. Let Q^+ be the dyadic "parent" of Q, i.e., the dyadic cube for which $Q^+ \supset Q$, $\delta_{Q^+} = 2\delta_Q$. Then we have $\delta_{Q^+} \leq \frac{1}{2}\delta_{Q'}$ and $(Q^+)^* \cap (Q')^* \neq \phi$. Consequently, $(Q^+)^{***} \subset (Q')^{***}$. Therefore, (1) holds for the cube Q^+, since it holds for $Q' \in \mathcal{Q}$.

This contradicts our assumption that $Q \in \mathcal{Q}$, completing the proof of (3).

For any dyadic cube Q, we introduce a function $\varphi_Q \in C^m(\mathbb{R}^n)$, with the properties:

(4) $\varphi_Q \geq 0$ on $\mathbb{R}^n$; $\varphi_Q = 1$ on Q; $\operatorname{supp} \varphi_Q \subset Q^*$; and

(5) $|\partial^\alpha \varphi_Q| \leq C\,\delta_Q^{-|\alpha|}$ on $\mathbb{R}^n$, for $|\alpha| \leq m$.

By taking φ_Q to be an appropriate spline, we can satisfy (4), (5), and give a query algorithm as follows.

Algorithm 9.1. ("Find-jet-of φ_Q"): *Given a dyadic cube* Q *and a point* $x \in \mathbb{R}^n$*, we compute the jet* $J_x(\varphi_Q)$ *with work at most* C.

We now define the "regularized distance" by Whitney's formula

(6) $$\delta(x) = \sum_{Q \in \mathcal{Q}} \delta_Q \cdot \varphi_Q(x) \text{ for } x \in \mathbb{R}^n \setminus E.$$

Using the definition of $\mathcal{Q}$ and properties (2), (3), one checks easily that

(7) $$c\,\operatorname{dist}(x, E) \leq \delta_Q \leq C\,\operatorname{dist}(x, E) \text{ for } x \in Q^*,\ Q \in \mathcal{Q},$$

and therefore the function $\delta(x)$ in (6) satisfies

(8) $$c\,\operatorname{dist}(x, E) \leq \delta(x) \leq C\,\operatorname{dist}(x, E) \text{ for } x \in \mathbb{R}^n \setminus E$$

and

(9) $$|\partial^\alpha \delta(x)| \leq C \cdot (\delta(x))^{1-|\alpha|} \text{ for } |\alpha| \leq m,\ x \in \mathbb{R}^n \setminus E.$$

Thus, the properties (2), (3) in Section 6 hold for our function $\delta(x)$.

We prepare to compute the function $\delta(x)$.

Given $x \in \mathbb{R}^n \setminus E$, we define

(10) $$\mathcal{Q}(x) = \{Q \in \mathcal{Q} : x \in Q^*\}.$$

Since $\operatorname{supp} \varphi_Q \subset Q^*$ for any Q, a glance at (6) gives

(11) $$J_x(\delta(\cdot)) = \sum_{Q \in \mathcal{Q}(x)} \delta_Q \cdot J_x(\varphi_Q), \text{ for } x \in \mathbb{R}^n \setminus E.$$

Moreover, (7) shows that

(12) $\quad c\,\mathrm{dist}\,(x, E) \le \delta_Q \le C\,\mathrm{dist}\,(x, E)$ for $x \in \mathbb{R}^n \setminus E$, $Q \in \mathcal{Q}(x)$.

Our computation of $\delta(x)$ proceeds as follows

Algorithm 9.2. ("Compute-Regularized-Distance"): *After one-time work at most* $CN \log N$ *in space* CN*, we can answer queries as follows. Given* $x \in \mathbb{R}^n \setminus E$*, we compute* $J_x(\delta(\cdot))$ *with work at most* $C \log N$.

Explanation: We perform the one-time work of the BBD Tree, and of Algorithm "Find-Repepresentative", as in Section 4. Recall that this one-time work is at most $CN \log N$, and requires space at most CN.

Suppose we have done the above one-time work, and suppose we are given a query point $x \in \mathbb{R}^n \setminus E$. Then, using the BBD Tree Query algorithm, we can compute a number $d > 0$ such that

(13) $\quad \frac{1}{2}\,\mathrm{dist}\,(x, E) \le d \le 2\mathrm{dist}\,(x, E)$.

The computation of d requires work at most $C \log N$. From (10), (12) and (13), we have

(14) $\quad cd \le \delta_Q \le Cd$ and $Q^* \ni x$ for each $Q \in \mathcal{Q}(x)$.

There are at most C dyadic cubes Q satisfying (14) for given x, d; and it takes work at most C to list them. Let $Q_1, \ldots, Q_L$ be a list of all the dyadic cubes satisfying (14).

Next, we test each $Q_\ell (1 \le \ell \le L)$, to decide whether $Q_\ell \in \mathcal{Q}$. To do so, we let Q_ℓ^+ be the dyadic "parent" of Q_ℓ, i.e., the dyadic cube containing Q_ℓ, with sidelength twice that of Q_ℓ. Then, by definition of $\mathcal{Q}$, we have $Q_\ell \in \mathcal{Q}$ if and only if

(15) $\quad E \cap Q_\ell^{***} = \phi$ but $E \cap (Q_\ell^+)^{***} \ne \phi$.

We can test whether (15) holds, thanks to Algorithm "Find-Representative" from Section 4, applied to the two dyadic cubes Q_ℓ and Q_ℓ^+. For each Q_ℓ, this requires work at most $C \log N$, and there are at most C distinct Q_ℓ. Thus, it takes work at most $C \log N$ to decide which of the cubes $Q_1, \ldots, Q_L$ belong to $\mathcal{Q}$.

However, one checks easily that $\mathcal{Q}(x)$ is precisely the set of all the cubes Q_ℓ $(1 \le \ell \le L)$ that belong to $\mathcal{Q}$.

Thus, we have produced a list of all the cubes in $\mathcal{Q}(x)$. There are at most C such cubes. We can now trivially compute $J_x(\delta(\cdot))$, using equation (11) and Algorithm 9.1. The work of this last step is at most C. This completes our explanation of Algorithm 9.2.

10. Computing Partitions of Unity

In this section, we compute cutoff functions $\chi_\ell(t)$, $\theta^\ell_\nu(x)$, as well as cubes Q^ℓ_ν, satisfying conditions (4)...(10) in Section 6. We also compute additional cutoff functions that will be used later.

By taking $\chi_0(t)$ to be an appropriate spline on $\mathbb{R}$, and then defining $\chi_\ell(t) = \chi_0(t-\ell)$ for $\ell \in \mathbb{Z}$, we can arrange that

(1) $\chi_\ell \in C^m(\mathbb{R})$ for each $\ell \in \mathbb{Z}$,

and that for each $\ell \in \mathbb{Z}$, we have

(2) $\chi_\ell \geq 0$ on $\mathbb{R}$; $\operatorname{supp} \chi_\ell \subset (\ell-1, \ell+1)$; $|\left(\frac{d}{dt}\right)^k \chi_\ell(t)| \leq C$ for $|k| \leq m$, $t \in \mathbb{R}$;

and that

(3) $\sum_{\ell\in\mathbb{Z}} \chi_\ell(t) = 1$ for all $t \in \mathbb{R}$;

moreover, we can answer queries as follows.

Algorithm 10.1. ("Compute-χ_ℓ"): *Given $\ell \in \mathbb{Z}$, $t \in \mathbb{R}$, $0 \leq k \leq m$, we compute $\left(\frac{d}{dt}\right)^k \chi_\ell(t)$ with work at most C.*

In view of (1), (2), (3), the χ_ℓ form a C^m partition of unity satisfying (4) and (5) in Section 6.

Next, we prepare to define and compute cubes Q^ℓ_ν and cutoff functions θ^ℓ_ν as in (6)...(10) in Section 6.

For $s \in \mathbb{Z}$, and for any lattice point $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$, we let $Q^{\langle s\rangle}_\nu$ denote the dyadic cube $Q^{\langle s\rangle}_\nu = [2^s \cdot \nu_1, 2^s \cdot (\nu_1 + 1)) \times \cdots \times [2^s \cdot \nu_n, 2^s \cdot (\nu_n + 1)) \subset \mathbb{R}^n$.

Thus, for fixed $s \in \mathbb{Z}$, the $Q^{\langle s\rangle}_\nu$ ($\nu \in \mathbb{Z}^n$) partition $\mathbb{R}^n$ into dyadic cubes of sidelength 2^s.

By taking $\hat\theta^{\langle 0\rangle}_0 \in C^m(\mathbb{R}^n)$ to be an appropriate spline, and then defining $\hat\theta^{\langle s\rangle}_\nu(x) = \hat\theta^{\langle 0\rangle}_0(2^{-s}x - \nu)$ for $s \in \mathbb{Z}$ and $\nu \in \mathbb{Z}^n$, we can arrange that, for each s, ν, we have:

$\hat\theta^{\langle s\rangle}_\nu \in C^m(\mathbb{R}^n)$;

$\hat\theta^{\langle s\rangle}_\nu \geq 0$ on $\mathbb{R}^n$;

$\hat\theta^{\langle s\rangle}_\nu \geq 1$ on $Q^{\langle s\rangle}_\nu$;

$\operatorname{supp} \hat\theta^{\langle s\rangle}_\nu \subset \left[\left(Q^{\langle s\rangle}_\nu\right)^*\right]^{\mathrm{int}}$ (recall, "int" denotes the interior);

$|\partial^\alpha \hat\theta^{\langle s\rangle}_\nu(x)| \leq C \cdot 2^{-s|\alpha|}$ for $|\alpha| \leq m$, $x \in \mathbb{R}^n$;

and we can answer queries as follows.

Algorithm 10.2. *Given $s \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$, $x \in \mathbb{R}^n$, we compute the jet $J_x(\hat{\theta}_\nu^{\langle s\rangle})$ with work at most C.*

We now define

$$\theta_\nu^{\langle s\rangle}(x) \;=\; \hat{\theta}_\nu^{\langle s\rangle}(x) \Big/ \sum_{\nu' \in \mathbb{Z}^n} \hat{\theta}_{\nu'}^{\langle s\rangle}(x) \quad \text{for } x \in \mathbb{R}^n,\ s \in \mathbb{Z},\ \nu \in \mathbb{Z}^n.$$

Here, we may restrict the sum to run over only those $\nu' \in \mathbb{Z}^n$ such that $x \in (Q_{\nu'}^{\langle s\rangle})^*$.
From the properties of the $\hat{\theta}_\nu^{\langle s\rangle}$, we see that the $\theta_\nu^{\langle s\rangle}$ satisfy the following.

$\theta_\nu^{\langle s\rangle} \in C^m(\mathbb{R}^n)$;
$\theta_\nu^{\langle s\rangle} \geq 0$ on $\mathbb{R}^n$;
$\operatorname{supp} \theta_\nu^{\langle s\rangle} \subset [(Q_\nu^{\langle s\rangle})^*]^{\mathrm{int}}$;
$\sum_{\nu \in \mathbb{Z}^n} \theta_\nu^{\langle s\rangle} = 1$ on $\mathbb{R}^n$;
$|\partial^\alpha \theta_\nu^{\langle s\rangle}(x)| \leq C \cdot 2^{-s|\alpha|}$ for $|\alpha| \leq m$, $x \in \mathbb{R}^n$;

and we can answer queries as follows.

Algorithm 10.3. *Given $s \in \mathbb{Z}$, $\nu \in \mathbb{Z}^m$, $x \in \mathbb{R}^n$, we compute $J_x(\theta_\nu^{\langle s\rangle})$ with work at most C.*

Now suppose we are given $0 < \epsilon < 1$ and $\ell \in \mathbb{Z}$. With work at most C, we can compute an $s \in \mathbb{Z}$, such that

$$\tfrac{1}{32}\,\epsilon^{-1} \exp((\ell+1)/\epsilon) \;\leq\; 2^s \;\leq\; \tfrac{1}{8}\,\epsilon^{-1} \exp((\ell+1)/\epsilon). \tag{4}$$

Fix s as in (4). For any $\nu \in \mathbb{Z}^n$, we then define

$$Q_\nu^\ell \;=\; (Q_\nu^{\langle s\rangle})^*, \quad \theta_\nu^\ell \;=\; \theta_\nu^{\langle s\rangle}, \quad \hat{Q}_\nu^\ell \;=\; Q_\nu^{\langle s\rangle}\,.$$

Also, we define $\delta_\ell = 3 \cdot 2^s$.

The properties of the $\theta_\nu^{\langle s\rangle}$ and $Q_\nu^{\langle s\rangle}$ then yield the following.

(5) Each θ_ν^ℓ belongs to $C^m(\mathbb{R}^n)$.

(6) $\sum_{\nu \in \mathbb{Z}^n} \theta_\nu^\ell = 1$ on $\mathbb{R}^n$, for each $\ell \in \mathbb{Z}$.

For each ℓ, ν,

(7) $\theta_\nu^\ell \geq 0$ on $\mathbb{R}^n$, and θ_ν^ℓ is supported in the interior of the cube Q_ν^ℓ;

(8) the sidelength of Q_ν^ℓ is δ_ℓ; and

$$\tfrac{1}{32}\,\epsilon^{-1} \exp((\ell+1)/\epsilon) \;\leq\; \delta_\ell \;\leq\; \epsilon^{-1} \exp((\ell+1)/\epsilon). \tag{9}$$

Also, for each ℓ, ν, we have

(10) $|\partial^\alpha \theta_\nu^\ell(x)| \leq C \cdot [\epsilon^{-1} \exp((\ell+1)/\epsilon)]^{-|\alpha|}$ for $|\alpha| \leq m$, $x \in \mathbb{R}^n$.

Moreover,

(11) For fixed $\ell \in \mathbb{Z}$ and $x \in \mathbb{R}^n$, we have $x \in Q_\nu^\ell$ for at most C distinct ν.

Furthermore, we can answer queries as follows.

Algorithm 10.4. ("Compute θ_ν^ℓ"): *Given $0 < \epsilon < 1$, $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$, $x \in \mathbb{R}^n$, we compute the jet $J_x(\theta_\nu^\ell)$, as well as the cubes $\hat{Q}_\nu^\ell$, Q_ν^ℓ. The work of this algorithm is at most C.*

Note that, by definition,

(12) Each $\hat{Q}_\nu^\ell$ is a dyadic cube, and $Q_\nu^\ell = (\hat{Q}_\nu^\ell)^*$.

In view of (5)...(12), the θ_ν^ℓ form a C^m partition of unity, satisfying conditions (6)...(10) in Section 6.

This concludes our discussion of the θ_ν^ℓ. Regarding the Q_ν^ℓ, we note that it is trivial to answer queries as follows.

Algorithm 10.5. ("Find-Relevant-Cubes"): *Given $\ell \in \mathbb{Z}$ and $x \in \mathbb{R}^n$, we produce a list of all the $\nu \in \mathbb{Z}^n$ such that $x \in (Q_\nu^\ell)^*$. There are at most C such $\nu \in \mathbb{Z}^n$, and the work of the algorithm is at most C.*

Next, we compute a cutoff function that will be used later in computing local extending functions as in (13)...(16) in Section 6.

Algorithm 10.6. *Given a point $y_0 \in \mathbb{R}^n$ and numbers $\delta > 0$ and $0 < \epsilon < 1$, we compute a function $\theta_0 \in C^{m+1}(\mathbb{R}^n)$, with the following properties:*

(a) $0 \leq \theta_0 \leq 1$ *on* $\mathbb{R}^n$;

(b) $\theta_0(x) = 1$ *for* $|x - y_0| \leq e^{1/(16\epsilon)}\delta$;

(c) $\operatorname{supp} \theta_0 \subset B(y_0, e^{1/(8\epsilon)}\delta)$; *and*

(d) $|\partial^\alpha \theta_0(x)| \leq C\epsilon \cdot |x - y_0|^{-|\alpha|}$ *for* $0 < |\alpha| \leq m+1$, $x \in \mathbb{R}^n \setminus \{y_0\}$.

The one-time work to compute θ_0 is zero, the storage used is at most C, and the work to answer a query (by computing $J_{\underline{x}}(\theta_0)$ at a query point $\underline{x}$) is at most C.

Explanation: By taking $\chi \in C^{m+1}(\mathbb{R})$ to be an appropriate spline, we can arrange the following.

(13) $0 \leq \chi \leq 1$; $\chi(t) = 1$ for $t \leq 1/16$; $\operatorname{supp} \chi \subset (-\infty, 1/8)$; $|\chi^{(r)}(t)| \leq C$ for $0 \leq r \leq m+1$, $t \in \mathbb{R}$ (where $\chi^{(r)}$ denotes the $r^{\underline{\text{th}}}$ derivative of χ); and,

(14) given $0 \leq r \leq m$, $t \in \mathbb{R}$, we can compute $\chi^{(r)}(t)$ with work at most C.

We then define

$$\theta_0(x) = \chi\Big(\epsilon \log \frac{|x - y_0|}{\delta}\Big) \quad \text{for } x \neq y_0,\ \theta_0(y_0) = 1.$$

Evidently, $\theta_0 \in C^m(\mathbb{R}^n)$, and (a), (b), (c) hold for θ_0. Also, evidently, $J_{\underline{x}}(\theta_0)$ can be computed with work at most C. It remains to check properly (d) for the function θ_0. Suppose $x \in \mathbb{R}^n \setminus \{y_0\}$. Then, for $0 < |\alpha| \leq m + 1$, the quantity $\partial^\alpha \theta_0(x)$ is a sum of terms of the form

$$(15) \quad \prod_{\nu=1}^{r} \Big(\partial^{\beta_\nu}[\epsilon \log \frac{|x - y_0|}{\delta}]\Big) \cdot \chi^{(r)}\Big(\epsilon \log \frac{|x - y_0|}{\delta}\Big).$$

with $\beta_1 + \cdots + \beta_r = \alpha$, and with each β_ν non-zero. Since $|\chi^{(r)}(t)| \leq C$ for all $t \in \mathbb{R}$, and since

$$\Big|\partial^{\beta_\nu}\Big[\epsilon \log \frac{|x - y_0|}{\delta}\Big]\Big| \leq C\epsilon|x - y_0|^{-|\beta_\nu|} \quad \text{for } 0 < |\beta_\nu| \leq m + 1,$$

it follows that each term (15) is less than or equal in absolute value to $C\epsilon^r|x - y_0|^{-|\alpha|}$, with $0 < r \leq m + 1$. This immediately implies (d), since $0 < \epsilon < 1$. Thus, our function θ_0 has all the required properties.

11. Computing Testing Sets

The goal of this section is to compute suitable "testing sets" S_ν^ℓ satisfying conditions (11) and (12) in Section 6.

Algorithm 11.1. ("Find-Testing-Set"): *Given $0 < \epsilon < 1$; given $E \subset \mathbb{R}^n$ with $\#(E) = N < \infty$; and given a dyadic cube $Q \subset \mathbb{R}^n$; we compute a finite set $S(Q) \subset E \cap Q^{**}$, such that*

(a) $|y - y'| > c\epsilon e^{-2/\epsilon}\,\delta_Q$ *for any two distinct points $y, y' \in S(Q)$; and such that*

(b) $\mathrm{dist}(y, S(Q)) < C\epsilon e^{-2/\epsilon}\,\delta_Q$ *for any $y \in E \cap Q^{**}$.*

After the one-time work of the BBD Tree Algorithm (See Section 4), the work to compute $S(Q)$ is at most $\exp(C/\epsilon) \cdot \log(N + 1)$.

The storage needed is at most $CN + \exp(C/\epsilon)$.

Explanation: Recall that δ_Q denotes the sidelength of Q. With work at most $\exp(C/\epsilon)$, we partition Q^{**} into dyadic cubes Q_ν ($1 \leq \nu \leq \nu_{\max}$) of common sidelength δ_{Q_ν}, such that

$$(1) \quad c\epsilon e^{-2/\epsilon}\,\delta_Q \leq \delta_{Q_\nu} \leq C\epsilon e^{-2/\epsilon}\,\delta_Q, \text{ and } \nu_{\max} \leq \exp(C/\epsilon).$$

For each Q_ν, we check whether $E \cap Q_\nu$ is empty; and, if $E \cap Q_\nu \neq \phi$, then we compute a point $y_\nu \in E \cap Q_\nu$.

To do so, we use the BBD Tree Algorithm. After the one-time work of the BBD Tree Algorithm, the work to examine a single Q_ν is at most $C\log(N+1)$. Hence, the work to produce the set

$$\tilde{S}(Q) = \{y_\nu : E \cap Q_\nu \neq \phi\} \tag{2}$$

is at most $C\log(N+1)\cdot\nu_{\max} \leq \exp(C'/\epsilon)\log(N+1)$. Obviously, the set $\tilde{S}(Q)$ in (2) satisfies

$$\tilde{S}(Q) \subset E \cap Q^{**}, \tag{3}$$

and also

$$\mathrm{dist}\,(y, \tilde{S}(Q)) \leq C\epsilon e^{-2/\epsilon}\,\delta_Q \text{ for any } y \in E \cap Q^{**}. \tag{4}$$

Unfortunately, $\tilde{S}(Q)$ may fail to satisfy condition (a) above. Therefore, we proceed as follows.

By induction on ν, we decide whether or not to discard y_ν, according to the "Vitali rule":

(5) We discard y_ν if and only if there exists $\nu' < \nu$, such that $y_{\nu'}$ was not discarded, and $|y_\nu - y_{\nu'}| \leq \epsilon e^{-2/\epsilon}\,\delta_Q$.

Let $S(Q)$ be the set of all the points $y_\nu \in \tilde{S}(Q)$ that were not discarded. We can compute $S(Q)$ from $\tilde{S}(Q)$ with work at most $C(\nu_{\max})^2 \leq \exp(C/\epsilon)$. Note that $S(Q) \subseteq \tilde{S}(Q) \subset E \cap Q^{**}$. Moreover, we cannot have

$$|y_\nu - y_{\nu'}| \leq \epsilon e^{-2/\epsilon}\delta_Q \text{ for two distinct points } y_\nu, y_{\nu'} \in S(Q). \tag{6}$$

In fact, suppose (6) holds. Without loss of generality, we may assume $\nu' < \nu$. Since $y_{\nu'} \in S(Q)$, the point $y_{\nu'} \in \tilde{S}(Q)$ was not discarded. Consequently, (5) and (6) tell us that y_ν is discarded, contradicting the assumption from (6) that $y_\nu \in S(Q)$.

Thus, as claimed, (6) cannot hold. The set $S(Q)$ therefore satisfies condition (a). Let us check that $S(Q)$ also satisfies condition (b).

Thus, let $y \in E \cap Q^{**}$. From (4), we have

$$|y - y'| \leq C\epsilon e^{-2/\epsilon}\,\delta_Q \text{ for some } y' \in \tilde{S}(Q). \tag{7}$$

Fix y' as in (7). If y' belongs to $S(Q)$, then (7) shows at once that

$$\mathrm{dist}(y, S(Q)) \leq C\epsilon e^{-2/\epsilon}\,\delta_Q. \tag{8}$$

On the other hand, if y' does not belong to $S(Q)$, then, according to our rule (5), there exists $y'' \in S(Q)$ such that $|y' - y''| \leq \epsilon e^{-2/\epsilon}\,\delta_Q$.

Together with (7), this shows that

$$\mathrm{dist}(y, S(Q)) \leq |y - y''| \leq |y - y'| + |y' - y''| \leq C\epsilon e^{-2/\epsilon}\,\delta_Q,$$

and hence again (8) holds. Thus, (8) holds in all cases, completing the proof of (b).

We have computed a set $S(Q) \subset E \cap Q^{**}$, satisfying (a) and (b). After the one-time work of the BBD Tree Algorithm, the total work to compute $S(Q)$ is at most $\exp(C/\epsilon) \cdot \log N$.

Regarding the storage needed for Algorithm 11.1, we first recall that we need space CN to allow us to use the BBD Tree. We then generate the cubes Q_ν and the points y_ν one at a time. We need to store $\tilde{S}(Q)$ from (2), which requires at most space $\exp(C/\epsilon)$. In a crude implementation, we can mark each of the y_ν to indicate whether it is discarded according to (5); and then we generate and store the set $S(Q)$. The space needed for these last steps is at most $\exp(C/\epsilon)$. Thus, the total storage required for Algorithm 11.1 is at most $CN + \exp(C/\epsilon)$.

The explanation of Algorithm 11.1 is complete.

For most dyadic cubes Q, the set $S(Q)$ produced by Algorithm 11.1 will be empty or a singleton. This fact will be clear from the following algorithm.

Algorithm 11.2. ("Find-Interesting-Cubes"): *Given* $0 < \epsilon < 1$*, and given* $E \subset \mathbb{R}^n$ *with* $\#(E) = N$, $2 \le N < \infty$*, we produce a list of dyadic cubes,* $Q^{(1)}, \ldots, Q^{(L)}$*, with the following properties.*

(a) *For* $1 \le \lambda \le L$*, the set* $S(Q^{(\lambda)})$ *computed from* $\epsilon, E, Q^{(\lambda)}$ *by Algorithm* 11.1 *has cardinality at least two.*

(b) *For any dyadic cube* Q *other than* $Q^{(1)}, \ldots, Q^{(L)}$*, the set* $S(Q)$ *computed from* ϵ, E, Q *by Algorithm* 11.1 *has cardinality at most one.*

(c) *No cube appears more than once in the list* $Q^{(1)}, \ldots, Q^{(L)}$.

(d) $L \le (C/\epsilon) \cdot N$.

(e) *The work to compute* $Q^{(1)}, \ldots, Q^{(L)}$ *is at most* $\exp(C/\epsilon) \cdot N \log N$ *and the storage needed is at most* $\frac{C}{\epsilon} N + \exp(C/\epsilon)$.

Explanation: First we discuss the properties of dyadic cubes Q such that $\#(S(Q)) \ge 2$. Then we give the algorithm to compute the list $Q^{(1)} \ldots Q^{(L)}$.

Let Q be a dyadic cube such that the set $S(Q)$ computed from ϵ, E, Q by Algorithm 11.1 has cardinality at least 2. Let y', y'' be two distinct points in $S(Q)$. By the defining properties of $S(Q)$, we have

(9) $\quad y', y'' \in E \cap Q^{**}$, and

(10) $\quad |y' - y''| > c\epsilon e^{-2/\epsilon}\, \delta_Q$.

Let x'_ν, x''_ν, E'_ν, E''_ν ($1 \le \nu \le \nu_{\max}$) be as in our discussion of the Well-Separated Pairs Decomposition in Section 4, with κ less than a small enough controlled constant. Since y', y'' are two distinct points of E, we have

(11) $\quad y' \in E'_\nu$ and $y'' \in E''_\nu$ for some ν.

Fix ν as in (11). We recall from Section 4 that also

(12) $x'_\nu \in E'_\nu$, $x''_\nu \in E''_\nu$, and

(13) $\mathrm{diam}(E'_\nu)$, $\mathrm{diam}(E''_\nu) \leq \kappa\, \mathrm{dist}(E'_\nu, E''_\nu)$.

From (9), (11), (12), (13), we learn that

(14) $|x'_\nu - y'|$, $|x''_\nu - y''| \leq \kappa |y' - y''| \leq C\kappa\delta_Q$,

and consequently,

(15) x'_ν, $x''_\nu \in Q^{***}$,

by another application of (9). Returning to (14), and applying (10), we see also that

(16) $|x'_\nu - x''_\nu| \geq \frac{1}{2}|y' - y''| \geq c'\epsilon e^{-2/\epsilon}\,\delta_Q$.

From (15), (16), we have learned the following: For any dyadic cube Q such that $\#(S(Q)) \geq 2$, there exists ν ($1 \leq \nu \leq \nu_{\max}$), such that

(17) $x'_\nu, x''_\nu \in Q^{***}$, and $\delta_Q \leq C\epsilon^{-1}e^{2/\epsilon}|x'_\nu - x''_\nu|$.

Now we can describe how to carry out Algorithm 11.2.

Step 1: First, we do the one-time work of the BBD Tree and the WSPD (for a small enough controlled constant κ).

In particular, we produce the "representatives" x'_ν, x''_ν ($1 \leq \nu \leq \nu_{\max}$) from the WSPD. Recall that $\nu_{\max} \leq CN$, and that the above one-time work is at most $CN \log N$, using storage CN.

Step 2: Next, for each ν ($1 \leq \nu \leq \nu_{\max}$), we list all the dyadic cubes Q such that (17) holds. For a given ν there are at most C/ϵ such cubes, and we can compute them with work at most C/ϵ.

Step 3: For each dyadic cube Q produced in Step 2, we use Algorithm 11.1 to compute $S(Q)$, and we check whether $\#(S(Q)) \geq 2$. If $\#(S(Q)) \geq 2$, then we add Q to a list $\mathcal{L}$ of "interesting cubes".

Because $\nu_{\max} \leq CN$, the number of cubes in our list $\mathcal{L}$ at the end of Step 3 is at most $(C/\epsilon) \cdot N$. Since $\nu_{\max} \leq CN$ and Algorithm 11.1 computes a single $S(Q)$ with work at most $\exp(C/\epsilon)\log N$, in space $CN + \exp(C/\epsilon)$, we see that the total work needed for Steps 2 and 3 is at most $\exp(C/\epsilon)N\log N$, and the storage required for these steps is at most $\frac{C}{\epsilon}N + \exp(C/\epsilon)$.

Thanks to our result (17), we know that any dyadic cube Q for which Algorithm 11.1 produces a set $S(Q)$ with $\#(S(Q)) \geq 2$ must appear on the list $\mathcal{L}$.

On the other hand, every cube Q appearing on our list $\mathcal{L}$ is such that Algorithm 11.1 produces a set $S(Q)$ with $\#(S(Q)) \geq 2$. (That's immediate from inspection of Step 3.)

Step 4: Finally, we sort our list $\mathcal{L}$, and remove duplicates, to guarantee that no cube Q appears more than once in our final list $\mathcal{L}^0$. Since the length of $\mathcal{L}$ is at most $\frac{C}{\epsilon} \cdot N$, the work of Step 4 is at most $(C/\epsilon)N \cdot \log((C/\epsilon)N)$, which is less than $\exp(C/\epsilon)N \log N$.

The space required for Step 4 is at most $(C/\epsilon)N$.

The list $\mathcal{L}^0$ consists precisely of all the dyadic cubes Q such that the set $S(Q)$ produced from Q by Algorithm 11.1 satisfies $\#(S(Q)) \geq 2$. Moreover, no cube Q appears more than once in the list $\mathcal{L}^0$. Also, the length of the list $\mathcal{L}^0$ is at most that of $\mathcal{L}$, which is at most $\frac{C}{\epsilon}N$.

Thus, our list $\mathcal{L}^0$ satisfies (a), (b), (c), (d).

Remark 11.1. *Note also that the cubes in $\mathcal{L}^0$ appear sorted.*

Let us estimate the work and storage used by the above algorithm. We have seen that Step 1 requires work at most $CN \log N$ in space CN. We have seen also that Steps 2 and 3 require work at most $\exp(C/\epsilon)N \log N$ in space $\frac{C}{\epsilon}N + \exp(C/\epsilon)$. Finally, we have seen that Step 4 requires work (much) less than $\exp(C/\epsilon)\, N \log N$, in space $\frac{C}{\epsilon}N$. Altogether, then, Algorithm 11.2 consumes work at most $\exp(C/\epsilon)N \log N$ in space $\frac{C}{\epsilon}N + \exp(C/\epsilon)$, as claimed in (e) above.

Thus, we have verified (a)...(e), completing our explanation of Algorithm 11.2.

The next algorithm will be used to construct local extending functions as in (13)...(16) in Section 6.

Algorithm 11.3. ("Produce-Fine-Net"): *Given a number $0 < \eta < 1$, a ball $B(x_0, r) \subset \mathbb{R}^n$, and a set $S \subset B(x_0, r)$ such that*

(WS) $|y - y'| \geq \frac{\eta r}{100n}$ *for any two distinct points $y, y' \in S$, we produce a set $S^+ \subset \mathbb{R}^n$, with the following properties:*

(a) $S \subseteq S^+ \subset B(x_0, r)$;

(b) $\mathrm{dist}(x, S^+) < \eta r$ *for any $x \in B(x_0, r)$; and*

(c) $|y - y'| \geq \frac{\eta r}{100n}$ *for any two distinct $y, y' \in S^+$.*

Moreover,

(d) *The work of the algorithm is at most $C\eta^{-2n}$, and the storage needed is at most $C\eta^{-n}$.*

Explanation: First, we define the set S^+; next, we show that S^+ satisfies (a), (b), (c); and finally, we check (d).

To define S^+, we set

$$\tilde{S} = \left[\left(\tfrac{\eta r}{100n}\right)\mathbb{Z}^n\right] \cap B(x_0, r), \tag{18}$$

and then put

$$S^+ = S \cup \{y \in \tilde{S} : \mathrm{dist}(y, S) \geq \tfrac{\eta r}{100n}\}. \tag{19}$$

To prove (a), (b), (c), we first check the following property of $\tilde{S}$.

$$\mathrm{dist}(x, \tilde{S}) < \tfrac{\eta r}{10} \text{ for any } x \in B(x_0, r). \tag{20}$$

In fact, let $x \in B(x_0, r)$. We pick $x' \in B(x_0, (1-\frac{\eta}{50})r)$ such that $|x-x'| \leq \frac{\eta r}{50}$; and then we pick $y \in \frac{\eta r}{100n}\mathbb{Z}^n$ such that $|x'-y| \leq \frac{\eta r}{100}$.
Since $x' \in B(x_0, (1-\frac{\eta}{50})r)$ and $|x'-y| \leq \frac{\eta r}{100}$, we have $y \in B(x_0, (1-\frac{\eta}{100})r) \subset B(x_0, r)$, and thus $y \in \tilde{S}$ by definition (18). Since also

$$|x-y| \leq |x-x'| + |x'-y| \leq \frac{\eta r}{50} + \frac{\eta r}{100} < \frac{\eta r}{10},$$

the proof of (20) is complete.

Now we can check (a), (b), (c) for our set S^+. In fact, (a) is obvious from (18), (19), since $S \subset B(x_0, r)$. To check (b), let $x \in B(x_0, r)$. If $\mathrm{dist}(x, S) < \eta r$, then (b) is obvious. Suppose $\mathrm{dist}(x, S) \geq \eta r$. Thanks to (20), there exists $y \in \tilde{S}$ with $|x-y| \leq \frac{\eta r}{10}$. For this y, we have $\mathrm{dist}(y, S) \geq \mathrm{dist}(x, S) - |x-y| \geq \eta r - \frac{\eta r}{10}$, and therefore $y \in S^+$ according to (19). Thus, $\mathrm{dist}(x, S^+) \leq |x-y| \leq \frac{\eta r}{10} < \eta r$, completing the proof of (b).

To check (c), let $y, y' \in S^+$ be two distinct points.

If $y, y' \in S$, then $|y-y'| \geq \frac{\eta r}{100n}$, by our assumption (WS).

If $y \notin S, y' \notin S$, then by (18), (19), we have $y, y' \in \frac{\eta r}{100n}\mathbb{Z}^n$, $y \neq y'$, and therefore $|y-y'| \geq \frac{\eta r}{100n}$.

If $y \notin S$, $y' \in S$, then since $y \in S^+$, we learn from (19) that $|y-y'| \geq \mathrm{dist}(y, S) \geq \frac{\eta r}{100n}$.

If $y \in S$, $y' \notin S$, then since $y' \in S^+$, we learn from (19) that $|y-y'| \geq \mathrm{dist}(y', S) \geq \frac{\eta r}{100n}$.

Thus, (c) holds in all cases.

Let us estimate the work and storage needed to compute S^+ from (18), (19). Since $S \subset B(x_0, r)$, it follows from our assumption (WS) that

$$\#(S) \leq C\eta^{-n}.$$

Also, a glance at (18) shows that

$$\#(\tilde{S}) \leq C\eta^{-n},$$

and that the work and storage required to compute and store $\tilde{S}$ are at most $C\eta^{-n}$.

To compute S^+ from (19), we compute the distance from each point of $\tilde{S}$ to each point of S. This requires work at most $C\eta^{-2n}$. Also, we want to store the set S^+, which requires space $C \cdot \#(S^+) \leq C \cdot [\#(S) + \#(\tilde{S})]$ (see (19)) $\leq C\eta^{-n}$.

Thus, the work and storage for Algorithm 11.3 are as given in (d). This completes our explanation of Algorithm 11.3.

Remark 11.2. *In a model of computation that includes round-off errors, there may be points* $y \in \frac{\eta r}{100n}\mathbb{Z}^n$ *such that we cannot determine whether* $y \in B(x_0, r)$. *In such delicate cases, we decide not to place* y *in the set* $\tilde{S}$. *Similarly, there may exist points* $y \in \tilde{S}$ *for which we cannot determine whether or not* $\mathrm{dist}(y, S) \geq \frac{\eta r}{100n}$. *We decide to omit such points from the set* S.

With the above modifications, our discussion of Algorithm 11.3 *carries over to a model of computation with small enough roundoff errors.*

Let ϵ, E, Q be as in Algorithm 11.1, and let $Q^{(1)}, \ldots, Q^{(L)}$ be the list of cubes produced by applying Algorithm 11.2 with inputs ϵ, E. If Q does not appear in the list $Q^{(1)}, \ldots, Q^{(L)}$, then we know that the set $S(Q)$ computed by Algorithm 11.1 satisfies $\#(S(Q)) \leq 1$. This will allow us to compute $S(Q)$ with less work than that of Algorithm 11.1, thanks to the following observation:

(21) If $\#(S(Q)) \leq 1$, then $E \cap Q^{**}$ has diameter at most $\hat{C}\epsilon e^{-2/\epsilon}\,\delta_Q$.

This follows at once from the defining property (b) of the set $S(Q)$ in Algorithm 11.1. Exploiting (21), we present the following algorithm.

Algorithm 11.4. *Given* $0 < \epsilon < 1$; *given* $E \subset \mathbb{R}^n$ *with* $\#(E) = N < \infty$; *and given a dyadic cube* $Q \subset \mathbb{R}^n$; *we compute a set* $S_{\mathrm{cheap}}(Q) \subset \mathbb{R}^n$ *with the following property:*
Let $S(Q)$ *be the set computed from* ϵ, E, Q *by Algorithm* 11.1.
If $\#(S(Q)) \leq 1$, *then* $S(Q) = S_{\mathrm{cheap}}(Q)$.

After the one-time work of the BBD Tree Algorithm (see Section 4*), the work to compute* $S_{\mathrm{cheap}}(Q)$ *is at most* $C \log N$, *and the storage required is at most* CN.

Explanation: First, we perform the algorithm Find-Representative from Section 4. With work at most $C \log N$, either we learn that $E \cap Q^{**} = \phi$ (in which case $S(Q) = \phi$, and we may return the set $S_{\text{cheap}}(Q) = \phi$), or else we obtain a point $x_Q \in E \cap Q^{**}$. If $\#(S(Q)) \leq 1$, then

$$E \cap Q^{**} \subset B(x_Q, 2\hat{C}\epsilon e^{-2/\epsilon}\,\delta_Q), \text{ by (21).} \tag{22}$$

Let $Q_\nu (1 \leq \nu \leq \nu_{\max})$ be as in the explanation of Algorithm 11.1. If $\text{dist}(x_Q, Q_\nu) > 2\hat{C}\epsilon e^{2/\epsilon}\,\delta_Q$, then $E \cap Q_\nu = \phi$ by (22). Consequently, if $\#(S(Q)) \leq 1$, then the only cubes Q_ν that may contribute to $\tilde{S}(Q)$ in (2) are those that satisfy

$$\text{dist}(x_Q, Q_\nu) \leq 2\,\hat{C}\epsilon e^{-2/\epsilon}\,\delta_Q. \tag{23}$$

Thanks to (1), there are at most C such cubes, and we can list them with work at most C.

Whether or not $\#(S(Q)) \leq 1$, we now proceed as in the explanation of Algorithm 11.1, except that instead of examining all the cubes Q_ν, we now examine only those satisfying (23). In place of $\tilde{S}(Q)$, $S(Q)$, we thus compute "cheap versions", which we call $\tilde{S}_{\text{cheap}}(Q)$, $S_{\text{cheap}}(Q)$, respectively. If $\#(S(Q)) \leq 1$, then we have $\tilde{S}_{\text{cheap}}(Q) = \tilde{S}(Q)$ and $S_{\text{cheap}}(Q) = S(Q)$.

Whether or not $\#(S(Q)) \leq 1$, the work to examine a single Q_ν as in our explanation of Algorithm 11.1 is at most $C \log N$, and we are now examining at most C cubes Q_ν.

Thus, the work to compute $\tilde{S}_{\text{cheap}}(Q)$ is at most $C \log N$. Moreover, since $\#(\tilde{S}_{\text{cheap}}(Q)) \leq C$, the work to compute $S_{\text{cheap}}(Q)$ from $\tilde{S}_{\text{cheap}}(Q)$ is at most C.

Thus, the total work to compute $S_{\text{cheap}}(Q)$ is at most $C \log N$. If $\#(S(Q)) \leq 1$, then we know that $S_{\text{cheap}}(Q) = S(Q)$. This completes our explanation of Algorithm 11.4.

12. Smoothing Lemmas

In this section, we show that any given $F \in C^m(B(x_0, r))$ can be closely approximated on a slightly smaller ball $B(x_0, r')$ by a function $\tilde{F} \in C^{m+1}(B(x_0, r'))$ with controlled C^{m+1}-norm. Our main result here is Lemma 12.2 below.

Lemma 12.1. *Assume that $\epsilon > 0$ is less than a small enough controlled constant, and let $0 < r \leq 1/\epsilon$. Let $F \in C^m(B(x_0, r))$, with $\| F \|_{C^m(B(x_0, r))} \leq 1$. Let $0 < \bar{\eta} < \min(r, \epsilon^{m+1})$.*

Then there exists $\tilde{F} \in C^{m+1}(B(x_0, r - \bar{\eta}))$, with the following properties:

(a) $\| \tilde{F} \|_{C^m(B(x_0, r-\bar{\eta}))} \leq 1 + C\epsilon$;

(b) $|\partial^\alpha(\tilde{F} - F)(x)| \leq C \cdot (\bar{\eta}/r) \cdot r^{m-|\alpha|}$ *for* $|\alpha| \leq m-1$, $x \in B(x_0, r-\bar{\eta})$;

(c) $|\partial^\alpha \tilde{F}(x)| \leq C \cdot \bar{\eta}^{-1}$ *for* $|\alpha| = m+1$, $x \in B(x_0, r-\bar{\eta})$.

Proof. Let

$$P_0 = J_{x_0}(F) \in \mathcal{P}, \tag{1}$$

and let

$$F_1 = F - P_0. \tag{2}$$

Since $\| F \|_{C^m(B(x_0,r))} \leq 1$, the Bounded Distortion Property and Taylor's theorem yield

$$|\partial^\alpha P_0(x_0)| \leq C \text{ for } |\alpha| \leq m, \text{ and} \tag{3}$$

$$|\partial^\alpha F_1(x)| \leq Cr^{m-|\alpha|} \text{ for } |\alpha| \leq m,\ x \in B(x_0, r). \tag{4}$$

Let $\varphi \in C^{m+1}(\mathbb{R}^n)$ be a function with the following properties.

(5) $\varphi \geq 0$ on $\mathbb{R}^n$; $\operatorname{supp}\varphi \subset B(0, \bar{\eta})$; $\int_{\mathbb{R}^n} \varphi(\tau)d\tau = 1$; $\varphi(\tau) \leq C\bar{\eta}^{-n}$ for $\tau \in \mathbb{R}^n$; and $|\nabla\varphi(\tau)| \leq C \cdot (\bar{\eta})^{-(n+1)}$ for $\tau \in \mathbb{R}^n$.

We define

$$\tilde{F} = P_0 + \varphi * F_1 \text{ on } B(x_0, r-\bar{\eta}), \text{ where } * \text{ denotes convolution.} \tag{6}$$

Note that the right-hand side of (6) is well-defined on $B(x_0, r-\bar{\eta})$, since $\operatorname{supp}\varphi \subset B(0, \bar{\eta})$ and $F_1 \in C^m(B(x_0, r))$. Since $\varphi \in C^{m+1}(\mathbb{R}^n)$, one sees at once from (6) that $\tilde{F} \in C^{m+1}(B(x_0, r-\bar{\eta}))$.

Let us check (a), (b), (c) for $\tilde{F}$. We begin with (b). From (2) and (6), we have $\tilde{F} - F = \varphi * F_1 - F_1$. Hence, for $|\alpha| \leq m-1$ and $x \in B(x_0, r-\bar{\eta})$, we have

$$\begin{aligned}
|\partial^\alpha(\tilde{F} - F)(x)| &= |(\varphi * \partial^\alpha F_1 - \partial^\alpha F_1)(x)| \\
&= \left| \int_{\tau \in B(0,\bar{\eta})} \varphi(\tau)\ [\partial^\alpha F_1(x-\tau) - \partial^\alpha F_1(x)]\ d\tau \right| \\
&\leq \int_{\tau \in B(0,\bar{\eta})} \varphi(\tau) \cdot \left[|\tau| \cdot \sup_{y \in B(x_0,r)} |\nabla\, \partial^\alpha F_1(y)| \right] d\tau \\
&\leq Cr^{m-|\alpha|-1} \int_{\tau \in B(0,\bar{\eta})} \varphi(\tau)|\tau| d\tau \leq C \cdot \left(\frac{\bar{\eta}}{r}\right) \cdot r^{m-|\alpha|},
\end{aligned}$$

thanks to (4), (5). Thus, (b) holds for $\tilde{F}$.

Next, we establish (c). For $|\beta| = m$, $|\gamma| = 1$, $x \in B(x_0, r - \bar{\eta})$, we have $\partial^{\beta+\gamma} P_0 = 0$ since $P_0 \in \mathcal{P}$. Hence, (6) yields $\partial^{\beta+\gamma} \tilde{F}(x) = \partial^\gamma \varphi * \partial^\beta F_1(x)$, and therefore

$$|\partial^{\beta+\gamma} \tilde{F}(x)| \leq \sup_{y \in B(x_0,r)} |\partial^\beta F_1(y)| \cdot \int_{\tau \in \mathbb{R}^n} |\partial^\gamma \varphi(\tau)| \, d\tau \leq C\bar{\eta}^{-1},$$

thanks to (4) and (5). Thus, (c) holds for $\tilde{F}$.

It remains to establish (a). To do so, we first note that

$$\tilde{F} = \varphi * F + (P_0 - \varphi * P_0) \text{ on } B(x_0, r - \bar{\eta}), \tag{7}$$

thanks to (2) and (6). Regarding the term $\varphi * F$ in (7), we note that

$$\varphi * F = \int_{\tau \in B(0,\bar{\eta})} \varphi(\tau) F_\tau \, d\tau \text{ on } B(x_0, r - \bar{\eta}), \tag{8}$$

where $F_\tau \in C^m(B(x_0, r - \bar{\eta}))$ is the translate,

$$F_\tau(x) = F(x - \tau) \text{ for } x \in B(x_0, r - \bar{\eta}). \tag{9}$$

Since $\| F \|_{C^m(B(x_0,r))} \leq 1$, the Approximate Translation-Invariance property and (9) yield

$$\| F_\tau \|_{C^m(B(x_0,r-\bar{\eta}))} \leq \exp(C|\tau|) \leq 1 + C'\bar{\eta} \quad \text{for } \tau \in B(0, \bar{\eta}),$$

and therefore (8) gives

$$\| \varphi * F \|_{C^m(B(x_0,r-\bar{\eta}))} \leq 1 + C'\bar{\eta}, \tag{10}$$

thanks to (5). Regarding the term $P_0 - \varphi * P_0$ in (7), we note that

$$P_0 - \varphi * P_0 \in \mathcal{P},$$

and that

$$|\partial^\alpha (P_0 - \varphi * P_0)(x_0)| \leq C\bar{\eta} \quad \text{for } |\alpha| \leq m,$$

thanks to (3) and (5). Consequently, for $x \in B(x_0, r) \subseteq B(x_0, \epsilon^{-1})$, we have

$$|\partial^\alpha (P_0 - \varphi * P_0)(x)| \leq C\bar{\eta}\epsilon^{-m}, \quad \text{for } |\alpha| \leq m.$$

The Bounded Distortion Property therefore gives $|J_x(P_0 - \varphi * P_0)|_x \leq C'\bar{\eta}\epsilon^{-m}$ for $x \in B(x_0, r)$, which in turn gives the estimate

$$\| P_0 - \varphi * P_0 \|_{C^m(B(x_0,r-\bar{\eta}))} \leq C''\bar{\eta}\epsilon^{-m} \tag{11}$$

From (7), (10), (11), we see that

$$\| \tilde{F} \|_{C^m(B(x_0,r-\bar{\eta}))} \leq 1 + C'\bar{\eta} + C''\bar{\eta}\epsilon^{-m} \leq 1 + C'''\epsilon,$$

since $\bar{\eta} \leq \epsilon^{m+1}$ and $\epsilon \leq 1$. Thus, (a) holds for $\tilde{F}$. The proof of Lemma 12.1 is complete. ∎

Lemma 12.2. *Let $\epsilon > 0$ be less than a small enough controlled constant, and let $B(x_0, r)$ be an open ball with radius*

(†1) $r \leq \epsilon^{-1}$.

Suppose that

(†2) $0 < \eta < \epsilon^2 e^{-1/\epsilon}$.

Let

(†3) $S \subset B(x_0, (1-\eta)r)$,

and assume that

(†4) $|y - y'| > 2\eta e^{1/\epsilon} r$ *for any two distinct points,* $y, y' \in S$.

Let $M \geq 0$, and let

(†5) $F \in C^m(B(x_0, r))$, *with* $\| F \|_{C^m(B(x_0,r))} \leq M$.

Then there exists $F^\# \in C^{m+1}(B(x_0, (1-\eta)r))$, with the following properties.

(A) $\| F^\# \|_{C^m(B(x_0,(1-\eta)r))} \leq (1 + C\epsilon)M$.

(B) $J_y(F^\#) = J_y(F)$ *for all* $y \in S$.

(C) $|\partial^\alpha F^\#(x)| \leq C\eta^{-m} r^{-1} M$ *for* $|\alpha| = m+1$, $x \in B(x_0, (1-\eta)r)$.

Proof. Without loss of generality, we may suppose $M = 1$. We apply Lemma 12.1, with

(12) $\bar{\eta} = \eta^m r$.

Let us check the hypotheses of Lemma 12.1. From our present hypotheses (with $M = 1$), we know that:
$\epsilon > 0$ is less than a small enough controlled constant; $0 < r \leq \epsilon^{-1}$; $F \in C^m(B(x_0, r))$; and $\| F \|_{C^m(B(x_0,r))} \leq 1$.

Also, since $\eta < 1$, we have $\bar{\eta} = \eta^m r < r$; moreover, since $r \leq \epsilon^{-1}$, we have $\bar{\eta} = \eta^m r < \eta^m \epsilon^{-1} \leq \epsilon^{2m-1} e^{-m/\epsilon} \leq \epsilon^{m+1}$, because ϵ is less than a small enough controlled constant, and thanks to (†2). Thus $0 < \bar{\eta} < \min(r, \epsilon^{m+1})$, completing our verification of the hypotheses of Lemma 12.1. Applying that result, we obtain a function

(13) $\tilde{F} \in C^{m+1}(B(x_0, (1-\eta^m)r))$,
with the following properties.

(14) $\| \tilde{F} \|_{C^m(B(x_0,(1-\eta^m)r))} \leq 1 + C\epsilon$.

(15) $|\partial^\alpha(\tilde{F} - F)(x)| \leq C\eta^m r^{m-|\alpha|}$ for $|\alpha| \leq m-1$, $x \in B(x_0, (1-\eta^m)r)$.

(16) $|\partial^\alpha \tilde{F}(x)| \leq C\eta^{-m} r^{-1}$ for $|\alpha| = m+1$, $x \in B(x_0, (1-\eta^m)r)$.

The function $\tilde{F}$ needn't satisfy (B). We prepare to correct it, using Lemma GPU. We set

(17) $\Omega = \tilde{U} = B(x_0, (1-\eta^m)r)$.

For each $y \in S$, we define

(18) $P^y = J_y(F) \in \mathcal{P}$, and

(19) $U^y = B(y, \eta e^{1/\epsilon} r)$.

(20) The $U^y (y \in S)$ are pairwise disjoint, thanks to (†4).

We estimate the C^m norm of P^y on U^y, and we compare $\tilde{F}$ with P^y on $\tilde{U} \cap U^y$. To do so, we first note that

(21) $|P^y|_y \leq 1$ for $y \in S$,

thanks to (18) and (†5) (with $M = 1$).

We have also

(22) $\eta e^{1/\epsilon} r < (\epsilon^2 e^{-1/\epsilon}) \cdot e^{1/\epsilon} \cdot (\epsilon^{-1}) = \epsilon < 1$, by (†1) and (†2).

From (19), (21), (22), and Lemma 1 in Section 3, we obtain the estimate

(23) $|P^y|_x \leq 1 + C\epsilon$ for $x \in U^y$, $y \in S$.

Next, fix $y \in S$ and $x \in U^y \cap \tilde{U}$ (see (17), (19)). From (14), the Bounded Distortion Property, and Taylor's theorem, we have

(24) $|\partial^\alpha(\tilde{F} - J_y(\tilde{F}))(x)| \leq C|x-y|^{m-|\alpha|}$ for $|\alpha| \leq m$.

(In the degenerate case $x = y$, $|\alpha| = m$, we define the right-hand side of (24) to be zero.)

We will check that

(25) $|\partial^\alpha(\tilde{F} - F)(y)| \leq C \cdot (\eta r)^{m-|\alpha|}$ for $|\alpha| \leq m$.

In fact, for $|\alpha| \leq m-1$, estimate (25) is immediate from (15), since $\eta < 1$. For $|\alpha| = m$, estimate (25) follows from (†5) (with $M = 1$) and (14), thanks to the Bounded Distortion Property (and the fact that $y \in B(x_0, (1-\eta^m)r)$ by (†3)). Thus, (25) holds in all cases.

From (25), we deduce the weaker estimate

$$|\partial^\alpha(J_y(\tilde{F}) - J_y(F))(y)| \leq C \cdot [\eta r + |x-y|]^{m-|\alpha|} \quad \text{for } |\alpha| \leq m,$$

which in turn yields

(26) $|\partial^\alpha(J_y(\tilde{F}) - J_y(F))(x)| \leq C \cdot [\eta r + |x-y|]^{m-|\alpha|}$ for $|\alpha| \leq m$, since $J_y(\tilde{F}) - J_y(F) \in \mathcal{P}$.

Combining (24) and (26), and recalling (18), we find that

(27) $|\partial^\alpha(\tilde{F} - P^y)(x)| \leq C \cdot [\eta r + |x-y|]^{m-|\alpha|}$ for $|\alpha| \leq m$, $x \in U^y \cap \tilde{U}$, $y \in S$.

Estimates (23) and (27) are our main results on the P^y and $\tilde{F} - P^y$.

Next, given $y \in S$, we define χ^y to be the function θ_0, computed by applying Algorithm 10.6, with $y_0 = y$, and with $\delta = \eta r$. We are not concerned here with computing the function χ^y, but we recall the properties (a)...(d) in Algorithm 10.6. Thus,

(28) $\chi^y \in C^{m+1}(\mathbb{R}^n)$,

(29) $0 \leq \chi^y \leq 1$ on $\mathbb{R}^n$,

(30) $\chi^y(x) = 1$ for $|x-y| \leq e^{1/(16\epsilon)}\eta r$,

(31) $\operatorname{supp}\chi^y \subset B(y, e^{1/(8\epsilon)}\eta r) \subset U^y$ (see (19)), and

(32) $|\partial^\alpha \chi^y(x)| \leq C\epsilon|x-y|^{-|\alpha|}$ for $0 < |\alpha| \leq m+1$, $x \in \mathbb{R}^n \setminus \{y\}$.

Properties (28)...(32) hold for each $y \in S$.
From (30) and (32), we obtain the estimate

(33) $|\partial^\alpha \chi^y(x)| \leq C\epsilon \cdot [\eta r + |x-y|]^{-|\alpha|}$ for $0 < |\alpha| \leq m+1$, $x \in \mathbb{R}^n$, $y \in S$.

In addition to the functions $\chi^y (y \in S)$, we define a function $\tilde{\chi}$, by setting

(34) $\tilde{\chi} = 1 - \sum_{y \in S} \chi^y$ on $\mathbb{R}^n$.

(Recall that S is finite; see (†3) and (†4).)

Thanks to (20) and (28)...(33), the function $\tilde{\chi}$ has the following properties.

(35) $\tilde{\chi} \in C^{m+1}(\mathbb{R}^n)$.

(36) $0 \leq \tilde{\chi} \leq 1$ on $\mathbb{R}^n$.

(37) $\tilde{\chi}(x) = 0$ for $|x-y| \leq e^{1/(16\epsilon)}\eta r$, $y \in S$.

(38) $\tilde{\chi} = 1$ in a neighborhood of x, for $x \notin \bigcup_{y \in S} U^y$.

(39) $|\partial^\alpha \tilde{\chi}(x)| \leq C\epsilon \cdot [\eta r + |x-y|]^{-|\alpha|}$ for $0 < |\alpha| \leq m+1$, $x \in U^y$, $y \in S$.

We now define

(40) $\delta(x) = \eta r + \operatorname{dist}(x, S) > 0$ for $x \in \mathbb{R}^n$.

We now check the hypotheses of Lemma GPU from Section 5, for the following data:

- The open cover $\{U^y \text{ (all } y \in S),\ \tilde{U}\}$ of the open set Ω, as in (17), (19);
- The function $\delta(x) > 0$, as in (40);
- The functions P^y (all $y \in S$) and $\tilde{F}$, as in (18), (13)...(16);
- The functions χ^y (all $y \in S$) and $\tilde{\chi}$, as in (28)...(34);
- The constants $A_0 = A_1 = A_2 = C$ for a large enough controlled constant C.
- The constant called M in Lemma GPU will be taken here to be $1 + C\epsilon$ for a large enough controlled constant C.

The verification of the hypotheses of Lemma GPU for the above data proceeds as follows.

First of all, $\{U^y(\text{all } y \in S),\ \tilde{U}\}$ is an open cover of an open set $\Omega \subset \mathbb{R}^n$. Also, $\delta(x) > 0$ for all $x \in \Omega$, thanks to (40). For each $y \in S$, we have $P^y \in C^m(U^y)$, since $P^y \in \mathcal{P}$. Moreover, $\tilde{F} \in C^m(\tilde{U})$, as we see from (13) and (17). We have $\chi^y \in C^m(\Omega)$ for $y \in S$, and $\tilde{\chi} \in C^m(\Omega)$; see (28) and (35). We have $\epsilon, A_0, A_1, A_2 > 0$ and $M \geq 0$.

We now check hypotheses (GPU1...7).

(GPU1) asserts here that any given point of Ω can belong to at most C of the sets $\mathrm{supp}_\Omega \chi^y$ ($y \in S$), and $\mathrm{supp}_\Omega \tilde{\chi}$. That assertion holds, thanks to (20) and (31); note that $\mathrm{supp}_\Omega \varphi \subseteq \mathrm{supp}\varphi$ for any function φ on $\mathbb{R}^n$.

(GPU2) asserts here that $\sum_{y \in S} \chi^y + \tilde{\chi} = 1$ on Ω, which is immediate from (34).

(GPU3) asserts here that $\chi^y \geq 0$ on Ω for each $y \in S$, and that $\tilde{\chi} \geq 0$ on Ω. These assertions are immediate from (29), (36).

(GPU4) asserts here that $\mathrm{supp}_\Omega \chi^y \subset U^y$ for $y \in S$, and that

(41) $\quad \mathrm{supp}_\Omega \tilde{\chi} \subset \tilde{U}$.

The assertion regarding the χ^y is immediate from (31), since $\mathrm{supp}_\Omega \chi^y \subseteq \mathrm{supp}\chi^y$. Assertion (41) holds trivially, since $\tilde{U} = \Omega$ (see (17)), and $\mathrm{supp}_\Omega \tilde{\chi}$ is defined to be a subset of Ω.

(GPU5) asserts here that

(42) $\quad |\partial^\alpha \chi^y(x)| \leq C\epsilon \cdot [\eta r + \mathrm{dist}(x, S)]^{-|\alpha|}$ for $0 < |\alpha| \leq m$, $x \in \Omega$, $y \in S$, and that

(43) $\quad |\partial^\alpha \tilde{\chi}(x)| \leq C\epsilon \cdot [\eta r + \mathrm{dist}(x, S)]^{-|\alpha|}$ for $0 < |\alpha| \leq m$, $x \in \Omega$.

We establish (42) and (43) by cases. If $x \notin \cup_{y\in S} U^y$, then the left-hand sides of (42), (43) are both zero, thanks to (31) and (38). Thus, we may assume that $x \in U^{\bar y}$ for some $\bar y \in S$.

In this case, we have $\mathsf{dist}(x, S) = |x-\bar y|$, by (19), (20). Consequently, (43) now follows from (39), and (42) for $y = \bar y$ follows from (33).

For $y \in S \setminus \{\bar y\}$, (42) holds thanks to (20) and (31), since here $x \in U^{\bar y}$. Thus, (42), (43) hold in all cases, proving (GPU5).

(GPU6) asserts here that

(44) $|J_x(P^y)|_x \le 1 + C\epsilon$ for $x \in \mathsf{supp}_\Omega \chi^y$, $y \in S$, and that

(45) $|J_x(\tilde F)|_x \le 1 + C\epsilon$ for $x \in \mathsf{supp}_\Omega \tilde\chi$.

Estimate (45) is immediate from (14) and (17). To check (44), we recall that $P^y \in \mathcal{P}$, hence $J_x(P^y) = P^y$ for $y \in S$, $x \in U^y$. Consequently, (44) follows from (23) and (31), since $\mathsf{supp}_\Omega \chi^y \subseteq \mathsf{supp}\chi^y$. Thus, (GPU6) holds.

(GPU7) here asserts that

(46) $|\partial^\alpha(\tilde F - P^y)(x)| \le C[\eta r + \mathsf{dist}(x, S)]^{m-|\alpha|}$ for $|\alpha| \le m-1$, $x \in \mathsf{supp}_\Omega \chi^y \cap \mathsf{supp}_\Omega \tilde\chi$, $y \in S$.

(See (20), (31), (40).)

To check (46), we recall that $x \in \mathsf{supp}_\Omega \chi^y \cap \mathsf{supp}_\Omega \tilde\chi$, $y \in S$ imply $x \in U^y \cap \tilde U$ by (17), (31); hence $\mathsf{dist}(x, S) = |x - y|$ thanks to (20). Thus, (46) follows from (27), proving (GPU7).

This completes the verification of the hypotheses of Lemma GPU for the above data. Applying that lemma, we learn the following.

Define a function $F^\#$ on $B(x_0, (1-\eta^m)r)$, by setting

(47) $$F^\#(x) = \sum_{y\in S} \chi^y(x)\cdot P^y(x) + \tilde\chi(x)\cdot \tilde F(x) \text{ for } x \in B(x_0, (1-\eta^m)r).$$

Then

(48) $F^\# \in C^m(B(x_0, (1-\eta^m)r))$, and $\| F^\# \|_{C^m(B(x_0,(1-\eta^m)r))} \le 1 + C\epsilon$.

We investigate $J_y(F^\#)$ for $y \in S$. From (18), (20), (30), (31), we learn that, for each $y \in S$, we have

(49) $J_y(P^y) = P^y = J_y(F)$, $J_y(\chi^y) = 1$, and $J_y(\chi^{\bar y}) = 0$ for $\bar y \in S \setminus \{y\}$.

Also, (37) yields

(50) $J_y(\tilde\chi) = 0$ for $y \in S$.

Substituting (49), (50) into (47), we learn that

(51) $J_y(F^\#) = J_y(F)$ for $y \in S$.

Next, we study the $(m+1)^{\underline{\text{rst}}}$ derivatives of $F^\#$. We recall that each P^ν belongs to $\mathcal{P}$, and that $\chi^\nu(\nu \in S)$, $\tilde{\chi}$, and $\tilde{F}$ all belong to $C^{m+1}(B(x_0,(1-\eta^m)r))$. (See (13), (18), (28), (35).)

Since S is finite (see (†3), (†4)), it follows that

(52) $\quad F^\# \in C^{m+1}(B(x_0,(1-\eta^m)r))$ (see (47)).

Let $|\alpha| = m+1$, and let $x \in \Omega = B(x_0,(1-\eta^m)r)$.
We estimate $|\partial^\alpha F^\#(x)|$. We proceed by cases.

Case 1: Suppose $x \in \Omega \setminus \bigcup_{\nu\in S} U^\nu$. Then (31), (38), (47) show that $F^\# = \tilde{F}$ in a neighborhood of x. Therefore, (16) gives

(53) $\quad |\partial^\alpha F^\#(x)| \le C\eta^{-m} r^{-1}$ in Case 1.

Case 2: Suppose $x \in U^{\bar\nu}$ for some $\bar\nu \in S$. From (20), (31), (34), we see that $\chi^\nu = 0$ on $U^{\bar\nu}$ for $\nu \in S \setminus \{\bar\nu\}$, and $\tilde{\chi} = 1-\chi^{\bar\nu}$ on $U^{\bar\nu}$. Hence, (47) gives

(54) $\quad F^\# = \chi^{\bar\nu}\cdot P^{\bar\nu} + (1-\chi^{\bar\nu})\cdot\tilde{F}$ on $U^{\bar\nu}$.

Since $|\alpha| = m+1$ and $P^{\bar\nu} \in \mathcal{P}$, we have $\partial^\alpha P^{\bar\nu} = 0$. Hence, from (54), we obtain

$$(55)\quad \partial^\alpha F^\#(x) = (1-\chi^{\bar\nu}(x))\cdot\partial^\alpha\tilde{F}(x) - \sum_{\substack{\beta+\gamma=\alpha\\ |\beta|\neq 0}} \frac{\alpha!}{\beta!\gamma!}\,\partial^\beta\chi^{\bar\nu}(x)\cdot\partial^\gamma(\tilde{F}-P^{\bar\nu})(x).$$

We estimate the terms on the right in (55). Recalling (16) and (29), we see that

(56) $\quad |(1-\chi^{\bar\nu}(x))\cdot\partial^\alpha\tilde{F}(x)| \le C\eta^{-m}r^{-1}$.

Also, for $\beta+\gamma=\alpha$, $|\beta|\neq 0$, estimates (27) and (33) yield

$$(57)\quad \begin{aligned}|\partial^\beta\chi^{\bar\nu}(x)|\cdot|\partial^\gamma(\tilde{F}-P^\nu)(x)| &\le C\epsilon\cdot[\eta r+|x-y|]^{-|\beta|}\cdot C\cdot[\eta r+|x-y|]^{m-|\gamma|}\\ &= C'\epsilon\cdot[\eta r+|x-y|]^{-1} \le C\eta^{-m}r^{-1},\end{aligned}$$

since $|\beta|+|\gamma| = |\alpha| = m+1$, and $\epsilon,\eta < 1$.

Putting (56) and (57) into (55), we learn that

(58) $\quad |\partial^\alpha F^\#(x)| \le C\eta^{-m}r^{-1}$ in Case 2.

In view of (53) and (58), we now have

(59) $\quad |\partial^\alpha F^\#(x)| \le C\eta^{-m}r^{-1}$ for $|\alpha| = m+1$, $x \in B(x_0,(1-\eta^m)r)$.

The conclusions of Lemma 12.2 are immediate from (52), (48), (51) and (59), since here $M = 1$. The proof of Lemma 12.2 is complete. ∎

13. Extending a Whitney Field from a Fine Net

The next several sections give algorithms that will allow us to compute functions F^ℓ_ν as in (13)...(16) in Section 6. The algorithm of this section is as follows.

Algorithm 13.1. *Suppose we are given the following data.*

- *A real number* $\epsilon > 0$*, assumed to be less than a small enough controlled constant.*
- *A real number* $0 < \eta < \epsilon^2$*.*
- *A real number* A*, assumed to be greater than a large enough controlled constant.*
- *An open ball* $B(x_0, r)$*, with* $r \leq \epsilon^{-1}$*.*
- *A Whitney field* $\vec{P} = (P^y)_{y \in S^+}$*, where* S^+ *is assumed to satisfy:*

(†0) $S^+ \subset B(x_0, r)$*;*

(†1) $|y - y'| \geq A^{-1}\eta r$ *for any two distinct points* $y, y' \in S^+$*; and*

(†2) $\mathrm{dist}(x, S^+) \leq \eta r$ *for all* $x \in B(x_0, r)$*.*

Given the above data, we compute a function $F \in C^m(\mathbb{R}^n)$*, with the following properties.*

(A) F *agrees with* $\vec{P}$*.*

(B) *Suppose* $M \geq 0$ *is a real number, and suppose that*

(†3) $|P^y|_y \leq M$ *for all* $y \in S^+$*, and*

(†4) $|\partial^\alpha(P^y - P^{y'})(y)| \leq \epsilon M \cdot (\eta r)^{-1} \cdot |y - y'|^{m+1-|\alpha|}$ *for* $|\alpha| \leq m$*,* $y, y' \in S^+$*.*

Then

$$\| F \|_{C^m(B(x_0,r))} \leq (1 + CA^m\epsilon) \cdot M\,.$$

The computation of F *uses one-time work at most* $C \cdot (A\eta^{-1})^{2n}$*, storage at most* $C \cdot (A\eta^{-1})^n$*, and query work at most* C*.*

Explanation: Our plan is as follows. We start by discussing the geometry of the ball $B(x_0, r)$ and the set S^+. Next, we recall a relevant partition of unity from Section 10. We then define a function F on $\mathbb{R}^n$, and prove that it satisfies (A) and (B). Finally, we show how to compute F, and we estimate the work and storage of the computation.

We begin with some trivial geometry. It is convenient to regard $B(x_0, r)$ as a subset of a cube $\bar{Q}$, which we will partition into dyadic cubes. To do so, we proceed as follows. First, we fix an integer s, satisfying

(1) $\frac{1}{100} A^{-1}\eta r \leq \sqrt{n} \cdot 2^s \leq \frac{1}{10} A^{-1}\eta r$,

and then fix an integer $L > 0$, satisfying

(2) $100r \leq 2^s \cdot L \leq 1000r$.

Note that

(3) $cA\eta^{-1} \leq L \leq CA\eta^{-1}$.

Next, let $(x_1^0, \ldots, x_n^0)$ be the coordinates of x_0. For each $j (1 \leq j \leq n)$, we produce an integer m_j, such that

(4) $[x_j^0 - 2r, x_j^0 + 2r) \subset [2^s \cdot m_j, 2^s \cdot (m_j + L))$ for each j.

We can find such m_j, thanks to (2). From (4), we have

$$B(x_0, r) \subset \prod_{j=1}^{n} [x_j^0 - r, x_j^0 + r) \subset \prod_{j=1}^{n} [2^s \cdot m_j, 2^s \cdot (m_j + L)).$$

Thus, setting

(5) $\bar{Q} = \prod_{j=1}^{n} [2^s \cdot m_j, 2^s \cdot (m_j + L))$,

we note that $\bar{Q}$ is a cube, and that

(6) $B(x_0, r) \subset \bar{Q}$, and

(7) $\delta_{\bar{Q}} = 2^s \cdot L \leq 1000r$,

thanks to (2).
From Section 10, we recall the dyadic cubes

(8) $Q_\nu^{\langle s \rangle} = \prod_{j=1}^{n} [2^s \cdot \nu_j, 2^s \cdot (\nu_j + 1))$ for $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$.

Comparing (5) with (8), we see that

(9) $\bar{Q}$ is partitioned into dyadic cubes $Q_\nu^{\langle s \rangle}$ $(\nu \in \mathcal{G})$, where

(10) $\mathcal{G} = \{\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n : m_j \leq \nu_j < m_j + L \text{ for } j = 1, \ldots, n\}$.

We check the following property of the cubes $Q_\nu^{\langle s \rangle}$.

(11) Let $\nu \in \mathbb{Z}^n$. If $(Q_\nu^{\langle s \rangle})^* \cap B(x_0, r) \neq \phi$, then $\nu \in \mathcal{G}$.

In fact, let $\nu = (\nu_1, \dots, \nu_n) \in \mathbb{Z}^n$, and suppose

$$(x_1, \dots, x_n) \in (Q_\nu^{\langle s \rangle})^* \cap B(x_0, r).$$

From (8), we have

$$(Q_\nu^{\langle s \rangle})^* = \prod_{j=1}^{n} [2^s \cdot (\nu_j - 1), 2^s \cdot (\nu_j + 2)),$$

and therefore $x_j \in [2^s \cdot (\nu_j - 1),\ 2^s \cdot (\nu_j + 2)) \cap [x_j^0 - r, x_j^0 + r)$ for each j.

In particular, $2^s \cdot \nu_j$ lies within distance $2 \cdot 2^s$ of the interval $[x_j^0 - r, x_j^0 + r)$. Since $2 \cdot 2^s < r$ by (1), it follows that $2^s \cdot \nu_j \in [x_j^0 - 2r\,,\ x_j^0 + 2r)$ for each j, and consequently $2^s \cdot \nu_j \in [2^s \cdot m_j\,,\ 2^s(m_j + L))$, thanks to (4). Thus, $m_j \le \nu_j < m_j + L$ for each j, i.e., $\nu = (\nu_1, \dots, \nu_n) \in \mathcal{G}$, completing the proof of (11).

Let us now bring in the set S^+. For each $\nu \in \mathcal{G}$, let

(12) $\quad x_\nu =$ center of $Q_\nu^{\langle s \rangle}$,

and let

(13) $\quad y_\nu =$ a point of S^+ as close as possible to x_ν.

Note that (13) makes sense, since S^+ is finite. In fact, our assumptions (†0), (†1) show that

(14) $\quad \#(S^+) \le C \cdot (A\eta^{-1})^n$.

We next establish the following properties of the points y_ν.

(15) Let $x \in B(x_0, r) \cap (Q_\nu^{\langle s \rangle})^*$. Then $\nu \in \mathcal{G}$ and $|x - y_\nu| \le C\eta r$.

(16) Let $\nu \in \mathcal{G}$. If $(Q_\nu^{\langle s \rangle})^* \cap S^+ \neq \phi$, then $(Q_\nu^{\langle s \rangle})^* \cap S^+ = \{y_\nu\}$.

To see (15), let $x \in B(x_0, r) \cap (Q_\nu^{\langle s \rangle})^*$. Then $\nu \in \mathcal{G}$, by (11). Since $x, x_\nu \in (Q_\nu^{\langle s \rangle})^*$, and since $(Q_\nu^{\langle s \rangle})^*$ has diameter $3\sqrt{n} \cdot 2^s \le \frac{1}{3} A^{-1} \eta r$ (see (1)), we have $|x - x_\nu| \le \frac{1}{3} A^{-1} \eta r \le \eta r$. Also, by assumption (†2), there exists $y \in S^+$ with $|x - y| \le \eta r$. Moreover, by definition (13), we have $|x_\nu - y_\nu| \le |x_\nu - y|$. In view of the above remarks, we have

$$\begin{aligned} |x - y_\nu| &\le |x - x_\nu| + |x_\nu - y_\nu| \le |x - x_\nu| + |x_\nu - y| \\ &\le |x - x_\nu| + |x - x_\nu| + |x - y| \le 3\eta r\,, \end{aligned}$$

proving (15).

To see (16), let $y \in (Q_\nu^{\langle s \rangle})^* \cap S^+$. Since x_ν and y belong to $(Q_\nu^{\langle s \rangle})^*$, a cube with diameter $\le \frac{1}{3} A^{-1} \eta r$, we have $|x_\nu - y| \le \frac{1}{3} A^{-1} \eta r$. Hence, by definition (13), we have also $|x_\nu - y_\nu| \le \frac{1}{3} A^{-1} \eta r$. Consequently, $|y - y_\nu| \le \frac{2}{3} A^{-1} \eta r$. Since also $y, y_\nu \in S^+$, it now follows from assumption (†1) that $y = y_\nu$.

This proves that y_ν is the one and only point belonging to $(Q_\nu^{\langle s\rangle})^* \cap S^+$, which is our desired conclusion (16).

Next, from Section 10, we recall the partition of unity

$$1 = \sum_{\nu\in\mathbb{Z}^n} \theta_\nu^{\langle s\rangle} \text{ on } \mathbb{R}^n, \tag{17}$$

where

$$\theta_\nu^{\langle s\rangle} \in C^m(\mathbb{R}^n) \text{ for } \nu \in \mathbb{Z}^n, \tag{18}$$

$$\operatorname{supp} \theta_\nu^{\langle s\rangle} \subset (Q_\nu^{\langle s\rangle})^* \text{ for each } \nu \in \mathbb{Z}^n, \tag{19}$$

and

$$|\partial^\alpha \theta_\nu^{\langle s\rangle}(x)| \le C \cdot 2^{-s|\alpha|} \text{ for } |\alpha| \le m,\ x \in \mathbb{R}^n,\ \nu \in \mathbb{Z}^n. \tag{20}$$

Thanks to (1), our estimate (20) is equivalent to

$$|\partial^\alpha \theta_\nu^{\langle s\rangle}(x)| \le C \cdot (A^{-1}\eta r)^{-|\alpha|} \le CA^m \cdot (\eta r)^{-|\alpha|} \text{ for } |\alpha| \le m,\ x \in \mathbb{R}^n,\ \nu \in \mathbb{Z}^n. \tag{21}$$

We want to restrict the sum in (17) to $\nu \in \mathcal{G}$. In view of (11) and (19), we have $\nu \in \mathcal{G}$ for every $\nu \in \mathbb{Z}^n$ such that $\operatorname{supp}\theta_\nu^{\langle s\rangle} \cap B(x_0, r) \neq \phi$. Hence, (17) yields

$$\sum_{\nu\in\mathcal{G}} \theta_\nu^{\langle s\rangle} = 1 \text{ on } B(x_0, r). \tag{22}$$

We are now ready to define our function F. We set

$$F = \sum_{\nu\in\mathcal{G}} \theta_\nu^{\langle s\rangle} \cdot P^{y_\nu} \text{ on } \mathbb{R}^n. \tag{23}$$

Note that

$$F \in C^m(\mathbb{R}^n), \tag{24}$$

and F is of compact support. (See (18), (19), and recall that $\mathcal{G}$ is finite and each P^{y_ν} is a polynomial.)

Next, we show that F satisfies (A) and (B). We first establish (A):

Fix $y \in S^+$. If $\nu \in \mathcal{G}$ and $y \in \operatorname{supp} \theta_\nu^{\langle s\rangle}$, then $y \in S^+ \cap (Q_\nu^{\langle s\rangle})^*$ by (19), and therefore $y_\nu = y$, by (16). Consequently, (23) gives

$$J_y(F) = \sum_{\nu\in\mathcal{G}} J_y(\theta_\nu^{\langle s\rangle} \cdot P^y) = P^y,$$

thanks to (22) and (†0).

Thus, F agrees with $\vec{P}$, proving (A).

We pass to (B). Suppose $M \geq 0$ is a real number, and suppose the $P^y (y \in S^+)$ satisfy (†3) and (†4). Fix $x \in B(x_0, r)$, and then fix $\bar{\nu} \in \mathcal{G}$ such that $x \in (Q_{\bar{\nu}}^{\langle s \rangle})^*$. (Such a $\bar{\nu}$ exists, by (19), (22).) From (22) and (23), we have

$$F = P^{y_{\bar{\nu}}} + \sum_{\nu \in \mathcal{G}} \theta_\nu^{\langle s \rangle} \cdot (P^{y_\nu} - P^{y_{\bar{\nu}}}) \text{ on } B(x_0, r). \tag{25}$$

We estimate the terms on the right in (25).

To do so, we note first that any $\nu \in \mathcal{G}$ such that $x \in (Q_\nu^{\langle s \rangle})^*$ satisfies $|x - y_\nu| \leq C\eta r$, thanks to (15). In particular,

$$|x - y_{\bar{\nu}}| \leq C\eta r, \tag{26}$$

and

$$|x - y_\nu| \leq C\eta r \text{ for any } \nu \in \mathcal{G} \text{ such that } x \in \operatorname{supp} \theta_\nu^{\langle s \rangle}. \text{ (See (19).)} \tag{27}$$

From (26), (27), we obtain at once $|y_\nu - y_{\bar{\nu}}| \leq C\eta r$ for any $\nu \in \mathcal{G}$ such that $x \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$. Therefore, (†4) yields

$$|\partial^\alpha (P^{y_\nu} - P^{y_{\bar{\nu}}})(y_{\bar{\nu}})| \leq C\epsilon M \cdot (\eta r)^{m - |\alpha|}$$

for $|\alpha| \leq m$, $\nu \in \mathcal{G}$ such that $x \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$.

Together with (26), this gives

$$|\partial^\alpha (P^{y_\nu} - P^{y_{\bar{\nu}}})(x)| \leq C'\epsilon M \cdot (\eta r)^{m - |\alpha|}$$

for $|\alpha| \leq m$, $\nu \in \mathcal{G}$ such that $x \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$.

Recalling (21), we deduce that

$$|\partial^\alpha \{\theta_\nu^{\langle s \rangle} \cdot (P^{y_\nu} - P^{y_{\bar{\nu}}})\}(x)| \leq A^m C\epsilon M \cdot (\eta r)^{m - |\alpha|} \text{ for } |\alpha| \leq m, \nu \in \mathcal{G} \text{ s.t. } x \in \operatorname{supp} \theta_\nu^{\langle s \rangle}. \tag{28}$$

Moreover, there are at most C distinct $\nu \in \mathcal{G}$ such that $x \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$, as we see from (19). Consequently, (28) yields

$$\Big| \partial^\alpha \Big\{ \sum_{\nu \in \mathcal{G}} \theta_\nu^{\langle s \rangle} \cdot (P^{y_\nu} - P^{y_{\bar{\nu}}}) \Big\}(x) \Big| \leq CA^m \epsilon M \cdot (\eta r)^{m - |\alpha|} \text{ for } |\alpha| \leq m. \tag{29}$$

We are assuming here that $\eta \leq \epsilon^2$, $r \leq \epsilon^{-1}$, and therefore

$$\eta r < \epsilon. \tag{30}$$

From (29), (30), and the Bounded Distortion Property, we conclude that

$$\Big| J_x \Big(\sum_{\nu \in \mathcal{G}} \theta_\nu^{\langle s \rangle} \cdot (P^{y_\nu} - P^{y_{\bar{\nu}}}) \Big) \Big|_x \leq CA^m \epsilon M. \tag{31}$$

Thus, we have estimated the sum in (25). Regarding the term $P^{y_{\bar\nu}}$ in (25), we recall from (†3) that

(32) $\quad |P^{y_{\bar\nu}}|_{y_{\bar\nu}} \leq M.$

Also, from (26), (30), we have $|x - y_{\bar\nu}| \leq C\eta r < C\epsilon < 1$. Hence, (32) and Lemma 1 in Section 3 yield the estimate $|P^{y_{\bar\nu}}|_x \leq (1 + C\epsilon)M$.

Substituting this estimate and (31) into (25), and recalling that $A > 1$, we obtain the estimate $|J_x(F)|_x \leq (1 + CA^m\epsilon)M$.

Since x is an arbitrary point in $B(x_0, r)$, it follows that

$$\| F \|_{C^m(B(x_0,r))} \leq (1 + CA^m\epsilon)M.$$

This completes the proof of (B).

It remains to compute the function F in (23), and to estimate the work and storage needed for the computation. We begin with the one-time work. We compute the integers s, L, $m_1, \ldots, m_n$ in (1)...(4), with work at most C. Note that the set $\mathcal{G}$ in (10) is then an n-dimensional box in an integer lattice; hence, any quantity indexed by $\mathcal{G}$ may be stored as an n-dimensional array, consisting of L^n entries. Recall from (3) that $L^n \leq C \cdot (A\eta^{-1})^n$.

Next, for each $\nu \in \mathcal{G}$, we compute crudely the point y_ν in (13), by examining each $y \in S^+$. We then store the polynomial P^{y_ν} in an n-dimensional array indexed by $\nu \in \mathcal{G}$. The work to compute a single P^{y_ν} is at most $C \cdot \#(S^+) \leq C' \cdot (A\eta^{-1})^n$; see (14).

The memory consumed in storing the P^{y_ν} (all $\nu \in \mathcal{G}$) is at most $C \cdot (A\eta^{-1})^n$. This completes the one-time work. Note that the total one-time work is at most $C' \cdot (A\eta^{-1})^n \cdot \#(\mathcal{G}) \leq C'' \cdot (A\eta^{-1})^{2n}$.

We pass to the query algorithm, which computes $J_{\underline{x}}(F)$ for a given query point $\underline{x} \in \mathbb{R}^n$. We assume we have done the above one-time work.

Let $\underline{x} \in \mathbb{R}^n$ be given, and let $\mathcal{G}(\underline{x}) = \{\nu \in \mathcal{G} : \underline{x} \in (Q_\nu^{\langle s \rangle})^*\}$. Note that $\#(\mathcal{G}(\underline{x})) \leq C$, and that $\mathcal{G}(\underline{x})$ can be computed with work at most C. From (19) and (23), we have

(33) $\quad J_{\underline{x}}(F) = \sum_{\nu \in \mathcal{G}(\underline{x})} J_{\underline{x}}(\theta_\nu^{\langle s \rangle}) \odot_{\underline{x}} P^{y_\nu}.$

For each $\nu \in \mathcal{G}(\underline{x})$, we compute $J_{\underline{x}}(\theta_\nu^{\langle s \rangle})$ by Algorithm 10.3, look up P^{y_ν} from the array created by our one-time work, and then compute the product $J_{\underline{x}}(\theta_\nu^{\langle s \rangle}) \odot_{\underline{x}} P^{y_\nu}$.

Summing on $\nu \in \mathcal{G}(\underline{x})$, we obtain $J_{\underline{x}}(F)$ from formula (33). Since Algorithm 10.3 takes work at most C, and since $\#(\mathcal{G}(\underline{x})) \leq C$, it follows that $J_{\underline{x}}(F)$ is computed using work at most C.

Thus, F is computed using one-time work at most $C \cdot (A\eta^{-1})^{2n}$ and storage at most $C \cdot (A\eta^{-1})^n$; and the query work is at most C.

This completes our explanation of Algorithm 13.1.

14. Local Extension of a Whitney Field from a Testing Set

In this section, we continue preparing to compute functions F^ℓ_ν as in (13)...(16) of Section 6. We will use Algorithm 3.1 ("Find-Unit-Ball"), and the Special Ellipsoid Algorithm for linear programming.

Algorithm 14.1. *Given a real number $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube Q, whose side-length δ_Q is assumed to satisfy*

(1) $\delta_Q \leq \tilde{c}\epsilon^{-1}$ *for a small enough controlled constant $\tilde{c}$;*

and given a Whitney field $\vec{P} = (P^y)_{y\in S}$, where the set S is assumed to satisfy

(2) $S \subset Q^{**}$, *and*

(3) $|y - y'| > e^{-3/\epsilon}\,\delta_Q$ *for any two distinct points $y, y' \in S$;*

we compute a ball $B(x_0, r)$, a real number $\mathcal{N}_\epsilon(\vec{P}, Q) \geq 0$, and a function $F \in C^m(\mathbb{R}^n)$, with the following properties.

(A) $Q^{**} \subset B(x_0, r)$, *and* $r \leq C\delta_Q$.

(B) F *agrees with* $\vec{P}$.

(C) $\| F \|_{C^m(B(x_0,r))} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, Q)$.

(D) $\mathcal{N}_\epsilon(\vec{P}, Q) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$, *and the work to answer a query is at most* C.

Explanation: Our algorithm consists of seven steps.

First, we present Steps 1...4. Then we introduce a family of linear constraints in a finite-dimensional vector space, and prove several lemmas on the feasible region. Next, we present Steps 5, 6, 7, in which we solve a linear programming problem and use the result to compute $\mathcal{N}_\epsilon(\vec{P}, Q)$ and F. Finally, we prove (A)...(D), and analyze the work and storage of our algorithm.

The first four steps of our algorithm are as follows.

Step 1: We compute a ball $B(x_0, r)$ such that

(4) $Q^{**} \subset B(x_0, r)$ and $r \leq C\delta_Q$.

From (1) and (4), we obtain

(5) $2r < \epsilon^{-1}$.

Note that

(6) $S \subset B(x_0, r)$, and

(7) $|y - y'| \geq c \cdot e^{-3/\epsilon} r$ for any two distinct points $y, y' \in S$,

thanks to (2), (3), (4).

Step 2: We apply Algorithm 11.3 ("Produce-Fine-Net") to the set S, the ball $B(x_0, r)$, and the number

(8) $\eta = e^{-6m/\epsilon}$.

Thanks to (6), (7), (8), the data $S, B(x_0, r)$, η satisfy the assumptions of Algorithm 11.3. Thus, that algorithm computes a set S^+, with the following properties.

(9) $S \subseteq S^+ \subset B(x_0, r)$

(10) $\mathrm{dist}(x, S^+) < \eta r$ for any $x \in B(x_0, r)$.

(11) $|y - y'| \geq \frac{\eta r}{100n}$ for any two distinct points $y, y' \in S^+$.

Note that (9) and (11) yield

(12) $\#(S) \leq \#(S^+) \leq C \cdot \eta^{-n}$.

Step 3: For each $y \in S^+$, we apply Algorithm 3.1 ("Find-Unit-Ball"), to compute a family $O(\epsilon, y)$ of linear functionals on $\mathcal{P}$, such that

(13) $(1+\epsilon)^{-1} \cdot |P|_y \leq \max\{\lambda(P) : \lambda \in O(\epsilon, y)\} \leq (1+\epsilon) \cdot |P|_y$ for all $P \in \mathcal{P}$ and $y \in S^+$; and

(14) $\#(O(\epsilon, y)) \leq \exp(C/\epsilon)$ for each $y \in S^+$.

Step 4: We compute the smallest real number $\hat{M} \geq 0$ such that

(15) $|\partial^\alpha P^y(y)| \leq \hat{M}$ for $|\alpha| \leq m$, $y \in S$; and

(16) $|\partial^\alpha (P^y - P^{y'})(y)| \leq \hat{M} \cdot |y - y'|^{m-|\alpha|}$ for $|\alpha| \leq m$, $y, y' \in S$, $y \neq y'$.

Before passing to Steps 5, 6, 7, we study the feasible region for a family of linear constraints, which we now introduce.

Let $\hat{A}$ be a constant, to be picked later. Assume that

(17) $\hat{A}$ exceeds a large enough controlled constant, and

(18) $\epsilon < \hat{A}^{-100}$.

Later, we will take $\hat{A}$ to be a controlled constant large enough to satisfy (17). Our assumption (18) will then hold, since we assume that ϵ is less than a small enough controlled constant. For the moment, however, we do not fix $\hat{A}$; we simply assume (17) and (18).

Now let X denote the vector space of all

(19) $\xi = [M, (P_y)_{y \in S^+}]$, with $M \in \mathbb{R}$ and each $P_y \in \mathcal{P}$.

Note that P_y in (19) is indexed by a subscript $y \in S^+$, while our given Whitney field $\vec{P}$ is equal to $(P^y)_{y \in S}$, with P^y indexed by a superscript $y \in S$.

We will study the following family of constraints on a vector $\xi \in X$ as in (19).

The Basic Constraints

(20) $0 \leq M \leq \hat{A} \cdot \hat{M}$.

(21) $\lambda(P_y) \leq (1 + \hat{A}\epsilon) \cdot M$ for each $\lambda \in O(\epsilon, y)$ and each $y \in S^+$.

(22) $|\partial^\alpha (P_y - P_{y'})(y)| \leq \hat{A} \cdot e^{4m/\epsilon} r^{-1} |y - y'|^{m+1-|\alpha|} \cdot M$ for $|\alpha| \leq m$ and $y, y' \in S^+$.

(23) $P_y = P^y$ for each $y \in S$.

We write $X(\epsilon, Q, \vec{P})$ to denote the set of all $\xi \in X$ that satisfy the above constraints. We will prove the following results about this feasible region.

Lemma 14.1. *For any* $[M, (P_y)_{y \in S^+}] \in X(\epsilon, Q, \vec{P})$*, we have*

(24) $c\hat{M} \leq M \leq \hat{A} \cdot \hat{M}$,

with $\hat{M}$ *as in Step* 4.

Lemma 14.2. *Fix* $y_0 \in S$*. Then every* $[M, (P_y)_{y \in S^+}] \in X(\epsilon, Q, \vec{P})$ *satisfies*

(25) $|\partial^\alpha (P_y - P^{y_0})(y)| \leq C\hat{A} e^{4m/\epsilon} r^{m-|\alpha|} M$*, for* $|\alpha| \leq m$*,* $y \in S^+$.

Lemma 14.3. *There exists* $\xi^\# = [M^\#, (P^\#_y)_{y \in S^+}] \in X$*, with the following properties.*

(26) $M^\# \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0, 2r))}$.

(27) *Let* $\xi' = [M', (P'_y)_{y \in S^+}] \in X$*, with*

(a) $|M'| \leq \exp(-\hat{A}/\epsilon) M^\#$*;*

(b) $|\partial^\alpha P'_y(y)| \leq \exp(-\hat{A}/\epsilon) r^{m-|\alpha|} M^\#$ *for* $|\alpha| \leq m$*,* $y \in S^+$*; and*

(c) $P'_y = 0$ *for all* $y \in S$.

Then

(d) $\xi^\# + \xi' \in X(\epsilon, Q, \vec{P})$.

Proof of Lemma 14.1. Let $[M, (P_y)_{y\in S^+}]$ satisfy (20) ...(23). Immediately from (20), we have $M \le \hat{A}\hat{M}$. Our desired result (24) thus amounts to saying that

(28) $\quad \hat{M} \le CM.$

Let us prove (28).

From (21) and the defining property (13) of $O(\epsilon, y)$, we obtain the estimate $|P_y|_y \le (1 + C\epsilon) \cdot (1 + \hat{A}\epsilon)M$ for each $y \in S^+$. Hence, by (17), (18), and the Bounded Distortion Property, we have

(29) $\quad |\partial^\alpha P_y(y)| \le CM$ for $|\alpha| \le m$, $y \in S^+$.

Since $r \le \tilde{c}\epsilon^{-1}$, we know that $\hat{A} \cdot e^{4m/\epsilon} r^{-1} > 1$. Hence, (22), (29) and the classical Whitney extension theorem for finite sets tell us that there exists

(30) $\quad F \in C^{m+1}(\mathbb{R}^n)$,

such that

(31) $\quad J_y(F) = P_y$ for each $y \in S^+$,

and

(32) $\quad |\partial^\alpha F(x)| \le C\hat{A} \cdot e^{4m/\epsilon} r^{-1} M$ for $|\alpha| \le m + 1$, $x \in \mathbb{R}^n$.

Here, $J_y(F)$ denotes an $m^{\underline{th}}$ order Taylor polynomial as usual, even though $F \in C^{m+1}(\mathbb{R}^n)$.

Let $x \in B(x_0, r)$ be given. By (8) and (10), there exists $y \in S^+$, such that

(33) $\quad |x - y| < e^{-6m/\epsilon} r.$

Note that $y \in B(x_0, r)$ as well; see (9). For $|\alpha| \le m$, we have

$$|\partial^\alpha F(x) - \partial^\alpha F(y)| \le |x - y| \cdot \max_{B(x_0,r)} |\nabla\, \partial^\alpha F| \le (e^{-6m/\epsilon} r) \cdot (C\hat{A}e^{4m/\epsilon} r^{-1} M)$$

by (32) and (33). Thus,

(34) $\quad |\partial^\alpha F(x) - \partial^\alpha F(y)| \le C\hat{A}e^{-2m/\epsilon} M \le M$ for $|\alpha| \le m$,

thanks to (17) and (18).

On the other hand, (29) and (31) give

(35) $\quad |\partial^\alpha F(y)| \le CM$ for $|\alpha| \le m$,

since $y \in S^+$.

From (34) and (35), we see that $|\partial^\alpha F(x)| \le CM$ for $|\alpha| \le m$, $x \in B(x_0, r)$. Hence, Taylor's theorem gives

(36) $\quad |\partial^\alpha (J_y(F) - J_{y'}(F))(y)| \le CM|y - y'|^{m-|\alpha|}$ for $|\alpha| \le m$, $y, y' \in B(x_0, r)$, $y \ne y'$.

From (31), (36) and (9), we conclude that

(37) $|\partial^\alpha(P_y - P_{y'})(y)| \le CM \cdot |y-y'|^{m-|\alpha|}$ for $|\alpha| \le m$, $y, y' \in S^+$, $y \neq y'$.

Now, recalling (23), and comparing (29) and (37) with the definition of $\hat{M}$ in Step 4, we see that

$$\hat{M} \le CM.$$

This is precisely our desired inequality (28). The proof of Lemma 14.1 is complete. ■

Proof of Lemma 14.2. Note that $P_{y_0} = P^{y_0}$ by (23), and that $|y-y_0| \le 2r$, by (9). Hence, for $|\alpha| \le m$, (22) gives

$$\begin{aligned} |\partial^\alpha(P_y - P^{y_0})(y)| &\le \hat{A}\, e^{4m/\epsilon}\, r^{-1} |y - y_0|^{m+1-|\alpha|}\, M \\ &\le 2\hat{A}\, e^{4m/\epsilon} |y - y_0|^{m-|\alpha|}\, M\,, \end{aligned}$$

which immediately implies the desired conclusion (25). ■

Proof of Lemma 14.3. We start by relating $\| \vec{P} \|_{C^m(B(x_0,2r))}$ to $\hat{M}$ (as in Step 4). By definition of $\hat{M}$, and by the classical Whitney extension theorem, there exists a function

(38) $\check{F} \in C^m(\mathbb{R}^n)$,

such that

(39) $J_y(\check{F}) = P^y$ for all $y \in S$,

and

(40) $|\partial^\alpha \check{F}(x)| \le C\hat{M}$ for $|\alpha| \le m$, $x \in \mathbb{R}^n$.

By (40) and the Bounded Distortion Property, we have $\| \check{F} \|_{C^m(\mathbb{R}^n)} \le C'\hat{M}$.

Together with (39) and the definition of the C^m-norm of a Whitney field, this yields

(41) $\| \vec{P} \|_{C^m(B(x_0,2r))} \le \| \vec{P} \|_{C^m(\mathbb{R}^n)} \le C'\hat{M}$.

Next, again using the definition of the C^m-norm of the Whitney field $\vec{P} = (P^y)_{y \in S}$, we see that there exists

(42) $\hat{F} \in C^m(B(x_0, 2r))$

with

(43) $J_y(\hat{F}) = P^y$ for $y \in S$,

and

(44) $\| \hat{F} \|_{C^m(B(x_0,2r))} \le (1+\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.

We now apply Lemma 12.2 to the following data.

(45)
- Our $\epsilon > 0$.
- The ball $B(x_0, 2r)$.
- The number $\hat{\eta} = \hat{c}\, e^{-4/\epsilon}$ for a small enough controlled constant $\hat{c}$.
- The set S.
- The number $M = (1 + \epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.
- The function $\hat{F}$.

Here, $\hat{\eta}$ in (45) plays the rôle of η in Lemma 12.2.

Let us check the hypotheses of that Lemma for the data (45).

We are assuming that $\epsilon > 0$ is less than a small enough controlled constant. We must check (†1)...(†5) from Lemma 12.2, for the data (45).

(†1) here asserts that $2r \leq \epsilon^{-1}$, which we know from (5).

(†2) here asserts that $0 < \hat{c}\, e^{-4/\epsilon} < \epsilon^2 e^{-1/\epsilon}$, which holds since $\hat{c} < 1$ and $\epsilon < 1$.

(†3) here asserts that $S \subset B(x_0, (1 - \hat{c}\, e^{-4/\epsilon}) \cdot 2r)$, which follows from (6), since $\hat{c} < 1$ and $\epsilon < 1$.

(†4) here asserts that $|y - y'| > 2 \cdot (\hat{c}\, e^{-4/\epsilon}) \cdot e^{1/\epsilon} \cdot (2r)$ for any two distinct points $y, y' \in S$; this holds (for $\hat{c}$ a small enough controlled constant), thanks to (7).

(†5) here asserts that $\hat{F} \in C^m(B(x_0, 2r))$, and that

$$\| \hat{F} \|_{C^m(B(x_0,2r))} \leq (1 + \epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}.$$

These assertions are precisely our results (42) and (44).

Thus, the hypotheses of Lemma 12.2 hold for the data (45). Applying that result, we learn that there exists

(46) $F^\# \in C^{m+1}(B(x_0, r))$,

with the following properties.

(47) $\| F^\# \|_{C^m(B(x_0,r))} \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.

(48) $J_y(F^\#) = J_y(\hat{F})$ for all $y \in S$.

(49) $|\partial^\alpha F^\#(x)| \leq C\, e^{4m/\epsilon}\, r^{-1} \| \vec{P} \|_{C^m(B(x_0,2r))}$ for $|\alpha| = m+1$, $x \in B(x_0, r)$.

Let us now define

(50) $M^\# = \| \vec{P} \|_{C^m(B(x_0,2r))}$

and also

(51) $P^\#_y = J_y(F^\#)$ for each $y \in S^+$.

Thus,

(52) $\xi^{\#} = [M^{\#}, (P^{\#}_y)_{y \in S^+}]$ belongs to X.

We derive some basic properties of $\xi^{\#}$.
First of all, (47), (51) and (13) yield the inequalities

(53) $\lambda(P^{\#}_y) = \lambda(J_y(F^{\#})) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0, 2r))}$ for $\lambda \in O(\epsilon, y)$ and $y \in S^+$.

Secondly, (49) and (51) yield the estimate

(54) $|\partial^{\alpha}(P^{\#}_y - P^{\#}_{y'})(y)| \leq C\, e^{4m/\epsilon}\, r^{-1} |y - y'|^{m+1-|\alpha|} \| \vec{P} \|_{C^m(B(x_0, 2r))}$ for $|\alpha| \leq m$, $y, y' \in S^+$,

by Taylor's theorem.

Thirdly, (43), (48) and (51) show that

(55) $P^{\#}_y = P^y$ for $y \in S$.

Finally, (41) and (50) give

(56) $0 \leq M^{\#} \leq C\hat{M}$.

The basic properties of $\xi^{\#}$ are (53)...(56).

Now let us show that the conclusions (26), (27) hold for $\xi^{\#}$. First of all, (26) is immediate from (50). To prove (27), let

$$\xi' = [M', (P'_y)_{y \in S^+}] \in X$$

satisfy (27)(a), (b), (c). We must show that

$$\xi^{\#} + \xi' = [M^{\#} + M', (P^{\#}_y + P'_y)_{y \in S^+}]$$

satisfies (27)(d), i.e., $\xi^{\#} + \xi'$ satisfies the Basic Constraints (20)...(23).

From (27)(a) and (17), we have

(57) $(1 - \epsilon)M^{\#} \leq M^{\#} + M' \leq (1 + \epsilon)M^{\#}$,

since $\exp(-\hat{A}/\epsilon) \leq \exp(-1/\epsilon) < \epsilon$. Immediately from (56) and (57), we have

$$0 \leq M^{\#} + M' \leq C'\hat{M} \leq \hat{A}\hat{M},$$

since $\hat{A}$ exceeds a large enough controlled constant.

Thus, constraint (20) is satisfied by the point $\xi^{\#} + \xi'$.

Next, note that (27)(b) and the Bounded Distortion Property show that

$$|P'_y|_y \leq C \exp(-\hat{A}/\epsilon) \cdot \epsilon^{-m} M^{\#}, \quad \text{for } y \in S^+,$$

since $r \leq \epsilon^{-1}$ (see (5)). By property (13) of the $O(\epsilon, y)$, it follows that

$$\lambda(P'_y) \leq C' \exp(-\hat{A}/\epsilon)\, \epsilon^{-m} M^{\#} \quad \text{for } \lambda \in O(\epsilon, y),\ y \in S^+.$$

Together with (53) and (50), this implies that

(58) $\lambda(P^{\#}_y + P'_y) \leq (1 + C''\epsilon) M^{\#}$ for $\lambda \in O(\epsilon, y)$, $y \in S^+$,

since $\exp(-\hat{A}/\epsilon)\, \epsilon^{-m} \leq \exp(-1/\epsilon) \cdot \epsilon^{-m} < \epsilon$.

From (57) and (58), we obtain the estimate

(59) $\lambda(P^{\#}_y + P'_y) \leq (1 + C'''\epsilon) \cdot [M^{\#} + M'] \leq (1 + \hat{A}\epsilon) \cdot [M^{\#} + M']$ for $\lambda \in O(\epsilon, y)$, $y \in S^+$,

since $\hat{A}$ exceeds a large enough controlled constant.
In view of (59), the constraint (21) is satisfied by the vector $\xi^{\#} + \xi'$.

We pass to constraint (22). Let $y, y' \in S^+$ be distinct points. Then by (9), we have

(60) $|y - y'| \leq 2r$.

From (27)(b), we have

(61) $|\partial^{\alpha} P'_y(y)| \leq \exp(-\hat{A}/\epsilon) r^{m-|\alpha|} M^{\#}$ for $|\alpha| \leq m$, and

(62) $|\partial^{\alpha} P'_{y'}(y')| \leq \exp(-\hat{A}/\epsilon)\, r^{m-|\alpha|} M^{\#}$ for $|\alpha| \leq m$.

From (60) and (62), we obtain $|\partial^{\alpha} P'_{y'}(y)| \leq C \exp(-\hat{A}/\epsilon)\, r^{m-|\alpha|} M^{\#}$ for $|\alpha| \leq m$.

Together with (61), this yields

(63) $|\partial^{\alpha}(P'_y - P'_{y'})(y)| \leq C \exp(-\hat{A}/\epsilon)\, r^{m-|\alpha|} M^{\#}$ for $|\alpha| \leq m$.

We have also

$$|y - y'| \geq c\, e^{-6m/\epsilon} r,$$

thanks to (8) and (11); consequently,

(64) $r^{m-|\alpha|} = r^{-1} \cdot r^{m+1-|\alpha|} \leq C r^{-1} \cdot e^{\frac{6m(m+1)}{\epsilon}} |y - y'|^{m+1-|\alpha|}$ for $|\alpha| \leq m$.

We substitute (64) into (63), and recall that $\hat{A} > 6m(m+1)$ by (17). This tells us that

(65) $|\partial^{\alpha}(P'_y - P'_{y'})(y)| \leq C\, r^{-1} |y - y'|^{m+1-|\alpha|} M^{\#}$ for $|\alpha| \leq m$.

From (50), (54) and (65) we now learn that

$$|\partial^\alpha([P^\#_y + P'_y] - [P^\#_{y'} + P'_{y'}])(y)| \le C\, e^{4m/\epsilon}\, r^{-1} |y - y'|^{m+1-|\alpha|} M^\# \quad \text{for } |\alpha| \le m.$$

Together with (57), this yields

(66) $|\partial^\alpha([P^\#_y + P'_y] - [P^\#_{y'} + P'_{y'}])(y)| \le C\, e^{4m/\epsilon}\, r^{-1} |y - y'|^{m+1-|\alpha|} \cdot [M^\# + M'] \le$

$$\le \hat{A} \cdot e^{4m/\epsilon}\, r^{-1} |y - y'|^{m+1-|\alpha|} \cdot [M^\# + M'] \text{ for } |\alpha| \le m,$$

by (17).

In view of (66), the constraints (22) hold for the vector $\xi^\# + \xi'$.

Finally, from (55) and (27)(c), we have

$$[P^\#_y + P'_y] = P^\#_y = P^y \quad \text{for } y \in S.$$

Thus, the constraints (23) are satisfied by $\xi^\# + \xi'$.

We have shown that all the constraints (20)...(23) are satisfied by the vector $\xi^\# + \xi'$. Thus, by definition, $\xi^\# + \xi' \in X(\epsilon, Q, \vec{P})$.

This is precisely conclusion (27)(d). Thus, (27)(a), (b), (c) imply (27)(d). This proves (27), and completes the proof of Lemma 14.3. ■

We now pick $\hat{A}$ to be a controlled constant, large enough to satisfy (17). As mentioned before, it follows that ϵ satisfies (18), since we are assuming that ϵ is less than a small enough controlled constant.

We note a simple consequence of Lemmas 14.1 and 14.3. Let $\xi^\#$ be as in Lemma 15.3. Then the point $\xi' = 0$ satisfies (27)(a), (b), (c), and therefore we learn from (27)(d) that $\xi^\#$ belongs to $X(\epsilon, Q, \vec{P})$. Consequently, Lemma 14.1 tells us that

(67) $c\hat{M} \le M^\# \le C\hat{M}$,

with $M^\#$ as in Lemma 14.1. Here, we have used the fact that we have picked $\hat{A}$ to be a controlled constant.

We prepare to use the Special Ellipsoid Algorithm from Section 4, to compute a point $[M^0, (P^0_y)_{y \in S^+}] \in X(\epsilon, Q, \vec{P})$, with M^0 nearly as small as possible. We have to check that the assumptions of that algorithm are satisfied. For this verification, we introduce rescaled (affine) coordinates as follows. We suppose for the moment that $\hat{M} \neq 0$. (See Step 5 below.)

Fix $y_0 \in S$. Given $\xi = [M, (P_y)_{y \in S^+}] \in X$, we define

(68) $\gamma(\xi) = [\nu_0, (\nu_{y,\alpha})_{\substack{y \in S^+ \smallsetminus S \\ |\alpha| \le m}}] \in \mathbb{R}^D$, where

(69) $\nu_0 = M/\hat{M}$,

(70) $\nu_{y,\alpha} = (\partial^\alpha(P_y - P^{y_0})(y))/(\hat{M}\, r^{m-|\alpha|})$ for $y \in S^+ \smallsetminus S$, $|\alpha| \le m$, and

(71) $D = 1 + \#\{\alpha : |\alpha| \le m\} \cdot \#(S^+ \smallsetminus S)$.

Note that the restriction of γ to the affine space of all $[M, (P_y)_{y\in S^+}] \in X$ satisfying (23) is an isomorphism; moreover, the inverse of this isomorphism is trivial to compute. We denote this inverse γ^{-1}.

The image $\gamma(X(\epsilon, Q, \vec{P})) \subset \mathbb{R}^D$ is the feasible region for a list of constraints that may be read off trivially from (20)...(23).

The number of constraints is

$$L = 1 + \sum_{y\in S^+} \#O(\epsilon, y) + 2[\#(S^+)]^2 \cdot \#\{\alpha : |\alpha| \le m\}, \tag{72}$$

as we see from (20)...(22); the constraints (23) do not contribute to L.

Recall that $\hat{A}$ is a controlled constant, and recall the estimate (67), with $M^\#$ as in Lemma 14.3. By comparing Lemmas 14.1, 14.2 and 14.3 with definitions (68), (69), (70), we see that the set $\gamma(X(\epsilon, Q, \vec{P}))$ has the following properties.

(73) For any $[\nu_0, (\nu_{y,\alpha})_{\substack{y\in S^+\smallsetminus S\\ |\alpha|\le m}}] \in \gamma(X(\epsilon, Q, \vec{P}))$, we have $c < \nu_0 < C$ and $|\nu_{y,\alpha}| \le C e^{4m/\epsilon}$ for $y \in S^+ \smallsetminus S$, $|\alpha| \le m$.

(74) Some translate of the set $\Big\{\big[\nu_0, (\nu_{y,\alpha})_{\substack{y\in S^+\smallsetminus S\\ |\alpha|\le m}}\big] \in \mathbb{R}^D : |\nu_0|, |\nu_{y,\alpha}| \le \exp(-C/\epsilon)$ (all $y \in S^+ \smallsetminus S$, $|\alpha| \le m)\Big\}$ is contained in $\gamma(X(\epsilon, Q, \vec{P}))$.

Consequently, the feasible region $\gamma(X(\epsilon, Q, \vec{P})) \subset \mathbb{R}^D$ satisfies the assumptions (a) and (b) of the Special Ellipsoid Algorithm, with

$$\wedge = C e^{4m/\epsilon} \text{ and } \lambda = e^{-C/\epsilon}. \tag{75}$$

(See Section 4.) Thus, the Special Ellipsoid Algorithm applies here.

We are now ready to describe the remaining steps in Algorithm 14.1.

Step 5: Using the Special Ellipsoid Algorithm, we compute a point

$$[\nu_0^0, (\nu_{y,\alpha}^0)_{\substack{y\in S^+\smallsetminus S\\ |\alpha|\le m}}] \in \gamma(X(\epsilon, Q, \vec{P})),$$

with

$$\nu_0^0 \le \inf\Big\{\nu_0 : \big[\nu_0, (\nu_{y,\alpha})_{\substack{y\in S^+\smallsetminus S\\ |\alpha|\le m}}\big] \in \gamma(X(\epsilon, Q, \vec{P}))\Big\} + \epsilon\,.$$

Applying γ^{-1} to $\big[\nu_0^0, (\nu_{y,\alpha}^0)_{\substack{y\in S^+\smallsetminus S\\ |\alpha|\le m}}\big]$, we obtain a point

$$\xi^0 = [M^0, (P_y^0)_{y\in S^+}] \in X(\epsilon, Q, \vec{P}), \tag{76}$$

such that

$$M^0 \le \inf\{M : [M, (P_y)_{y\in S^+}] \in X(\epsilon, Q, \vec{P})\} + \epsilon\hat{M}. \tag{77}$$

(See (69).)

In the degenerate case $\widehat{M} = 0$, we can check easily that (76), (77) hold for $\xi^0 = 0$. Thus, (76) and (77) hold in all cases.

Step 6: With ξ^0 and M^0 as in (76), we set

(78) $\mathcal{N}_\epsilon(\vec{P}, Q) = M^0$.

Step 7: We apply Algorithm 13.1 (extending a Whitney field from a fine net) to the following data.

(79)
- Our given $\epsilon > 0$.
- The number η from (8).
- A = a large enough controlled constant.
- The open ball $B(x_0, r)$ from (4).
- The Whitney field $\vec{P}^0 = (P^0_y)_{y \in S^+}$ from (76).

Let us check that the data (79) satisfy the assumptions of Algorithm 13.1. We know that $\epsilon > 0$ is less than a small enough controlled constant, and that $0 < \eta < \epsilon^2$ (see (8)). Also, A exceeds a large enough controlled constant, as assumed in Algorithm 13.1, since in (79) we make take A to be an even larger controlled constant. Recall from (5) that $r \leq \epsilon^{-1}$.

We now verify conditions (†0), (†1), (†2) of Algorithm 13.1, for the data (79). In fact, (†0) is immediate from (9); (†1) follows from (11), since we may take $A > 100n$ in (79); and (†2) is immediate from (10).

Thus, as claimed, all the assumptions of Algorithm 13.1 hold for the data(79). Applying that algorithm, we compute a function

(80) $F \in C^m(\mathbb{R}^n)$,

with the following properties.

(81) F agrees with $\vec{P}^0$.

(82) Let $M \geq 0$ be a real number, such that

(a) $|P^0_y|_y \leq M$ for all $y \in S^+$, and

(b) $|\partial^\alpha(P^0_y - P^0_{y'})(y)| \leq \epsilon M \cdot (\eta r)^{-1} \cdot |y - y'|^{m+1-|\alpha|}$ for $|\alpha| \leq m$, $y, y' \in S^+$.

Then

(c) $\| F \|_{C^m(B(x_0, r))} \leq (1 + C\epsilon) \cdot M$.

Thus, we have computed a ball $B(x_0, r)$ (see Step 1), a number $\mathcal{N}_\epsilon(\vec{P}, Q)$ (see Step 6), and a function $F \in C^m(\mathbb{R}^n)$ (see Step 7).

This completes our description of Algorithm 14.1.

Next, we prove (A)...(D) for the ball $B(x_0, r)$, the number $\mathcal{N}_\epsilon(\vec{P}, Q)$, and the function F computed above. First of all, (A) is simply (4). To check (B), note first that $[M^0, (P^0_y)_{y\in S^+}]$ satisfies constraints (20)...(23), thanks to (76). In particular, we have $P^0_y = P^y$ for $y \in S$. Therefore, (B) follows from (81).

To establish (C), we again note that constraints (20)...(23) are satisfied by $\xi^0 = [M^0, (P^0_y)_{y\in S^+}]$, thanks to (76). Applying (21) and (13), and recalling that $\hat{A}$ is a controlled constant, we learn that

$$|P^0_y|_y \le (1+\epsilon)\cdot(1+\hat{A}\epsilon)M^0 \le (1+C\epsilon)M^0 \text{ for } y \in S^+. \tag{83}$$

Also, applying (22) for ξ^0 and recalling (8), we find that

$$\begin{aligned}|\partial^\alpha(P^0_y - P^0_{y'})(y)| &\le \hat{A}\, e^{4m/\epsilon}\, r^{-1}|y-y'|^{m+1-|\alpha|}\cdot M^0\\ &= \hat{A}\cdot e^{-2m/\epsilon}\cdot(\eta r)^{-1}\cdot|y-y'|^{m+1-|\alpha|}\cdot M^0 \quad \text{for } |\alpha|\le m,\ y,y'\in S^+.\end{aligned}$$

Since $\hat{A}$ is a controlled constant, and since ϵ is less than a small enough controlled constant, we have $\hat{A}\, e^{-2m/\epsilon} < \epsilon$, and thus

$$|\partial(P^0_y - P^0_{y'})(y)| \le \epsilon M^0\cdot(\eta r)^{-1}\cdot|y-y'|^{m+1-|\alpha|} \text{ for } |\alpha|\le m \text{ and } y,y'\in S^+. \tag{84}$$

Estimates (83) and (84) show that conditions (82)(a) and (b) hold for $M = (1+C\epsilon)M^0$. Consequently, (82) tells us that

$$\| F \|_{C^m(B(x_0,r))} \le (1+C'\epsilon)M^0. \tag{85}$$

Since we set $\mathcal{N}_\epsilon(\vec{P}, Q) = M^0$ in Step 6, the estimate (85) is precisely our desired conclusion (C).

To prove (D), let $\xi^\# = [M^\#, (P^\#_y)_{y\in S^+}]$ be as in Lemma 14.3. Thus, (26) and (27) hold. Taking $\xi' = 0$ in (27), we conclude (as before) that $\xi^\# \in X(\epsilon, Q, \vec{P})$. Consequently, (26) implies that

$$\inf\{M : [M, (P_y)_{y\in S^+}] \in X(\epsilon, Q, \vec{P})\} \le (1+C\epsilon)\cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$$

and therefore (77) yields

$$M^0 \le (1+C\epsilon)\cdot \| \vec{P} \|_{C^m(B(x_0,2r))} + \epsilon\hat{M}. \tag{86}$$

Also, (76) and Lemma 14.1 show that

$$\hat{M} \le CM^0. \tag{87}$$

Substituting (87) into (86), we find that

$$M^0 \le (1+C\epsilon)\cdot \| \vec{P} \|_{C^m(B(x_0,2r))} + C\epsilon M^0,$$

and therefore

$$M^0 \le (1+C'\epsilon)\cdot \| \vec{P} \|_{C^m(B(x_0,2r))}. \tag{88}$$

Again recalling that $\mathcal{N}_\epsilon(\vec{P}, Q) = M^0$ by Step 6, we see that (88) is precisely our desired conclusion (D).

Thus, (A)...(D) hold for the ball $B(x_0, r)$, the number $\mathcal{N}_\epsilon(\vec{P}, Q)$, and the function F computed by Algorithm 14.1.

It remains to estimate the work and storage needed for Algorithm 14.1. Let us go over each of our seven steps.

Obviously, Step 1 requires work and storage at most C.

Step 2 requires work at most $C\eta^{-2n}$ and storage at most $C\eta^{-n}$. (See our description of Algorithm 11.3.) Since η is given by (8), the work and storage for Step 2 are at most $\exp(C/\epsilon)$.

Step 3 requires work and storage at most $\exp(C/\epsilon)$ for each $y \in S^+$. (See our description of Algorithm 3.1.) Thanks to (8) and (12), it follows that the work and storage for Step 3 are at most $\exp(C'/\epsilon)$.

Step 4 requires storage at most C (aside from the space used to hold our input $\vec{P}$). The work required for Step 4 is at most $C \cdot (\#(S))^2$, which is at most $\exp(C/\epsilon)$, thanks to (8) and (12).

Step 5 entails setting up and solving a linear programming problem. To set up and store the constraints (20)...(23) requires work and storage at most $\exp(C/\epsilon)$, as we see from (8), (12), (14) and from (20)...(23). To pass from (20)...(23) to the constraints defining $\gamma(X(\epsilon, Q, \vec{P}))$ as in (68)...(70) also requires work at most $\exp(C/\epsilon)$.

From (71), (72) and (8), (12), (14), we see that the dimension D, and the number of constraints L for our linear programming problem are both at most $\exp(C/\epsilon)$. Also, the numbers Λ and λ appearing in the assumptions (a), (b) of the Special Ellipsoid Algorithm are given by (75). Thus, $\Lambda \leq \exp(C/\epsilon)$ and $\lambda \geq \exp(-C/\epsilon)$.

Moreover, the quantity that plays the rôle of the "ϵ" in the Special Ellipsoid Algorithm is equal to $\epsilon' = \epsilon\Lambda^{-1} \geq \exp(-C'/\epsilon)$.

Consequently, the work and storage needed to apply the Ellipsoid Algorithm here are at most $CD^4 L \log\left(\frac{D\Lambda}{\lambda}\right) \log\left(\frac{D}{\epsilon'}\right)$ (see Section 4), which is at most $\exp(C/\epsilon)$, thanks to the above estimates for $D, L, \Lambda, \lambda, \epsilon'$.

Finally, it takes work at most $\exp(C/\epsilon)$ to apply γ^{-1} to the vector $\left[v_0^0, (v_{y,\alpha}^0)_{\substack{y \in S^+ \setminus S \\ |\alpha| \leq m}}\right]$ to obtain $\xi^0 = [M^0, (P_y^0)_{y \in S^+}]$.

Thus, the work and storage needed for Step 5 are at most $\exp(C/\epsilon)$.

Obviously, Step 6 requires work and storage at most C.

Finally, Step 7 entails one-time work at most $C \cdot \eta^{-2n}$, storage at most $C \cdot \eta^{-n}$, and query work at most C; this follows from our description of Algorithm 13.1, since we took A to be a controlled constant in (79). Since η is given by (8), we see that Step 7 requires storage and one-time work at most $\exp(C/\epsilon)$, and query work at most C.

Note that all the work of Steps 1...6 is one-time work. The only query work performed by Algorithm 14.1 occurs in Step 7. Consequently, the above estimates for the work and storage requirements of Steps 1...7 tell us the following:

Algorithm 14.1 requires storage and one-time work at most $\exp(C/\epsilon)$, and its query work is at most C.

Thus, Algorithm 14.1 performs as claimed. This completes our explanation of that algorithm.

For future reference, we record the following simple observations on the internal workings of Algorithm 14.1.

Proposition 14.1. *Let $\epsilon, Q, \vec{P} = (P^y)_{y\in S}$ be as assumed in Algorithm* 14.1. *Then Algorithm* 14.1 *performs the following actions.*

(a) *Using only Q (not ϵ or $\vec{P}$), it computes x_0, r such that $Q^{**} \subset B(x_0, r)$, and $r \leq C\delta_Q$.*

(b) *Using only ϵ, Q, S (not the P^y), it computes a set S^+, such that $S \subseteq S^+ \subset B(x_0, r)$ and $\#(S) \leq \exp(C/\epsilon)$.*

(c) *Using only ϵ and S^+, it computes for each $y \in S^+$ a finite family $O(\epsilon, y)$ of (real) linear functionals on $\mathcal{P}$, symmetric about the origin, with $\#(O(\epsilon, y)) \leq \exp(C/\epsilon)$.*

(d) *It computes $\xi^0 = [M^0, (P^0_y)_{y\in S^+}] \in \mathbb{R} \oplus \sum_{y\in S^+} \oplus \mathcal{P}$, with the following properties.*

- *The vector ξ^0 satisfies the constraints*

$$(\dagger)\quad \left[\begin{array}{l} P^0_y = P^y \ \ \textit{for } y \in S\,; \\ \\ \lambda(P^0_y) \leq (1 + \hat{A}\epsilon)\cdot M^0 \ \textit{for each } \lambda \in O(\epsilon, y),\ y \in S^+; \\ \\ |\partial^\alpha(P^0_y - P^0_{y'})(y)| \leq \hat{A}\cdot e^{4m/\epsilon}\, r^{-1}|y - y'|^{m+1-|\alpha|}\cdot M^0 \\ \\ \qquad\qquad\qquad\qquad\qquad\qquad \textit{for } |\alpha| \leq m \ \ \textit{and } y, y' \in S^+. \end{array}\right]$$

  Here $\hat{A}$ denotes a particular controlled constant.

- *Suppose $\xi^+ = [M^+, (P^+_y)_{y\in S^+}]$ is another vector satisfying the constraints (†). Then $M^0 \leq (1 + C\epsilon)M^+$.*

(e) *It returns the number $\mathcal{N}_\epsilon(\vec{P}, Q) = M^0$, with M^0 as in (d).*

(f) *The work and storage used to compute x_0, r, S^+ and all the $O(\epsilon, y)$ are at most $\exp(C/\epsilon)$.*

Proof. Assertions (a), (b), (c) are obvious from inspection of Steps 1,2,3 in Algorithm 14.1, together with property (O_0) of the sets $O(\epsilon, y)$ (see Sect. 3).

To check (d), we first note that the constraints (†) in (d) differ from the Basic Constraints (20)...(23), only in that (20) is missing from (†). However, omitting (20) has no effect on the infimum of all M^+ such that some $[M^+, (P^+_y)_{y\in S^+}]$ satisfies (20)...(23).
(To see this, note that the $\xi^0 = [M^0, (P^0_y)_{y\in S^+}]$ in (76) satisfies (20)...(23), with $M^0 \le \hat{A}\hat{M}$. If $\xi^+ = [M^+, (P^+_y)_{y\in S^+}]$ satisfies (21)...(23) but not (20), then we have $M^+ > \hat{A}\hat{M} \ge M^0$. Consequently, omitting such ξ^+ leaves the infimum in question unchanged.)

Thus, (d) follows, once we show that the ξ^0 computed in (Step 5) belongs to $X(\epsilon, Q, \vec{P})$ and satisfies

$$(\dagger\dagger) \quad M^0 \le (1 + C\epsilon) \cdot \inf\{M^+ : [M^+, (P^+_y)_{y\in S^+}] \in X(\epsilon, Q, \vec{P})\}.$$

Let $\Omega = \inf\{M^+ : [M^+, (P^+_y)_{y\in S^+}] \in X(\epsilon, Q, \vec{P})\}$. From (76), (77), we learn that $\xi^0 \in X(\epsilon, Q, \vec{P})$, and that $\Omega \le M^0 \le \Omega + \epsilon\hat{M}$. Moreover, (76) and Lemma 14.1 yield $\hat{M} \le CM^0$. Consequently,

$$\Omega \le M^0 \quad \text{and} \quad M^0 \le \Omega + C\epsilon M^0,$$

from which (††) follows trivially. This completes the proof of (d).

Next, note that (e) holds, by inspection of Step 6. Finally, (f) holds, since all the computations referred to in (f) are part of the one-time work of Algorithm 14.1, which is at most $\exp(C/\epsilon)$. The proof of Proposition 14.1 is complete. ■

15. Singletons

In this section, we give the algorithm that will be used later to compute the functions F^x in (15), (16) of Section 6.

Algorithm 15.1. ("Singleton") *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a Whitney field $\vec{P} = (P^y)_{y\in S}$ on a singleton $S = \{y_0\}$, we compute a number $\mathcal{N}_\epsilon(\vec{P}) \ge 0$, and a function $F \in C^m(\mathbb{R}^n)$, such that:*

(A) *F agrees with $\vec{P}$;*

(B) *$\| F \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$; and*

(C) *$\mathcal{N}_\epsilon(\vec{P}) \le (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.*

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$; and the work to answer a query is at most C.

Explanation: First we define F and $\mathcal{N}_\epsilon(\vec{P})$ and check (A), (B), (C). Then we explain how to compute our F and $\mathcal{N}_\epsilon(\vec{P})$. We compute a dyadic cube Q containing y_0, such that

(1) $\frac{1}{8}\tilde{c}\epsilon^{-1} \leq \delta_Q \leq \tilde{c}\epsilon^{-1}$, with $\tilde{c}$ as in Algorithm 14.1.

(The work and storage needed to compute Q are at most C.)

Then $\epsilon, Q, \vec{P}$ satisfy conditions (1), (2), (3) in Section 14, and therefore Algorithm 14.1 applies. That algorithm computes a ball $B(x_0, r)$, a number $\mathcal{N}_\epsilon(\vec{P}, Q) \geq 0$, and a function $F_0 \in C^m(\mathbb{R}^n)$, with the following properties.

(2) $Q^{**} \subset B(x_0, r)$ and $r \leq C\delta_Q$.

(3) F_0 agrees with $\vec{P}$.

(4) $\| F_0 \|_{C^m(B(x_0,r))} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, Q)$.

(5) $\mathcal{N}_\epsilon(\vec{P}, Q) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.

Moreover, the storage and one-time work used by Algorithm 14.1 are at most $\exp(C/\epsilon)$, and the work at query time is at most C.

We now take $\theta_0 \in C^m(\mathbb{R}^n)$ to be a cutoff function, such that

(6) $0 \leq \theta_0 \leq 1$ on $\mathbb{R}^n$; $\theta_0 = 1$ on Q; $\operatorname{supp}\theta_0 \subset (Q^{**})^{\text{int}}$; $|\partial^\alpha \theta_0(x)| \leq C\epsilon^{|\alpha|}$ for $0 < |\alpha| \leq m$, $x \in \mathbb{R}^n$.

This is possible thanks to (1). Moreover, we may take θ_0 to be an appropriate spline, so that we can answer queries as follows.

(7) Given a query point $\underline{x} \in \mathbb{R}^n$, we can compute $J_{\underline{x}}(\theta_0)$ with work and storage at most C.

There is no one-time work involved in computing θ_0.
We also take $\epsilon_0 = C\epsilon$, so that (6) yields

(8) $|\partial^\alpha \theta_0(x)| \leq \epsilon_0$ for $0 < |\alpha| \leq m$, $x \in \mathbb{R}^n$.

Since ϵ is assumed to be less than a small enough controlled constant, we have

(9) $0 < \epsilon_0 < 1$.

We take $Q_0 = Q^{**}$ and $M = (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, Q)$, with C as in (4). Thus (2), (4) and (6) yield

(10) $\operatorname{supp}\theta_0 \subset (Q_0)^{\text{int}}$, and

(11) $F_0 \in C^m(Q_0^{\text{int}})$, $\| F_0 \|_{C^m(Q_0^{\text{int}})} \leq M$.

Thanks to (6), (8), (9), (10), (11), the hypotheses of Corollary 2 in Section 5 hold for $\epsilon_0, \theta_0, F_0, Q_0$. Applying that corollary, we see that the function

(12) $F = \theta_0 \cdot F_0$ on $\mathbb{R}^n$

satisfies

(13) $F \in C^m(\mathbb{R}^n)$, and

(14) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon)\mathcal{N}_\epsilon(\vec{P}, Q)$.

We define

(15) $\mathcal{N}_\epsilon(\vec{P}) = \mathcal{N}_\epsilon(\vec{P}, Q) \geq 0$,

and we check that $\mathcal{N}_\epsilon(\vec{P})$, F satisfy (A), (B), (C). In fact, since $\theta_0 = 1$ on Q and $S = \{y_0\}$ with $y_0 \in Q$, we have $J_{y_0}(F) = J_{y_0}(\theta_0 \cdot F_0) = J_{y_0}(F_0) = P^{y_0}$, thanks to (3). Thus, (A) holds. Also, (B) is immediate from (14), (15); and (C) follows from (5), since

$$\| \vec{P} \|_{C^m(B(x_0,2r))} \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)} .$$

It remains to show how to compute the above $\mathcal{N}_\epsilon(\vec{P})$ and F, and to estimate the work and storage of the computation.

This is easy. The one-time work is as follows.

- Compute Q.
- Perform the one-time work of Algorithm 14.1.
- Set $\mathcal{N}_\epsilon(\vec{P}) = \mathcal{N}_\epsilon(\vec{P}, Q)$. (The right-hand side has been computed as part of the one-time work of Algorithm 14.1)

After we have done the one-time work, we can answer queries as follows. Given a query point $\underline{x} \in \mathbb{R}^n$,

- We compute $J_{\underline{x}}(F_0)$ by the query algorithm in Algorithm 14.1.
- We compute $J_{\underline{x}}(\theta_0)$ as in (7).
- We return the polynomial $J_{\underline{x}}(F) = J_{\underline{x}}(\theta_0) \odot_{\underline{x}} J_{\underline{x}}(F_0)$.

One checks easily that the storage and one-time work here are at most $\exp(C/\epsilon)$, and the query work is at most C.

This completes our explanation of Algorithm 15.1.

For future reference, we record a few remarks on the inner workings of Algorithm 15.1.

Proposition 15.1. *Let* $\epsilon, \vec{P} = (P^y)_{y\in S}$, $S = \{y_0\}$ *be as assumed in Algorithm* 15.1. *Then Algorithm* 15.1 *performs as follows.*

(a) *It computes a dyadic cube* Q *containing* y_0. *The cube* Q *is computed from* ϵ *and* y_0, *without using the polynomial* P^{y_0}. *The work and storage used to compute* Q *are at most* C.

(b) *The number* $\mathcal{N}_\epsilon(\vec{P})$ *returned by Algorithm* 15.1 *is equal to the number* $\mathcal{N}_\epsilon(\vec{P}, Q)$ *returned by Algorithm* 14.1 *for the input data* $\epsilon, Q, \vec{P}$. *(In particular,* $\epsilon, Q, \vec{P}$ *are as assumed in Algorithm* 14.1).

Proof. Assertion (a) is immediate from the first paragraph of our explanation of Algorithm 15.1.

Regarding (b), we recall from the second paragraph of our explanation of Algorithm 15.1 that $\epsilon, Q, \vec{P}$ are as assumed in Algorithm 14.1. The assertion in (b) regarding $\mathcal{N}_\epsilon(\vec{P})$ is simply equation (15). ■

16. Extending a Whitney Field from a Testing Set I

The next few sections provide the algorithms that compute the functions F^ℓ_ν in (13)...(16) of Section 6. We treat Whitney fields $\vec{P} = (P^y)_{y\in S}$, with $S \subset Q^{**}$ for a dyadic cube Q. We distinguish several cases, depending on the size of Q. This section deals with the case in which Q is quite small.

Algorithm 16.1. *Given* $\epsilon > 0$, *assumed to be less than a small enough controlled constant; and given a dyadic cube* Q *whose sidelength satisfies*

(1) $\delta_Q \leq e^{-1/(4\epsilon)}$;

and given a Whitney field $\vec{P} = (P^y)_{y\in S}$, *where the set* S *is assumed to satisfy*

(2) $S \subset Q^{**}$, *and*

(3) $|y - y'| > \epsilon^2\, e^{-2/\epsilon}\, \delta_Q$ *for any two distinct points* $y, y' \in S$;

we compute a number $\mathcal{N}_\epsilon(\vec{P})$ *and a function* $F \in C^m(\mathbb{R}^n)$, *such that*

(A) F *agrees with* $\vec{P}$;

(B) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$; *and*

(C) $\mathcal{N}_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$, *and the work to answer a query is at most* C.

Explanation: We first define $\mathcal{N}_\epsilon(\vec{P})$, F and prove (A), (B), (C); then, we explain how to compute our $\mathcal{N}_\epsilon(\vec{P}), F$. Fix a point

(4) $\quad y_0 \in S$.

Applying Algorithm 10.6, we obtain a function

(5) $\quad \theta_0 \in C^{m+1}(\mathbb{R}^n)$,

with the following properties.

(6) $\quad 0 \le \theta_0 \le 1$ on $\mathbb{R}^n$.

(7) $\quad \operatorname{supp}\theta_0 \subset B(y_0, e^{1/(8\epsilon)}\delta_Q)$, and $\theta_0 = 1$ on $B(y_0, e^{1/(16\epsilon)}\delta_Q)$.

(8) $\quad |\partial^\alpha \theta_0(x)| \le C\epsilon \cdot |x - y_0|^{-|\alpha|}$ for $0 < |\alpha| \le m+1$, $x \in \mathbb{R}^n \setminus \{y_0\}$.

Moreover, we can answer queries as follows.

(9) $\quad$ Given $\underline{x} \in \mathbb{R}^n$, we can compute $J_{\underline{x}}(\theta_0)$ with work and storage at most C.

We fix a dyadic cube Q_{00}, such that

(10) $\quad B(y_0, 2\cdot e^{1/(8\epsilon)}\delta_Q) \subset Q_{00}^*$,

and

(11) $\quad e^{1/(8\epsilon)}\delta_Q \le \delta_{Q_{00}} \le C \cdot e^{1/(8\epsilon)}\delta_Q$.

Note that (1) and (11) yield

(12) $\quad \delta_{Q_{00}} \le Ce^{-1/(8\epsilon)} < 1$.

Note also that, since $y_0 \in Q^{**}$ by (2) and (4), we have

(13) $\quad S \subset Q^{**} \subset B(y_0, C\delta_Q) \subset B(y_0, e^{1/(16\epsilon)}\delta_Q) \subset B(y_0, 2\cdot e^{1/(8\epsilon)}\delta_Q) \subset Q_{00}^*$,

thanks to (2) and (10).

Moreover, if y and y' are distinct points of S, then (3) and (11) yield

(14) $\quad |y - y'| > \epsilon^2 e^{-2/\epsilon}\delta_Q \ge \epsilon^2 \cdot e^{-2/\epsilon} \cdot c\, e^{-1/(8\epsilon)}\delta_{Q_{00}} > e^{-3/\epsilon}\,\delta_{Q_{00}}$.

We apply Algorithm 14.1 to the data $\epsilon, Q_{00}, \vec{P}$. (Note that the assumptions (1), (2), (3) of Algorithm 14.1 follow at once from our present results (12), (13), (14).) Thus, we obtain a ball $B(x_0, r)$, a number $\mathcal{N}_\epsilon(\vec{P}, Q_{00}) \ge 0$, and a function $F_0 \in C^m(\mathbb{R}^n)$, with the following properties.

(15) $\quad Q_{00}^{**} \subset B(x_0, r)$ and $r \le C\delta_{Q_{00}}$.

(16) $\quad F_0$ agrees with $\vec{P}$.

(17) $\quad \| F_0 \|_{C^m(B(x_0,r))} \le (1 + C'\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, Q_{00})$.

(18) $\quad \mathcal{N}_\epsilon(\vec{P}, Q_{00}) \le (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.

(19) $\quad$ Moreover, we can compute $\mathcal{N}_\epsilon(\vec{P}, Q_{00})$ and F_0, with storage and one-time work at most $\exp(C/\epsilon)$ and with query work at most C.

Also, by applying Algorithm 15.1 ("Singleton"), we compute a number $\mathcal{N}_\epsilon(\vec{P}, y_0) \geq 0$, and a function $F_1 \in C^m(\mathbb{R}^n)$, with the following properties.

(20) F_1 agrees with $\vec{P}$ at y_0.

(21) $\| F_1 \|_{C^m(\mathbb{R}^n)} \leq (1 + C''\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, y_0)$.

(22) $\mathcal{N}_\epsilon(\vec{P}, y_0) \leq (1 + C\epsilon) \cdot \| (\vec{P}|_{\{y_0\}}) \|_{C^m(\mathbb{R}^n)}$.

(23) Moreover, we can compute $\mathcal{N}_\epsilon(\vec{P}, y_0)$ and F_1, with storage and one-time work at most $\exp(C/\epsilon)$, and with query work at most C.

We prepare to apply Corollary 1 in Section 5 to the above $y_0, F_0, F_1, \theta_0, \delta_Q$, taking $\tilde{A}$ to be a large enough controlled constant, and taking

(24) $M = (1 + C\epsilon) \max(\mathcal{N}_\epsilon(\vec{P}, Q_{00}), \mathcal{N}_\epsilon(\vec{P}, y_0))$, with $C = \max(C', C'')$;

here, C' and C'' are as in (17) and (21).

The hypotheses of that corollary regarding θ_0 hold here, thanks to our present results (5)...(8). Also, (10) and (15) yield $B(y_0, 2 \cdot e^{1/(8\epsilon)}\delta_Q) \subset B(x_0, r)$, and therefore (17) and (24) give

$$\| F_0 \|_{C^m(B(y_0, e^{1/(8\epsilon)}\delta_Q))} \leq M\,.$$

Since furthermore

$$\| F_1 \|_{C^m(\mathbb{R}^n)} \leq M$$

by (21) and (24), the hypotheses of Corollary 1 in Section 5 concerning the C^m-norms of F_0 and F_1 are satisfied here.

Finally, (16) and (20) show that

$$J_{y_0}(F_0) = J_{y_0}(F_1)\,,$$

completing our verification of the hypotheses of Corollary 1 in Section 5. Applying that Corollary, we now learn that the function

(25) $F = \theta_0 \cdot F_0 + (1 - \theta_0) \cdot F_1$ on $\mathbb{R}^n$

satisfies

(26) $F \in C^m(\mathbb{R}^n)$,

and

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot M;$$

hence

(27) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \max[\mathcal{N}_\epsilon(\vec{P}, Q_{00}), \mathcal{N}_\epsilon(\vec{P}, y_0)]$.

We take F as in (25), and set

$$\mathcal{N}_\epsilon(\vec{P}) = \max[\mathcal{N}_\epsilon(\vec{P}, Q_{00}), \mathcal{N}_\epsilon(\vec{P}, y_0)]. \tag{28}$$

We check that (A), (B), (C) hold for the above F and $\mathcal{N}_\epsilon(\vec{P})$. From (7) and (13), we have $J_y(\theta_0) = 1$ for $y \in S$, and therefore (25), (16) yield

$$J_y(F) = J_y(F_0) = P^y \quad \text{for } y \in S\ ,$$

proving (A). Also, (B) is immediate from (27) and (28). Finally, (C) follows from (18), (22) and (28), since

$$\| \vec{P} \|_{C^m(B(x_0,2r))} \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)}, \quad \text{and} \quad \| (\vec{P}|_{\{y_0\}}) \|_{C^m(\mathbb{R}^n)} \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)}\ .$$

Thus, (A), (B), (C) hold for our F and $\mathcal{N}_\epsilon(\vec{P})$.

It remains to compute F and $\mathcal{N}_\epsilon(\vec{P})$, and to estimate the work and storage needed for the computation. This is straightforward. The one-time work is as follows.

- Find a point y_0 in S.
- Compute a dyadic cube Q_{00} satisfying (10) and (11).
- Perform the one-time work associated with Algorithm 14.1 for the input data $(\epsilon, Q_{00}, \vec{P})$.
- Perform the one-time work associated with Algorithm 15.1 for the input data $(\epsilon, \vec{P}|_{\{y_0\}})$.
- Compute $\mathcal{N}_\epsilon(\vec{P})$ from (28).

Thanks to (19) and (23), one-time work and storage consumed in carrying out the above steps are at most $\exp(C/\epsilon)$

The query algorithm proceeds as follows.

Given a query point $\underline{x} \in \mathbb{R}^n$, we compute $J_{\underline{x}}(\theta_0)$, $J_{\underline{x}}(F_0)$, $J_{\underline{x}}(F_1)$ from (9), (19), (23), respectively. We then return the polynomial

$$J_{\underline{x}}(F) = J_{\underline{x}}(\theta_0) \odot_{\underline{x}} J_{\underline{x}}(F_0) + (1 - J_{\underline{x}}(\theta_0)) \odot_{\underline{x}} J_{\underline{x}}(F_1)$$

(See (25)). The work involved here is at most C, as we see from (9), (19), (23).

Thus, the storage and one-time work to compute $\mathcal{N}_\epsilon(\vec{P})$ and F are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C.

This concludes our explanation of Algorithm 16.1.

For future reference, we record a few remarks on the inner workings of Algorithm 16.1.

Proposition 16.1. *Let* $\epsilon, Q, \vec{P} = (P^y)_{y \in S}$ *be as assumed in Algorithm* 16.1. *Then Algorithm* 16.1 *performs as follows.*

(a) *It computes a point* $y_0 \in S$ *and a dyadic cube* Q_{00}. *The point* y_0 *and the cube* Q_{00} *are computed from* ϵ, Q, S, *without using the polynomials* $P^y (y \in S)$. *The work and storage used to compute* y_0 *and* Q_{00} *are at most* C.

(b) *It applies Algorithm* 14.1 *to the input data* $\epsilon, Q_{00}, \vec{P}$, *to compute a number* $\mathcal{N}_\epsilon(\vec{P}, Q_{00})$. *(In particular, the input data* $\epsilon, Q_{00}, \vec{P}$ *are as assumed for Algorithm* 14.1.)

(c) *It applies Algorithm* 15.1 *to the input data* $\epsilon, \vec{P}|_{\{y_0\}}$, *to compute a number* $\mathcal{N}_\epsilon(\vec{P}, y_0)$. *(In particular, the input data* $\epsilon, \vec{P}|_{\{y_0\}}$ *are as assumed for Algorithm* 15.1.)

(d) *It returns the number* $\mathcal{N}_\epsilon(\vec{P}) = \max[\mathcal{N}_\epsilon(\vec{P}, Q_{00}), \mathcal{N}_\epsilon(\vec{P}, y_0)]$.

Proof. Immediate from our explanation of Algorithm 16.1. ■

17. Extending a Whitney Field from a Testing Set II

In this section, we present the analogue of Algorithm 16.1 in the case of a cube Q that is neither very big nor very small.

Algorithm 17.1. *Given* $\epsilon > 0$, *assumed to be less than a small enough controlled constant; and given a dyadic cube* Q *whose sidelength satisfies*

(1) $e^{-1/(2\epsilon)} \leq \delta_Q \leq c^\# \epsilon^{-1}$ *for a small enough controlled constant* $c^\#$;

and given a Whitney field $\vec{P} = (P^y)_{y \in S}$, *where the set* S *is assumed to satisfy*

(2) $S \subset Q^{**}$, *and*

(3) $|y - y'| \geq \epsilon^2 e^{-2/\epsilon} \delta_Q$ *for any two distinct points* $y, y' \in S$;

we compute a number $\mathcal{N}_\epsilon(\vec{P})$ *and a function* $F \in C^m(\mathbb{R}^n)$, *such that*

(A) F *agrees with* $\vec{P}$;

(B) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$; *and*

(C) $\mathcal{N}_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$, *and the work to answer a query is at most* C.

Explanation: First, we define $\mathcal{N}_\epsilon(\vec{P})$ and F and prove that they satisfy (A), (B) and (C); then we discuss the computation of our $\mathcal{N}_\epsilon(\vec{P})$ and F.

We fix a dyadic cube Q_{00}, such that

(4) $Q^{**} \subset Q^*_{00}$, and

(5) $\frac{1}{80}\,\tilde{c}\,\epsilon^{-1} \le \delta_{Q_{00}} \le \frac{1}{20}\,\tilde{c}\,\epsilon^{-1}$, with $\tilde{c}$ as in equation (1) in Section 14.

Such a cube Q_{00} exists, thanks to our assumption (1).

Next, we fix a cutoff function $\theta_0 \in C^m(\mathbb{R}^n)$, with the following properties:

(6) $0 \le \theta_0 \le 1$ on $\mathbb{R}^n$; $\theta_0 = 1$ on Q^*_{00}; $\operatorname{supp}\theta_0 \subset (Q^{**}_{00})^{\mathrm{int}}$; and

(7) $|\partial^\alpha \theta_0(x)| \le C\,\delta^{-|\alpha|}_{Q_{00}}$ for $|\alpha| \le m$, $x \in \mathbb{R}^n$.

By taking θ_0 to be an appropriate spline, we can answer queries as follows.

(8) Given a point $\underline{x} \in \mathbb{R}^n$, we compute $J_{\underline{x}}(\theta_0)$ with work and storage at most C.

In view of (5) and (7), we have

(9) $|\partial^\alpha \theta_0(x)| \le C\epsilon$ for $0 < |\alpha| \le m$, $x \in \mathbb{R}^n$, since $\epsilon < 1$.

We prepare to apply Algorithm 14.1 to the data $\epsilon, Q_{00}, \vec{P}$. Let us check that these data satisfy the assumptions of that algorithm. We are assuming here that ϵ is less than a small enough controlled constant. Also, Q_{00} is a dyadic cube, whose sidelength $\delta_{Q_{00}}$ satisfies equation (1) in Section 14, thanks to our present equation (5). Also, our Whitney field $\vec{P} = (P^y)_{y\in S}$ satisfies

(10) $S \subset Q^{**} \subset Q^*_{00} \subset Q^{**}_{00}$

thanks to (2) and (4). Consequently, inclusion (2) in Section 14 holds for the data $\epsilon, Q_{00}, \vec{P}$.

Next, note that

$$\delta_Q \ge e^{-1/(2\epsilon)} \ge c\,\epsilon\,e^{-1/(2\epsilon)}\,\delta_{Q_{00}},$$

by (1) and (5). Therefore, (3) yields

$$|y - y'| \ge \epsilon^2 e^{-2/\epsilon}\,\delta_Q \ge c\,\epsilon^3 e^{-2.5/\epsilon}\,\delta_{Q_{00}} > e^{-3/\epsilon}\,\delta_{Q_{00}}$$

for any two distinct points $y, y' \in S$, since ϵ is less than a small enough controlled constant.

Thus, equation (3) in Section 14 holds for the data $\epsilon, Q_{00}, \vec{P}$. This completes the verification of the assumptions of Algorithm 14.1 for the data $\epsilon, Q_{00}, \vec{P}$.

Applying Algorithm 14.1 to $\epsilon, Q_{00}, \vec{P}$, we compute a ball $B(x_0, r)$, a number $\mathcal{N}_\epsilon(\vec{P}, Q_{00})$, and a function $F_0 \in C^m(\mathbb{R}^n)$, with the following properties.

(11) $(Q_{00})^{**} \subset B(x_0, r)$.

(12) F_0 agrees with $\vec{P}$.

(13) $\| F_0 \|_{C^m(B(x_0,r))} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, Q_{00})$.

(14) $\mathcal{N}_\epsilon(\vec{P}, Q_{00}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(B(x_0,2r))}$.

Moreover,

(15) The storage and one-time work to compute $B(x_0, r), \mathcal{N}_\epsilon(\vec{P}, Q_{00})$, F_0 are at most $\exp(C/\epsilon)$; and the work to answer a query is at most C.

We now define

(16) $\mathcal{N}_\epsilon(\vec{P}) = \mathcal{N}_\epsilon(\vec{P}, Q_{00})$, and

(17) $F = \theta_0 \cdot F_0 \in C^m(\mathbb{R}^n)$.

Let us check that (A), (B), (C) hold for $\mathcal{N}_\epsilon(\vec{P})$ and F as in (16), (17). First of all, let $y \in S$. Then $y \in Q_{00}^*$ by (10), hence $J_y(\theta_0) = 1$ by (6). Consequently, $J_y(F) = J_y(F_0)$ by (17), and therefore $J_y(F) = P^y$ by (12). Thus, our F satisfies (A).

Next, we estimate the norm of F in $C^m(\mathbb{R}^n)$, in order to check (B).

We will apply Corollary 2 in Section 5 to the following data:

(18)
- The number $\epsilon_0 = C\epsilon$ for a large enough controlled constant C.
- The number $M = \| F_0 \|_{C^m(B(x_0,r))}$.
- The cube $Q_0 = Q_{00}^{**}$.
- The cutoff function θ_0 from (6)...(9).
- The function F_0 from (11)...(15).

Let us check that the hypotheses of that corollary are satisfied by the data (18). In fact, we have $0 < \epsilon_0 < 1$, since ϵ is less than a small enough controlled constant. Also, $M \geq 0$, Q_0 is a cube, $\theta_0 \in C^m(\mathbb{R}^n)$, and $F_0 \in C^m(Q_0^{int})$. We have $0 \leq \theta_0 \leq 1$ on $\mathbb{R}^n$; $\mathrm{supp}\theta_0 \subset Q_0^{int}$; and $|\partial^\alpha \theta_0(x)| \leq \epsilon_0$ for $0 < |\alpha| \leq m$, $x \in \mathbb{R}^n$; all thanks to (6), (9) and (18). Also,

$$\| F_0 \|_{C^m(Q_0^{int})} = \| F_0 \|_{C^m((Q_{00}^{**})^{int})} \leq \| F_0 \|_{C^m(B(x_0,r))} = M,$$

thanks to (11) and (18). This completes the verification of the hypotheses of Corollary 2 in Section 5 for the data (18).

Applying that corollary, and recalling (17), we learn that

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \| F_0 \|_{C^m(B(x_0,r))} .$$

Together with (13), this yields

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}, Q_{00}),$$

which is our desired conclusion (B), in view of definition (16).

Next, note that conclusion (C) follows at once from (14) and (16). Thus, conclusions (A), (B), (C) hold for $\mathcal{N}_\epsilon(\vec{P})$, F as in (16), (17).

We turn to the computation of $\mathcal{N}_\epsilon(\vec{P})$ and $\vec{F}$. The one-time work is as follows.

Step 1: We compute a dyadic cube Q_{00} satisfying (4) and (5).

Step 2: We perform the one-time work of Algorithm 14.1 for the data $\epsilon, Q_{00}, \vec{P}$.

Step 3: We set $\mathcal{N}_\epsilon(\vec{P}) = \mathcal{N}_\epsilon(\vec{P}, Q_{00})$, where the right-hand side has been computed in Step 2.

Given a query point $\underline{x} \in \mathbb{R}^n$, the query algorithm proceeds as follows.

Step Q1: We compute $J_{\underline{x}}(F_0)$ by the query algorithm of Algorithm 14.1, applied to the data $\epsilon, Q_{00}, \vec{P}$.

Step Q2: We compute $J_{\underline{x}}(\theta_0)$ as in (8).

Step Q3: We return the polynomial $J_{\underline{x}}(F) = J_{\underline{x}}(\theta_0) \odot_{\underline{x}} J_{\underline{x}}(F_0)$. (See (17).)

From (8) and (15), one sees trivially that the storage and one-time work of the above algorithm are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C. This completes our explanation of Algorithm 17.1.

For future reference, we record a few observations on the inner workings of Algorithm 17.1.

Proposition 17.1. *Let $\epsilon, Q, \vec{P}$ be as assumed in Algorithm* 17.1. *Then the Algorithm* 17.1 *performs as follows.*

(a) *It computes a dyadic cube Q_{00}, using only ϵ and Q (but not using $\vec{P}$). The work and storage used to compute Q_{00} are at most C.*

(b) *It applies Algorithm* 14.1 *to the input data $\epsilon, Q_{00}, \vec{P}$, to compute a number $\mathcal{N}_\epsilon(\vec{P}, Q_{00})$. (In particular, the input data $\epsilon, Q_{00}, \vec{P}$ are as assumed in Algorithm* 14.1.)

(c) *It returns the number*

$$\mathcal{N}_\epsilon(\vec{P}) = \mathcal{N}_\epsilon(\vec{P}, Q_{00}) .$$

Proof. Immediate from our explanation of Algorithm 17.1. ∎

18. Extending a Whitney Field from a Testing Set III

In this section, we provide the analogue of Algorithms 16.1 and 17.1 in the case of a rather large cube Q.

Algorithm 18.1. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube Q whose sidelength satisfies*

(1) $\frac{1}{2}c^{\#}\epsilon^{-1} \leq \delta_Q \leq e^{10/\epsilon}$ *with $c^{\#}$ as in Algorithm* 17.1;

and given a Whitney field $\vec{P} = (P^y)_{y\in S}$, where the set S is assumed to satisfy

(2) $S \subset Q^{**}$, *and*

(3) $|y - y'| \geq \epsilon^2\, e^{-2/\epsilon}\, \delta_Q$ *for any two distinct points $y, y' \in S$;*

we compute a number $N_\epsilon(\vec{P})$ and a function $F \in C^m(\mathbb{R}^n)$, such that

(A) F *agrees with $\vec{P}$;*

(B) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot N_\epsilon(\vec{P})$; *and*

(C) $N_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C.

Explanation: We will reduce matters to Algorithm 17.1 by a partition of unity. We begin by introducing that partition of unity. Next, we define $N_\epsilon(\vec{P})$ and F, and check (A), (B), (C). Then we show how to compute our $N_\epsilon(\vec{P})$ and F, and estimate the work and storage needed.

Recall the cubes $Q_\nu^{\langle s\rangle}$ and cutoff functions $\theta_\nu^{\langle s\rangle}$, $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$, defined in Section 10. We fix $s \in \mathbb{Z}$, such that

(4) $\frac{1}{1000}\, c^{\#}\, \epsilon^{-1} \leq 2^s \leq \frac{1}{100}\, c^{\#}\, \epsilon^{-1}$.

Thanks to (1) and (4), the cube Q^{***} is partitioned into dyadic subcubes $Q_\nu^{\langle s\rangle}$, for $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$ varying over the set

(5) $\mathcal{G} = \{(\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n\colon\ \nu_i^{\min} \leq \nu_i \leq \nu_i^{\max}$ for each $i\}$, for suitable $\nu_i^{\max}$, $\nu_i^{\min}$ $(i = 1, \ldots, n)$.

Also from (1) and (4), we see that

(6) $\#(\mathcal{G}) \leq e^{C/\epsilon}$.

Note that, for any $\nu \in \mathbb{Z}^n$, we have $\theta_\nu^{\langle s\rangle} = 0$ on a neighborhood of Q^{**}, unless $\nu \in \mathcal{G}$. (To see this, we recall that $\operatorname{supp} \theta_\nu^{\langle s\rangle} \subset (Q_\nu^{\langle s\rangle})^*$; see Section 10.)

Consequently, the $\theta_\nu^{\langle s\rangle}, \nu \in \mathcal{G}$, have the following properties.

(7) $\theta_\nu^{\langle s\rangle} \in C^m(\mathbb{R}^n)$, $\theta_\nu^{\langle s\rangle} \geq 0$ on $\mathbb{R}^n$, $\operatorname{supp}\theta_\nu^{\langle s\rangle} \subset [(Q_\nu^{\langle s\rangle})^*]^{\mathrm{int}}$, $|\partial^\alpha \theta_\nu^{\langle s\rangle}| \leq C\,\epsilon^{|\alpha|}$ on $\mathbb{R}^n$ for $|\alpha| \leq m$; and also

(8) $\sum_{\nu\in\mathcal{G}} \theta_\nu^{\langle s\rangle} \leq 1$ on $\mathbb{R}^n$, and $\sum_{\nu\in\mathcal{G}} \theta_\nu^{\langle s\rangle} = 1$ on Q^{**}.

Moreover, we can answer queries as follows.

(9) Given $\underline{x} \in \mathbb{R}^n$ and $\nu \in \mathcal{G}$, we can compute $J_{\underline{x}}(\theta_\nu^{\langle s\rangle})$ with work and storage at most C.

Note also that

(10) Given $\underline{x} \in \mathbb{R}^n$, we can compute the set of $\nu \in \mathcal{G}$ such that $\underline{x} \in (Q_\nu^{\langle s\rangle})^*$. This computation takes work and storage at most C. There are at most C such ν.

For each $\nu \in \mathcal{G}$, we define

(11) $S_\nu = S \cap (Q_\nu^{\langle s\rangle})^{**}$.

We will show that the data

(12) $(\epsilon, Q_\nu^{\langle s\rangle}, \vec{P}|_{S_\nu})$

satisfy the assumptions of Algorithm 17.1, for each $\nu \in \mathcal{G}$.

To see this, fix $\nu \in \mathcal{G}$. We know that ϵ is less than a small enough controlled constant. Also, thanks to (4), the sidelength of $Q_\nu^{\langle s\rangle}$ satisfies

$$e^{-1/(2\epsilon)} \leq \delta_{Q_\nu^{\langle s\rangle}} \leq c^{\#}\epsilon^{-1}. \quad \text{(Recall that } \delta_{Q_\nu^{\langle s\rangle}} = 2^s.)$$

Next, note that $S_\nu \subset (Q_\nu^{\langle s\rangle})^{**}$, thanks to (11).

Finally, for any two distinct points $y, y' \in S_\nu$, we learn from (1), (3), (4) that

$$|y - y'| \geq \epsilon^2 e^{-2/\epsilon}\,\delta_Q > \epsilon^2 e^{-2/\epsilon}\,\delta_{Q_\nu^{\langle s\rangle}}.$$

This completes our verification of the assumptions of Algorithm 17.1 for the data (12).

For each $\nu \in \mathcal{G}$, let $\mathcal{N}_\epsilon(\vec{P}|_{S_\nu})$ and $F_\nu \in C^m(\mathbb{R}^n)$ be the number and function computed by applying Algorithm 17.1 to the data (12). Then the following hold.

(13) F_ν agrees with $\vec{P}|_{S_\nu}$ for each $\nu \in \mathcal{G}$.

(14) $\| F_\nu \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}|_{S_\nu})$ for each $\nu \in \mathcal{G}$.

(15) $\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) \leq (1 + C\epsilon) \cdot \| (\vec{P}|_{S_\nu}) \|_{C^m(\mathbb{R}^n)}$ for each $\nu \in \mathcal{G}$.

(16) For each $\nu \in \mathcal{G}$, the storage and the one-time work to compute $\mathcal{N}_\epsilon(\vec{P}|_{S_\nu})$ and F_ν (given the data (12)) are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C.

We now define $\mathcal{N}_\epsilon(\vec{P})$ and F. We set

(17) $\mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) : \nu \in \mathcal{G}\}$

and we define

(18) $F = \sum_{\nu \in \mathcal{G}} \theta_\nu^{\langle s \rangle} \cdot F_\nu \in C^m(\mathbb{R}^n)$.

Let us check that (A), (B), (C) hold for our $\mathcal{N}_\epsilon(\vec{P})$ and F. We begin with (A). Fix $y \in S$. We have $y \in Q^{**}$ by (2), hence

(19) $\sum_{\nu \in \mathcal{G}} J_y(\theta_\nu^{\langle s \rangle}) = 1$,

by (8). For each $\nu \in \mathcal{G}$ such that $y \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$, we have $y \in (Q_\nu^{\langle s \rangle})^*$ by (7), hence $y \in S_\nu$ by (11), and therefore $J_y(F_\nu) = P^y$ by (13). Thus,

(20) $J_y(F_\nu) = P^y$ for each $\nu \in \mathcal{G}$ such that $y \in \operatorname{supp}(\theta_\nu^{\langle s \rangle})$.

From (18), (19), (20), we see that

$$J_y(F) = \sum_{\nu \in \mathcal{G}} J_y(\theta_\nu^{\langle s \rangle}) \odot_y J_y(F_\nu) = \sum_{\nu \in \mathcal{G}} J_y(\theta_\nu^{\langle s \rangle}) \odot_y P^y = P^y,$$

proving (A).

We pass to (B). We apply Lemma LGPU from Section 5, taking

(21)
$$\begin{aligned} A &= \text{Large enough controlled constant,} \\ \hat{Q}_\nu &= Q_\nu^{\langle s \rangle} \text{ for each } \nu \in \mathcal{G}, \\ \hat{\delta} &= 2^s, \\ M &= (1 + C\epsilon) \cdot \max\{\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) : \nu \in \mathcal{G}\}. \end{aligned}$$

(The hypotheses of that lemma hold for the data (21), thanks to (4), (7), (8) and (14).) From Lemma LGPU, we learn that the function F in (18) satisfies

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C'\epsilon) \cdot \max\{\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) : \nu \in \mathcal{G}\}.$$

Together with (17), this proves conclusion (B).

Next, we establish (C). From (15), we have

$$\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)} \quad \text{for each } \nu \in \mathcal{G}.$$

Consequently, (C) follows trivially from (17). Thus, (A), (B), (C) hold for our $\mathcal{N}_\epsilon(\vec{P})$ and F.

We turn to the computation of $\mathcal{N}_\epsilon(\vec{P})$ and F. The one-time work is as follows.

Step 1: Compute $s \in \mathbb{Z}$ satisfying (4), and then compute the $\nu_i^{\min}$ and $\nu_i^{\max}$ $(i = 1, \ldots, n)$ as in (5).

Step 2: For each $\nu \in \mathcal{G}$, we compute $Q_\nu^{\langle s \rangle}$, $S_\nu = S \cap (Q_\nu^{\langle s \rangle})^{**}$, and $\vec{P}|_{S_\nu}$.

Step 3: For each $\nu \in \mathcal{G}$, we perform the one-time work of Algorithm 17.1 for the data $(\epsilon, Q_\nu^{\langle s \rangle}, \vec{P}|_{S_\nu})$. This produces the number $\mathcal{N}_\epsilon(\vec{P}|_{S_\nu})$ for each $\nu \in \mathcal{G}$, and prepares us to answer queries on $J_{\underline{x}}(F_\nu)$ for any given $\underline{x} \in \mathbb{R}^n$ and $\nu \in \mathcal{G}$.

Step 4: We set $\mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) : \nu \in \mathcal{G}\}$.

Given $\underline{x} \in \mathbb{R}^n$, the query algorithm proceeds as follows

Step Q1: Find $\mathcal{G}(\underline{x}) = \{\nu \in \mathcal{G} : \underline{x} \in (Q_\nu^{\langle s \rangle})^*\}$.

Step Q2: For each $\nu \in \mathcal{G}(\underline{x})$, compute $J_{\underline{x}}(\theta_\nu^{\langle s \rangle})$ and $J_{\underline{x}}(F_\nu)$.

Step Q3: Return the polynomial $J_{\underline{x}}(F) = \sum_{\nu \in \mathcal{G}(\underline{x})} J_{\underline{x}}(\theta_\nu^{\langle s \rangle}) \odot_{\underline{x}} J_{\underline{x}}(F_\nu)$.

Since $J_{\underline{x}}(\theta_\nu^{\langle s \rangle}) = 0$ for $\nu \in \mathcal{G} \setminus \mathcal{G}(\underline{x})$, our query algorithm correctly calculates $J_{\underline{x}}(F)$, with F as in (18).

Also, we note that $\#(S) \leq e^{C/\epsilon}$, thanks to (2) and (3). Consequently, (6) and (16) show that the one-time work and storage required for our algorithm are at most $\exp(C/\epsilon)$. From (9), (10), (16), we see that our query algorithm requires work at most C.

This completes our explanation of Algorithm 18.1.

For future reference, we record a few remarks on the inner working of Algorithm 18.1.

Proposition 18.1. *Let $\epsilon, Q, \vec{P} = (P^y)_{y \in S}$ be as assumed in Algorithm* 18.1. *Then the Algorithm* 18.1 *performs as follows*

(a) *It computes an integer s and a finite subset $\mathcal{G} \subset \mathbb{Z}^n$, using only ϵ, Q (and not using $\vec{P}$). The set $\mathcal{G}$ satisfies $\#(\mathcal{G}) \leq \exp(C/\epsilon)$. The work and storage used to compute $s, \mathcal{G}$ are at most $\exp(C/\epsilon)$.*

(b) *For each $\nu \in \mathcal{G}$, it computes the set $S_\nu = S \cap (Q_\nu^{\langle s \rangle})^{**}$, with $Q_\nu^{\langle s \rangle}$ as in Section* 10. *The work and storage to compute and store all the sets $S_\nu (\nu \in \mathcal{G})$ are at most $\exp(C/\epsilon)$.*

(c) *For each $\nu \in \mathcal{G}$, it applies Algorithm* 17.1 *to the input data $\epsilon, Q_\nu^{\langle s \rangle}, \vec{P}|_{S_\nu}$, to compute a number $\mathcal{N}_\epsilon(\vec{P}|_{S_\nu})$. (In particular, these input data are as assumed in Algorithm* 17.1.)

(d) *It returns the number $\mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_\epsilon(\vec{P}|_{S_\nu}) : \nu \in \mathcal{G}\}$.*

Proof. Obvious from our explanation of Algorithm 18.1. ∎

19. Extending a Whitney Field from a Testing Set IV

In this section, we give an analogue of Algorithms 16.1, 17.1 and 18.1 for the case of a huge cube Q.

Algorithm 19.1. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube Q whose sidelength satisfies*

(1) $\delta_Q \geq e^{5/\epsilon}$;

and given a Whitney field $\vec{P} = (P^y)_{y \in S}$, where the set S is assumed to satisfy

(2) $S \subset Q^{**}$, *and*

(3) $|y - y'| \geq \epsilon^2 \, e^{-2/\epsilon} \delta_Q$ *for any two distinct points $y, y' \in S$;*

we compute a number $\mathcal{N}_\epsilon(\vec{P})$ and a function $F \in C^m(\mathbb{R}^n)$, such that

(A) F *agrees with $\vec{P}$;*

(B) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$; *and*

(C) $\mathcal{N}_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C.

Explanation: By a partition of unity, we reduce matters to Algorithm 15.1. Our discussion is close to that of Section 18. Fix an integer s, such that

(4) $\frac{1}{10} e^{-3/\epsilon} \, \delta_Q \leq 2^s \leq e^{-3/\epsilon} \, \delta_Q$.

Recall from Section 10 the dyadic cubes $Q_\nu^{\langle s \rangle}$ and cutoff functions $\theta_\nu^{\langle s \rangle}$, for $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$.

Thanks to (4), the cube Q^{***} is partitioned into dyadic subcubes $Q_\nu^{\langle s \rangle}$, for $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$ varying over the set

(5) $\mathcal{G} = \{(\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n : \nu_i^{\min} \leq \nu_i \leq \nu_i^{\max} \text{ for each } i\}$, for suitable $\nu_i^{\min}, \nu_i^{\max} (i = 1, \ldots, n)$.

Also thanks to (4), we have

(6) $\#(\mathcal{G}) \leq e^{C/\epsilon}$.

Note that, for any $\nu \in \mathbb{Z}^n$, we have $\theta_\nu^{\langle s \rangle} = 0$ on a neighborhood of Q^{**}, unless $\nu \in \mathcal{G}$. (To see this, recall that $\operatorname{supp} \theta_\nu^{\langle s \rangle} \subset (Q_\nu^{\langle s \rangle})^*$; see Section 10.)

Consequently, the $\theta_\nu^{\langle s\rangle}$ ($\nu \in \mathbb{Z}^n$) have the following properties.

(7) $\theta_\nu^{\langle s\rangle} \in C^m(\mathbb{R}^n)$, $\theta_\nu^{\langle s\rangle} \geq 0$ on $\mathbb{R}^n$, $\operatorname{supp} \theta_\nu^{\langle s\rangle} \subset [(Q_\nu^{\langle s\rangle})^*]^{\mathrm{int}}$;

(8) $|\partial^\alpha \theta_\nu^{\langle s\rangle}| \leq C \cdot (e^{-3/\epsilon}\delta_Q)^{-|\alpha|}$ on $\mathbb{R}^n$, for $|\alpha| \leq m$;

(9) $\sum_{\nu\in\mathbb{Z}^n} \theta_\nu^{\langle s\rangle} = 1$ on $\mathbb{R}^n$; and

(10) $\sum_{\nu\in\mathcal{G}} \theta_\nu^{\langle s\rangle} = 1$ on Q^{**}.

Moreover, we can answer queries as follows.

(11) Given $\underline{x} \in \mathbb{R}^n$ and $\nu \in \mathbb{Z}^n$, we can compute $J_{\underline{x}}(\theta_\nu^{\langle s\rangle})$ with work and storage at most C.

(12) Given $\underline{x} \in \mathbb{R}^n$, we can compute $\{\nu \in \mathcal{G} : \underline{x} \in (Q_\nu^{\langle s\rangle})^*\}$ with work and storage at most C. In particular, there are at most C such $\nu \in \mathcal{G}$.

For each $\nu \in \mathcal{G}$, we define

(13) $S_\nu = S \cap (Q_\nu^{\langle s\rangle})^{**}$.

We will check that

(14) $\#(S_\nu) \leq 1$ for each $\nu \in \mathcal{G}$.

To see this, we fix $\nu \in \mathcal{G}$, and suppose that y, y' are two distinct points of S_ν. We will derive a contradiction. In fact, since y, y' both belong to $(Q_\nu^{\langle s\rangle})^{**}$, we have

(15) $|y - y'| \leq C\delta_{Q_\nu^{\langle s\rangle}} = C \cdot 2^s$.

On the other hand, (3) and (4) yield

(16) $|y-y'| \geq \epsilon^2 e^{-2/\epsilon} \cdot \delta_Q = (\epsilon^2 \cdot e^{1/\epsilon}) \cdot (e^{-3/\epsilon}\,\delta_Q) \geq (\epsilon^2 e^{1/\epsilon}) \cdot (c \cdot 2^s)$.

Since (15) contradicts (16), there cannot be two distinct points in S_ν, completing the proof of (14).

For each $\nu \in \mathcal{G}$, we now define $\mathcal{N}_{\epsilon,\nu}(\vec{P}) \geq 0$ and $F_\nu \in C^m(\mathbb{R}^n)$, as follows.

Let $\nu \in \mathcal{G}$. Then S_ν is empty or a singleton, by (14). If S_ν is empty, then we set $\mathcal{N}_{\epsilon,\nu}(\vec{P}) = 0$ and $F_\nu = 0$.

If $\#(S_\nu) = 1$, then we define $\mathcal{N}_{\epsilon,\nu}(\vec{P})$ and $F_\nu \in C^m(\mathbb{R}^n)$ to be the number and the function computed by Algorithm 15.1 applied to the data $(\epsilon, \vec{P}|_{S_\nu})$. (Note that the assumptions of Algorithm 15.1 hold here, since ϵ is less than a small enough controlled constant, and $\vec{P}|_{S\nu}$ is a Whitney field on a singleton.)

In either case ($\#(S_\nu) = 0$ or $\#(S_\nu) = 1$), the following properties hold.

(17) F_ν agrees with $\vec{P}$ on S_ν, for each $\nu \in \mathcal{G}$.

(18) $\| F_\nu \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_{\epsilon,\nu}(\vec{P})$, for each $\nu \in \mathcal{G}$.

(19) $\mathcal{N}_{\epsilon,\nu}(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$ for each $\nu \in \mathcal{G}$.

(20) Given $\nu \in \mathcal{G}$, and given the data $(\epsilon, \vec{P}|_{S_\nu})$, the storage and one-time work to compute $\mathcal{N}_{\epsilon,\nu}(\vec{P})$ and F_ν are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C.

In fact, if S_ν is non-empty, then (17)...(20) are immediate from the basic assertions of Algorithm 15.1. If S_ν is empty, then (17)...(20) hold trivially.

We now define $\mathcal{N}_\epsilon(\vec{P})$ and F. We set

(21) $\mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G}\}$,

and we define

(22) $F = \sum_{\nu \in \mathcal{G}} \theta_\nu^{\langle s \rangle} \cdot F_\nu \in C^m(\mathbb{R}^n)$.

Let us check that (A), (B), (C) hold for the above $\mathcal{N}_\epsilon(\vec{P})$ and F. We begin with (A). Fix $y \in S$. We have $y \in Q^{**}$ by (2), hence

(23) $\sum_{\nu \in \mathcal{G}} J_y(\theta_\nu^{\langle s \rangle}) = 1$,

by (10).

For each $\nu \in \mathcal{G}$ such that $y \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$, we have $y \in (Q_\nu^{\langle s \rangle})^*$ by (7), hence $y \in S_\nu$ by (13), and therefore $J_y(F_\nu) = P^y$ by (17). Thus,

(24) $J_y(F_\nu) = P^y$ for each $\nu \in \mathcal{G}$ such that $y \in \operatorname{supp} \theta_\nu^{\langle s \rangle}$.

From (22), (23), (24), we see that

$$J_y(F) = \sum_{\nu \in \mathcal{G}} J_y(\theta_\nu^{\langle s \rangle}) \odot_y J_y(F_\nu) = \sum_{\nu \in \mathcal{G}} J_y(\theta_\nu^{\langle s \rangle}) \odot_y P^y = P^y,$$

proving (A).

We pass to (B). Our plan is to apply Lemma LGPU to the following data.

(25)
$$\left[\begin{array}{l} \hat{\delta} = 2^s \\ \hat{\epsilon} = 2^{-s} \text{ in place of } \epsilon \text{ in Lemma LGPU.} \\ A = \text{ Large enough controlled constant.} \\ M = (1 + C\epsilon) \cdot \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G}\}. \\ \text{The functions } \theta_\nu^{\langle s \rangle} \text{ and } F_\nu,\ \nu \in \mathcal{G} \quad . \\ \text{The cubes } Q_\nu^{\langle s \rangle},\ \nu \in \mathcal{G} \quad . \end{array}\right]$$

Let us check the hypotheses of Lemma LGPU for the data (25). We have $0 < \hat{\epsilon} < 1$, by (1) and (4). Also, $A \geq 1$ and $M \geq 0$. Moreover, $A^{-1}\hat{\epsilon}^{-1} \leq \hat{\delta} \leq \hat{\epsilon}^{-1}$; in fact, $\hat{\delta} = \hat{\epsilon}^{-1}$.

The desired properties of the $\theta_\nu^{\langle s \rangle}$ are immediate from (7), (8), (9). Finally, the desired properties of the F_ν are immediate from (18), and from our definition of M in (25).

Thus, the hypotheses of Lemma LGPU all hold for the data (25).

Applying that lemma, we see that the function F in (22) satisfies

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C'\epsilon) \cdot \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G}\}.$$

Together with (21), this yields conclusion (B).

Note also that conclusion (C) follows trivially from (19) and (21). Thus, (A), (B) and (C) hold for our $\mathcal{N}_\epsilon(\vec{P})$ and F.

We turn to the computation of our $\mathcal{N}_\epsilon(\vec{P})$ and F. The one-time work is as follows.

Step 1: We compute $s \in \mathbb{Z}$ satisfying (4), and then we compute $\nu_i^{\min}, \nu_i^{\max}$ $(i = 1, \ldots, n)$ as in (5).

Step 2: For each $\nu \in \mathcal{G}$, we compute $S_\nu = S \cap (Q_\nu^{\langle s \rangle})^{**}$ and $\vec{P}|_{S_\nu}$.

Step 3: For each $\nu \in \mathcal{G}$, we set $\Lambda(\nu) = 1$ if S_ν is non-empty, $\Lambda(\nu) = 0$ otherwise.

Step 4: For each $\nu \in \mathcal{G}$ such that $\Lambda(\nu) = 1$, we perform the one-time work of Algorithm 15.1 for the data $(\epsilon, \vec{P}|s_\nu)$. This produces a number $\mathcal{N}_{\epsilon,\nu}(\vec{P})$, and allows us to answer queries regarding a function F_ν.

Step 5: We set $\mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P})$: all $\nu \in \mathcal{G}$ such that $\Lambda(\nu) = 1\}$.

Given $\underline{x} \in \mathbb{R}^n$, our query algorithm proceeds as follows.

Step Q1: Find $\mathcal{G}(\underline{x}) = \{\nu \in \mathcal{G} : \underline{x} \in (Q_\nu^{\langle s \rangle})^*$ and $\Lambda(\nu) = 1\}$.

Step Q2: For each $\nu \in \mathcal{G}(\underline{x})$, compute $J_{\underline{x}}(\theta_\nu^{\langle s \rangle})$ and $J_{\underline{x}}(F_\nu)$.

Step Q3: Return $J_{\underline{x}}(F) = \sum_{\nu \in \mathcal{G}(\underline{x})} J_{\underline{x}}(\theta_\nu^{\langle s \rangle}) \odot_{\underline{x}} J_{\underline{x}}(F_\nu)$.

Since $J_{\underline{x}}(\theta_\nu^{\langle s \rangle}) = 0$ for $\nu \in \mathcal{G}$, $\underline{x} \notin (Q_\nu^{\langle s \rangle})^*$, and since $F_\nu = 0$ for $\nu \in \mathcal{G}$, $\Lambda(\nu) = 0$, it follows that our query algorithm correctly computes $J_{\underline{x}}(F)$, with F as in (22).

Also, we note that $\#(S) \leq e^{C/\epsilon}$, thanks to (2) and (3). Consequently, (6) and (20) show that the storage and one-time work of our algorithm are at most $\exp(C/\epsilon)$. From (11), (12) and (20), we see that the work to answer a query is at most C. This completes our explanation of Algorithm 19.1.

For future reference, we give a simple proposition concerning the output of Algorithm 19.1.

Proposition 19.1. *Let* $\epsilon, Q, \vec{P} = (P^y)_{y\in S}$ *be as assumed in Algorithm* 19.1. *Then the following hold.*

(a) $\#(S) \leq \exp(C/\epsilon)$.

(b) *For each* $y_0 \in S$, *the input data* $\epsilon, \vec{P}|_{\{y_0\}}$ *are as assumed in Algorithm* 15.1.

(c) *For each* $y_0 \in S$, *let* $\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})$ *be the number computed by applying Algorithm* 15.1 *to the input data* $\epsilon, \vec{P}|_{\{y_0\}}$.

Then the number $\mathcal{N}_\epsilon(\vec{P})$ *returned by Algorithm* 15.1 *is equal to*

$$\max\{\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}) : y_0 \in S\}.$$

Proof. Assertion (a) follows trivially from (2) and (3). Assertion (b) holds, since the only assumptions for Algorithm 15.1. are that ϵ is less than a small enough controlled constant, and that the input Whitney field is defined on a singleton.

We turn to assertion (c). We refer to our explanation of Algorithm 19.1.

For any $\nu \in \mathcal{G}$ such that $\wedge(\nu) = 1$, the set S_ν is a singleton, and thus $\mathcal{N}_{\epsilon,\nu}(\vec{P})$ is among the numbers $\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})$, $y_0 \in S$. Consequently,

$$\max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G},\, \wedge(\nu) = 1\} \leq \max\{\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}) : y_0 \in S\}. \tag{26}$$

On the other hand, let $y_0 \in S$. Then $y_0 \in Q^{**} \subset Q^{***}$, hence $y_0 \in Q_{\bar{\nu}}^{\langle s\rangle}$ for some $\bar{\nu} \in \mathcal{G}$. Fix such a $\bar{\nu}$. Then since $y_0 \in S \cap (Q_{\bar{\nu}}^{\langle s\rangle})^{**} = S_{\bar{\nu}}$, we have $\wedge(\bar{\nu}) = 1$. Recalling (14), we conclude that $S_{\bar{\nu}} = \{y_0\}$. Consequently, in Step 4, we set $\mathcal{N}_{\epsilon,\bar{\nu}}(\vec{P}) = \mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})$. Hence,

$$\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}) \leq \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G}\,,\, \wedge(\nu) = 1\}.$$

Since $y_0 \in S$ was arbitrary, it follows that

$$\max\{\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}) : y_0 \in S\} \leq \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G}\,, \wedge(\nu) = 1\}.$$

Together with (26), this yields the equality

$$\max\{\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}) : y_0 \in S\} = \max\{\mathcal{N}_{\epsilon,\nu}(\vec{P}) : \nu \in \mathcal{G}, \wedge(\nu) = 1\}. \tag{27}$$

The desired conclusion (c) now follows from (27) and Step 5 above. ∎

20. Extending a Whitney Field from a Testing Set V

In this section, we combine the results of the last several sections.

Algorithm 20.1. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube Q; and given a Whitney field $\vec{P} = (P^y)_{y \in S}$, where the set S is assumed to satisfy*

(1) $S \subset Q^{**}$, *and*

(2) $|y - y'| > \epsilon^2 e^{-2/\epsilon} \delta_Q$ *for any two distinct points $y, y' \in S$;*

we compute a number $\mathcal{N}_\epsilon(\vec{P}) \geq 0$ and a function $F \in C^m(\mathbb{R}^n)$, such that

(A) F *agrees with $\vec{P}$;*

(B) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$; *and*

(C) $\mathcal{N}_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

The storage and the one-time work needed for the computation are at most $\exp(C/\epsilon)$, and the work to answer a query is at most C.

Explanation: The sidelength δ_Q must satisfy at least one of the following.

Case 1: $\delta_Q \leq e^{-1/(4\epsilon)}$.

Case 2: $e^{-1/(2\epsilon)} \leq \delta_Q \leq c^\# \epsilon^{-1}$, with $c^\#$ as in Algorithm 17.1.

Case 3: $\frac{1}{2} c^\# \epsilon^{-1} \leq \delta_Q \leq e^{10/\epsilon}$, with $c^\#$ as in Algorithm 17.1.

Case 4: $\delta_Q \geq e^{5/\epsilon}$.

With work and storage at most C, we can identify one of the above cases that holds for our given Q. In cases 1,2,3,4, we can find $\mathcal{N}_\epsilon(\vec{P})$ and F satisfying (A), (B), (C) by applying Algorithm 16.1, 17.1, 18.1 or 19.1, respectively. The work and storage of our computation are as asserted above.

This completes our explanation of Algorithm 20.1.

21. The Main Extension Algorithm

In the next several sections, we present the following algorithm.

Algorithm 21.1. ("Main Extension Algorithm"): *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a Whitney field $\vec{P} = (P^x)_{x \in E}$, with $\#(E) = N$, $2 \leq N < \infty$; we compute a real number $\mathcal{N}_\epsilon(\vec{P}) \geq 0$ and a function $F \in C^m(\mathbb{R}^n)$, such that:*

(A) F *agrees with* $\vec{P}$*;*

(B) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$*; and*

(C) $\mathcal{N}_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

The storage and one-time work of the algorithm are at most $\exp(C/\epsilon)N$ *and* $\exp(C/\epsilon)\ N \log N$*, respectively.*

The work to answer a query is at most $C \cdot \log(N/\epsilon)$.

Theorem 1 (from the Introduction) is an obvious consequence of Algorithms 21.1 and 15.1.

The explanation of the above algorithm occurs in the next three sections.

In Section 22, we present the one-time work of Algorithm 21.1. This one-time work computes the number $\mathcal{N}_\epsilon(\vec{P})$ and allows us to answer queries. In Section 23, we define a function $F \in C^m(\mathbb{R}^n)$, and we prove that our $\mathcal{N}_\epsilon(\vec{P})$ and F satisfy conditions (A), (B), (C) above. Finally, in Section 24, we present the query algorithm that computes $J_{\underline{x}}(F)$ for any given $\underline{x} \in \mathbb{R}^n$.

22. The One-Time Work

In this section, we carry out the one-time work of Algorithm 21.1. We let $\kappa > 0$ be a small enough controlled constant. Let $\epsilon, \vec{P}, E, N$ be as in Section 21. We proceed as follows.

Step 1: We perform the one-time work of the BBD Tree, and we carry out Algorithm WSPD with input (E, κ). (See Section 4.) Let $(x'_\nu, x''_\nu) \in E \times E$ $(\nu = 1, \ldots, \nu_{\max})$ be the "representatives" produced by Algorithm WSPD, as explained in Section 4.

This step takes work at most $CN \log N$ in space CN, since κ is a controlled constant. Also, we have $\nu_{\max} \leq CN$.

Step 2: We compute the smallest real number $\hat{M} \geq 0$ such that

(1) $|\partial^\alpha P^x| \leq \hat{M}$ for $|\alpha| \leq m$, $x \in E$; and

(2) $|\partial^\alpha (P^{x'_\nu} - P^{x''_\nu})(x'_\nu)| \leq \hat{M} \cdot |x'_\nu - x''_\nu|^{m-|\alpha|}$ for $|\alpha| \leq m - 1$, $1 \leq \nu \leq \nu_{\max}$.

Note that Lemma 8.2 assures us that

(3) $|\partial^\alpha (P^x - P^y)(x)| \leq C\hat{M} \cdot |x - y|^{m-|\alpha|}$ for $|\alpha| \leq m$, $x, y \in E$, $x \neq y$.

The work involved in Step 2 is at most CN, and the storage needed (once we have already stored the input $\vec{P}$ and the representatives x'_ν, x''_ν $(\nu = 1, \ldots, \nu_{\max})$) is at most C.

Thanks to (3), we will be able to prove assumption (1) in Section 6 for a suitable M to be picked later.

In addition to (3), we will use the following estimate.

(4) $\hat{M} \le C^* \parallel \vec{P} \parallel_{C^m(\mathbb{R}^n)}$, for a large enough controlled constant C^*.

To prove (4), let M^+ be any real number greater than $\parallel \vec{P} \parallel_{C^m(\mathbb{R}^n)}$. Then, by definition of the C^m-norm of a Whitney field, there exists $F^+ \in C^m(\mathbb{R}^n)$ such that F^+ agrees with $\vec{P}$, and $\parallel F^+ \parallel_{C^m(\mathbb{R}^n)} \le M^+$.

The Bounded Distortion Property gives

(5) $|\partial^\alpha F^+(x)| \le CM^+$ for $|\alpha| \le m$, $x \in \mathbb{R}^n$,

which in turn yields

(6) $|\partial^\alpha (J_x(F^+) - J_y(F^+))(x)| \le C'M^+|x-y|^{m-|\alpha|}$ for $|\alpha| \le m$, $x, y \in \mathbb{R}^n$, $x \neq y$,

by Taylor's theorem.

Since F^+ agrees with $\vec{P}$, (5) and (6) imply

$$|\partial^\alpha P^x(x)| \le CM^+ \quad \text{for } |\alpha| \le m,\ x \in E,$$

and

$$|\partial^\alpha (P^x - P^y)(x)| \le C'M^+|x-y|^{m-|\alpha|} \quad \text{for } |\alpha| \le m,\ x, y \in E,\ x \neq y.$$

Comparing these estimates with the definition of $\hat{M}$, we learn that $\hat{M} \le C^*M^+$. Since M^+ is any real number greater than $\parallel \vec{P} \parallel_{C^m(\mathbb{R}^n)}$, the proof of (4) is complete.

For this section and the next, we fix C^* as in (4).

Step 3: We perform the one-time work of Algorithm 9.2 ("Compute-Regularized-Distance").

This requires work at most $CN \log N$ in space CN. After this step, we can answer queries regarding a certain function $\delta(\cdot) \in C^m_{loc}(\mathbb{R}^n \setminus E)$.

The function $\delta(\cdot)$ satisfies

(7) $c\,\mathrm{dist}\,(x, E) \le \delta(x) \le C\,\mathrm{dist}(x, E)$ for $x \in \mathbb{R}^n \setminus E$, and

(8) $|\partial^\alpha \delta(x)| \le C \cdot (\delta(x))^{1-|\alpha|}$ for $|\alpha| \le m$, $x \in \mathbb{R}^n \setminus E$.

The query algorithm regarding $\delta(\cdot)$ performs as follows.

(9) Given $\underline{x} \in \mathbb{R}^n \setminus E$, we can compute $J_{\underline{x}}(\delta(\cdot))$ with work at most $C \log N$, in space CN.

For (7), (8), (9), see estimates (8), (9) in Section 9, as well as Algorithm 9.2.

Note that our present estimates (7) and (8) give us assumptions (2) and (3) in Section 6.

Step 4: We carry out Algorithm 11.2 ("Find-Interesting-Cubes"), to compute a list $Q^{(1)}, \ldots, Q^{(L)}$ of dyadic cubes, with the following properties.

(10) The cubes $Q^{(1)}, \ldots, Q^{(L)}$ are all distinct.

(11) $L \leq (C/\epsilon) \cdot N$.

(12) For any given dyadic cube Q, the set $S(Q)$ computed from ϵ, E, Q by Algorithm 11.1 has cardinality ≥ 2 if and only if Q is one of the $Q^{(\lambda)}$, $\lambda = 1, \ldots, L$.

The work of Step 4 is at most $\exp(C/\epsilon) \cdot N \log N$, and the storage needed is at most $\frac{C}{\epsilon}N + \exp(C/\epsilon)$.

Step 5: We introduce any convenient ordering on the set of dyadic cubes. (Say, we use lexicographic order in terms of the sidelength and coordinates of the center point of a given cube.) We then sort our list $Q^{(1)}, \ldots, Q^{(L)}$ with respect to that order.

This takes work at most $CL \log L \leq \frac{C'}{\epsilon}N \cdot \log(\frac{C'}{\epsilon}N)$ in space $CL \leq \frac{C'}{\epsilon}N$; see (11). Thus, (10), (11), (12) hold, and we can perform binary searches as follows.

(13) Given a dyadic cube Q, we can decide whether $Q = Q^{(\lambda)}$ for some λ, with work at most $C \log(N/\epsilon)$.

If $Q = Q^{(\lambda)}$ for some λ, then we can compute that λ, again with work at most $C \log(N/\epsilon)$.

Step 6: For each $\lambda = 1, \ldots, L$, we apply Algorithm 11.1 to $\epsilon, E, Q^{(\lambda)}$, to compute a finite set $S^{(\lambda)}$, with the following properties.

(14) $S^{(\lambda)} \subset E \cap (Q^{(\lambda)})^{**}$.

(15) $|y - y'| > c\epsilon e^{-2/\epsilon}\, \delta_{Q^{(\lambda)}}$ for any two distinct points $y, y' \in S^{(\lambda)}$.

(16) $\mathrm{dist}\,(y, S^{(\lambda)}) \leq C\epsilon e^{-2/\epsilon}\, \delta_{Q^{(\lambda)}}$ for any $y \in E \cap (Q^{(\lambda)})^{**}$.

(17) $\#(S^{(\lambda)}) \geq 2$, as we see from (12).

The work to compute a single $S^{(\lambda)}$ is at most $\exp(C/\epsilon) \cdot \log N$, and the storage needed is at most $CN + \exp(C/\epsilon)$.

As we loop over λ, we can re-use the above storage, but we want to store all the sets $S^{(1)}, \ldots, S^{(L)}$.

From (14), (15) we see that $\#(S^{(\lambda)}) \leq \exp(C/\epsilon)$ for each λ.

Along with (11), the above remarks show that Step 6 consumes altogether at most $\exp(C'/\epsilon)N \log N$ work, and at most $\exp(C'/\epsilon)N$ storage.

The sets $S^{(\lambda)}$ computed in this step will be used as the sets S_ν^ℓ in Section 6, for certain ℓ and ν. In particular, assumptions (11) and (12) in Section 6 will be proven using (14) and (16) here.

Step 7: For each $\lambda = 1, \ldots, L$, we perform the one-time work of Algorithm 20.1 for the inputs $\epsilon, Q^{(\lambda)}, \vec{P}|_{S^{(\lambda)}}$.

Note that these inputs satisfy the assumptions of Algorithm 20.1 since $\epsilon > 0$ is less than a small enough controlled constant, and thanks to (14) and (15).

From this step, we obtain for each λ a number $\mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \geq 0$, and we are able to answer queries as follows, regarding a function $F^{(\lambda)} \in C^m(\mathbb{R}^n)$.

(18) Given $\lambda (1 \leq \lambda \leq L)$, and given $\underline{x} \in \mathbb{R}^n$, we can compute $J_{\underline{x}}(F^{(\lambda)})$ with work at most C.

The number $\mathcal{N}_\epsilon^{(\lambda)}(\vec{P})$ and the function $F^{(\lambda)}$ satisfy the following conditions.

(19) $F^{(\lambda)}$ agrees with $\vec{P}$ on $S^{(\lambda)}$.

(20) $$\| F^{(\lambda)} \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon^{(\lambda)}(\vec{P}).$$

(21) $$\mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \leq (1 + C\epsilon) \cdot \| (\vec{P}|_{S^{(\lambda)}}) \|_{C^m(\mathbb{R}^n)}.$$

For each λ, Step 7 requires work and storage at most $\exp(C/\epsilon)$. Hence, by (11), the total work and storage of Step 7 are at most $\exp(C'/\epsilon) \cdot N$.

The functions $F^{(\lambda)}$ computed in this step will be used as the functions F_ν^ℓ in Section 6, for certain ℓ and ν. Compare our present (19), (20) with assumptions (13), (14) in Section 6.

Step 8: For each $y_0 \in E$, we carry out the one-time work of Algorithm 15.1 for the inputs $\epsilon, \vec{P}|_{\{y_0\}}$.

After this step, we have computed numbers $\mathcal{N}_\epsilon^{(y_0)}(\vec{P}) \geq 0$, and we can answer queries regarding functions $F^{(y_0)} \in C^m(\mathbb{R}^n)$. These numbers and functions have the following properties, for each $y_0 \in E$.

(22) $F^{(y_0)}$ agrees with $\vec{P}$ at y_0.

(23) $$\| F^{(y_0)} \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon^{(y_0)}(\vec{P}).$$

(24) $$\mathcal{N}_\epsilon^{(y_0)}(\vec{P}) \leq (1 + C\epsilon) \cdot \| (\vec{P}|_{\{y_0\}}) \|_{C^m(\mathbb{R}^n)}.$$

The relevant query algorithm is as follows.

(25) Given $y_0 \in E$ and $\underline{x} \in \mathbb{R}^n$, we can compute $J_{\underline{x}}(F^{(y_0)})$ with work at most C.

Step 8 requires work and storage at most $\exp(C/\epsilon)$ for each $y_0 \in E$. Hence, the total work and storage required for Step 8 are at most $\exp(C/\epsilon)N$.

The functions $F^{(y_0)}(y_0 \in E)$ will be used as the functions $F^x(x \in E)$ in Section 6.

Compare our present (22) with assumption (15) in Section 6.

Step 9: We compute

$$(26)\quad \mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \text{ (all } \lambda = 1,\dots,L),\ \mathcal{N}_\epsilon^{(y_0)}(\vec{P}) \text{ (all } y_0 \in E),\ \tfrac{\hat{M}}{2C^*}\},$$

where C^* is the controlled constant in estimate (4).

Here, we are simply computing the maximum of a list of numbers that were already computed in Steps 1...8 above. The work of this step is thus at most

$$C \cdot (N + L + 1) \le \frac{C'}{\epsilon} N \quad \text{(see (11))},$$

and we need storage at most C (aside from the storage already used to hold the numbers $\mathcal{N}_\epsilon^{(\lambda)}(\vec{P})$, $\mathcal{N}_\epsilon^{(y_0)}(\vec{P})$, $\hat{M}$).

This completes our description of the one-time work of Algorithm 21.1.

Thanks to our estimates for the work and storage of each particular step, we now see that the one-time work of Algorithm 21.1 is at most $\exp(C/\epsilon)N\log N$, and the storage required is at most $\exp(C/\epsilon)N$. These are as asserted in Section 21.

We have computed the number $\mathcal{N}_\epsilon(\vec{P})$ in Step 9. In the next section, we will define a function $F \in C^m(\mathbb{R}^n)$, and show that $\mathcal{N}_\epsilon(\vec{P})$, F satisfy conditions (A), (B), (C) from Section 21. In Section 24, we give the query algorithm to compute $J_{\underline{x}}(F)$ for any given $\underline{x} \in \mathbb{R}^n$.

23. The Main Extending Function

In this section, we retain the notation and conventions of Sections 21 and 22. We recall the convention that (k.ℓ) refers to equation ℓ in Section k. Our goal here is to define a function $F \in C^m(\mathbb{R}^n)$, which, together with $\mathcal{N}_\epsilon(\vec{P})$ from (22.26), satisfies (A), (B), (C) of Section 21. We apply the Main Patching Lemma from Section 6, along with the results of our one-time work from Section 22. We proceed as follows. We define

(1) $M = (1 + \check{C}\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P})$ for a large enough controlled constant $\check{C}$, with $\mathcal{N}_\epsilon(\vec{P})$ as in (22.26).

From (1) and (22.26), we have $\hat{M} \le 2C^*M$, with $\hat{M}$ as in (22.1) and (22.2). Therefore, estimate (22.3) shows that assumption (6.1) holds here, where we may take A_0 to be a controlled constant.

Let $\delta(\cdot) \in C^m_{loc}(\mathbb{R}^n \setminus E)$ be as in Step 3 in Section 22. Then (22.7) and (22.8) tell us that assumptions (6.2) and (6.3) hold here, where we may take A_1 and A_2 to be controlled constants.

For $\ell \in \mathbb{Z}$, let $\chi_\ell \in C^m(\mathbb{R})$ be as in (10.1), (10.2), (10.3). From those equations and estmates, we learn that assumptions (6.4) and (6.5) hold here, where we may take A_3 to be a controlled constant.

For each $\ell \in \mathbb{Z}$ and $\nu \in \mathbb{Z}^n$, let θ_ν^ℓ and Q_ν^ℓ be as in (10.5)...(10.11). Those results show that assumptions (6.6)...(6.10) hold here, where we may take A_4, A_5, A_6 to be controlled constants.

We recall from (10.4) and the remarks just after it, that

(2) $Q_\nu^\ell = (Q_\nu^{\langle s\rangle})^*$ for a suitable dyadic cube $Q_\nu^{\langle s\rangle}$, where

(3) sidelength $(Q_\nu^{\langle s\rangle}) = 2^s$, and $\frac{1}{32}\,\epsilon^{-1}\cdot\exp((\ell+1)/\epsilon) \le 2^s \le \frac{\epsilon^{-1}}{8}\exp((\ell+1)/\epsilon)$. Thus,

(4) $c\,\exp((\ell-1)/\epsilon) \le \epsilon\, e^{-2/\epsilon}\,\delta_{Q_\nu^{\langle s\rangle}} \le C\,\exp((\ell-1)/\epsilon)$.

Let $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$ be given; and let $Q_\nu^{\langle s\rangle}$ be as in (2), (3). We define

$$S_\nu^\ell = S(Q_\nu^{\langle s\rangle})$$

to be the set obtained by applying Algorithm 11.1 to the input $\epsilon, E, Q_\nu^{\langle s\rangle}$. The following are among the defining properties of Algorithm 11.1.

(5) $S_\nu^\ell \subset E \cap (Q_\nu^{\langle s\rangle})^{**} = E \cap (Q_\nu^\ell)^*$ (see (2)).

(6) $|y - y'| \ge c\epsilon e^{-2/\epsilon}\,\delta_{Q_\nu^{\langle s\rangle}} \ge c'\exp((\ell-1)/\epsilon)$ for any two distinct $y, y' \in S_\nu^\ell$ (see (4)).

(7) $\mathrm{dist}\,(y, S_\nu^\ell) \le C\epsilon e^{-2/\epsilon}\,\delta_{Q_\nu^{\langle s\rangle}} \le C'\exp((\ell-1)/\epsilon)$ for any $y \in E \cap (Q_\nu^\ell)^*$ (see (2) and (4)).

Thus, assumptions (6.11) and (6.12) hold here, where we may take A_7 to be a controlled constant.

Comparing our present definition of S_ν^ℓ with that of $S^{(\lambda)}$ in Step 6 of Section 22, we learn the following.

(8) Let $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$, $1 \le \lambda \le L$. If the cube $Q_\nu^{\langle s\rangle}$ in (2) satisfies $Q_\nu^{\langle s\rangle} = Q^{(\lambda)}$, then $S_\nu^\ell = S^{(\lambda)}$.

Also, comparing our present definition of S_ν^ℓ with (22.12), we learn the following.

(9) Let $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$, and let $Q_\nu^{\langle s\rangle}$ be as in (2). If the cube $Q_\nu^{\langle s\rangle}$ does not appear in the list $Q^{(1)}, \ldots, Q^{(L)}$, then $\#(S_\nu^\ell) \le 1$.

Next, let $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$, and let $Q_\nu^{\langle s \rangle}$ be as in (2). We will define a function $F_\nu^\ell \in C^m(\mathbb{R}^n)$, proceeding by cases as follows.

Case 1: Suppose $Q_\nu^{\langle s \rangle}$ appears in the list $Q^{(1)}, \dots, Q^{(L)}$. Say, $Q_\nu^{\langle s \rangle} = Q^{(\lambda)}$. Then we define

$$F_\nu^\ell = F^{(\lambda)},$$

with $F^{(\lambda)}$ as in (22.19)...(22.21). From (22.19) and (8), we learn that

(10) F_ν^ℓ agrees with $\vec{P}$ on S_ν^ℓ;

moreover, (22.20), (22.26) and (1) show that

(11) $$\| F_\nu^\ell \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \leq (1 + C\epsilon)\, \mathcal{N}_\epsilon(\vec{P}) \leq M,$$

provided we take $\check{C}$ in (1) to be larger than C in (11).

Case 2: Suppose $Q_\nu^{\langle s \rangle}$ does not appear in the list $Q^{(1)}, \dots, Q^{(L)}$.

Then $\#(S_\nu^\ell) \leq 1$, by (9). Thus, either $S_\nu^\ell = \phi$, or else $S_\nu^\ell = \{y_0\}$ for some $y_0 \in E$. (See (5).)

Subcase 2a: If $S_\nu^\ell = \phi$, then we set $F_\nu^\ell = 0$. Trivially,

(12) F_ν^ℓ agrees with $\vec{P}$ on S_ν^ℓ, and

(13) $$\| F_\nu^\ell \|_{C^m(\mathbb{R}^n)} \leq M.$$

Subcase 2b: If $S_\nu^\ell = \{y_0\}$ with $y_0 \in E$, then we set $F_\nu^\ell = F^{(y_0)}$, with $F^{(y_0)}$ as in (22.22)...(22.24). From (22.22), we see that

(14) F_ν^ℓ agrees with $\vec{P}$ on S_ν^ℓ.

Moreover, (22.23), (22.26) and (1) show that

(15) $$\| F_\nu^\ell \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon^{(y_0)}(\vec{P}) \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}) \leq M,$$

provided we take $\check{C}$ in (1) to be larger than C in (15).

We now pick $\check{C}$ to be a controlled constant, large enough to guarantee (11) and (15). Thus, $F_\nu^\ell \in C^m(\mathbb{R}^n)$ is defined for all $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$. From (10)...(15), we see that assumptions (6.13) and (6.14) hold here in all cases.

Next, for each $y_0 \in E$, let $F^{(y_0)} \in C^m(\mathbb{R}^n)$ be as in (22.22)...(22.24). From (22.22), we see that assumption (6.15) holds here. Moreover, let $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$, and assume that $S_\nu^\ell = \{y_0\}$ for some $y_0 \in E$. Then we cannot be in

Case 1 above, since we would then have $\#(S_\nu^\ell) = \#(S^{(\lambda)}) \geq 2$ for some λ, by (8) and (22.17). Evidently, we cannot be in Subcase 2a either. Thus, we are in Subcase 2b, and consequently $F_\nu^\ell = F^{(y_0)}$ by definition. This shows that assumption (6.16) holds here.

We have now shown that (6.1)...(6.16) hold for the $\epsilon, M, \vec{P}$, $\delta(\cdot)$, $\chi_\ell(\cdot)$, θ_ν^ℓ, Q_ν^ℓ, S_ν^ℓ, F_ν^ℓ and $F^{(y_0)}$ given above, where we may take $A_0, \ldots, A_7$ to be controlled constants. Consequently, the Main Patching Lemma from Section 6 applies, and it tells us the following.

We define a function F on $\mathbb{R}^n$, by setting

$$F(x) = (P^x)(x) \text{ for } x \in E, \text{ and} \tag{16}$$

$$F(x) = \sum_{\ell,\nu} \chi_\ell(\epsilon \log \delta(x)) \cdot \theta_\nu^\ell(x) \cdot F_\nu^\ell(x) \text{ for } x \in \mathbb{R}^n \setminus E. \tag{17}$$

Then

$$F \in C^m(\mathbb{R}^n), \tag{18}$$

$$F \text{ agrees with } \vec{P}, \text{ and} \tag{19}$$

$$\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(\vec{P}). \text{ (See (1).)} \tag{20}$$

Thus, the number $\mathcal{N}_\epsilon(\vec{P})$ in (22.26) and the function F in (16), (17), together satisfy (A) and (B) in Section 21. We now show that (C) holds as well.

Estimates (22.21) and (22.24) show at once that

$$\mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)} \text{ for all } \lambda, \tag{21}$$

and that

$$\mathcal{N}_\epsilon^{(y_0)}(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)} \text{ for all } y_0 \in E. \tag{22}$$

Also, from estimate (22.4), we have

$$\frac{\hat{M}}{C^*} \leq \| \vec{P} \|_{C^m(\mathbb{R}^n)}. \tag{23}$$

(Recall that the constant C^* has been fixed as in (22.4); it does not vary from one occurrence to the next.)

From estimates (21), (22), (23) and the definition (22.26) of $\mathcal{N}_\epsilon(\vec{P})$, we see that

$$\mathcal{N}_\epsilon(\vec{P}) \leq (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)},$$

which is the desired conclusion (C).

Thus, we have shown that our $\mathcal{N}_\epsilon(\vec{P})$ and F satisfy conditions (A), (B) and (C) from Section 21.

In the next section, we give the query algorithm to compute $J_{\underline{x}}(F)$ for a given query point $\underline{x} \in \mathbb{R}^n$.

24. The Query Algorithm

In this section, we adopt the notation and assumptions of Sections 21, 22, and 23. Our goal is to give the query algorithm within Algorithm 21.1. We assume here that we have already carried out the one-time work in Section 22. We begin with the following query algorithm.

Algorithm 24.1. ("Compute F_ν^ℓ"): *Given* $\ell \in \mathbb{Z}$, $\nu \in \mathbb{Z}^n$ *and* $\underline{x} \in \mathbb{R}^n$, *we compute* $J_{\underline{x}}(F_\nu^\ell)$, *using work at most* $C \log(N/\epsilon)$.

Explanation: We first compute $Q_\nu^{\langle s \rangle}$ from (23.2). Recall that $Q_\nu^{\langle s \rangle}$ is a dyadic cube, and $Q_\nu^\ell = (Q_\nu^{\langle s \rangle})^*$. We then perform a binary search as in (22.13), to determine whether $Q_\nu^{\langle s \rangle}$ appears in the list $Q^{(1)}, \dots, Q^{(L)}$.
If $Q_\nu^{\langle s \rangle}$ does appear in the list $Q^{(1)}, \dots, Q^{(L)}$, then the binary search (22.13) returns a λ such that $Q_\nu^{\langle s \rangle} = Q^{(\lambda)}$. We find ourselves in Case 1 from Section 23, and therefore, we have $F_\nu^\ell = F^{(\lambda)}$. We compute $J_{\underline{x}}(F^{(\lambda)})$ by applying (22.18), and we return $J_{\underline{x}}(F_\nu^\ell) = J_{\underline{x}}(F^{(\lambda)})$.
If $Q_\nu^{\langle s \rangle}$ does not appear in the list $Q^{(1)}, \dots, Q^{(L)}$, then we find ourselves in Case 2 from Section 23. We defined S_ν^ℓ by applying Algorithm 11.1 to the inputs $\epsilon, E, Q_\nu^{\langle s \rangle}$ (as explained in Section 23, just after (23.4)). We are guaranteed that S_ν^ℓ is empty or a singleton. (See (23.9)). Consequently, we may compute the set S_ν^ℓ by applying Algorithm 11.4.

- If $S_\nu^\ell = \phi$, then we are in Subcase 2a from Section 23.
  In that subcase, we defined $F_\nu^\ell = 0$. We return $J_{\underline{x}}(F_\nu^\ell) = 0$.
- If $S_\nu^\ell = \{y_0\}$, then we are in Subcase 2b from Section 23.
  In that subcase, we defined $F_\nu^\ell = F^{(y_0)}$. We compute $J_{\underline{x}}(F^{(y_0)})$ by applying (22.25), and we return $J_{\underline{x}}(F) = J_{\underline{x}}(F^{(y_0)})$.

Thus, in all cases, we have succeeded in computing $J_{\underline{x}}(F_\nu^\ell)$.
Let us estimate the work of the above computation.
The computation of $Q_\nu^{\langle s \rangle}$ takes work at most C.
The binary search using (22.13) takes work at most $C \log(N/\epsilon)$.
The computation of $J_{\underline{x}}(F^{(\lambda)})$ using (22.18) takes work at most C.
The computation of S_ν^ℓ using Algorithm 11.4 (not Algorithm 11.1) takes work at most $C \log N$.
The computation of $J_{\underline{x}}(F^{(y_0)})$ using (22.25) requires work at most C.

Consequently, the total work to compute $J_{\underline{x}}(F_\nu^\ell)$ is at most $C \log(N/\epsilon)$. This completes our explanation of Algorithm 24.1.

We are almost ready to present the query algorithm within Algorithm 21.1. We prepare the way by making a few remarks.

We recall that F is given by (23.16), (23.17), and that the cutoff functions appearing there satisfy

$$\operatorname{supp} \chi_\ell \subset (\ell - 1, \ell + 1), \text{ and } \operatorname{supp} \theta_\nu^\ell \subset Q_\nu^\ell; \text{ see (10.2) and (10.7)}.$$

Consequently, for $\underline{x} \in \mathbb{R}^n \setminus E$ it is natural to define

$$(1) \qquad \Lambda(\underline{x}) = \{\ell \in \mathbb{Z} : \epsilon \log \delta(\underline{x}) \in (\ell - 1, \ell + 1)\}.$$

Also, for $\underline{x} \in \mathbb{R}^n \setminus E$ and for $\ell \in \Lambda(\underline{x})$, we define

$$(2) \qquad V(\underline{x}, \ell) = \{\nu \in \mathbb{Z}^n : Q_\nu^\ell \ni \underline{x}\}.$$

We then have

$$(3) \qquad J_{\underline{x}}(F) = \sum_{\ell \in \Lambda(\underline{x})} \sum_{\nu \in V(\underline{x}, \ell)} J_{\underline{x}}(\chi_\ell(\epsilon \log \delta(\cdot))) \odot_{\underline{x}} J_{\underline{x}}(\theta_\nu^\ell) \odot_{\underline{x}} J_{\underline{x}}(F_\nu^\ell), \text{ for } \underline{x} \in \mathbb{R}^n \setminus E.$$

On the other hand, for $\underline{x} \in E$, we have

$$(4) \qquad J_{\underline{x}}(F) = P^{\underline{x}},$$

since F agrees with $\vec{P}$ (as shown in Section 23).

Our query algorithm for $J_{\underline{x}}(F)$ makes straightforward use of formulas (3) and (4).

Algorithm 24.2 (Query Algorithm within Algorithm 21.1): *Given* $\underline{x} \in \mathbb{R}^n$, *we compute* $J_{\underline{x}}(F)$, *using work at most* $C \log(N/\epsilon)$.

Explanation: First, we perform a binary search to decide whether $\underline{x} \in E$.

If $\underline{x} \in E$, then we return the polynomial $P^{\underline{x}}$, which is the correct answer, thanks to (4).

If $\underline{x} \notin E$, then we proceed as follows.

We compute $J_{\underline{x}}(\delta(\cdot))$, using (22.9). In particular, this gives us the value $\delta(\underline{x})$, from which we can compute the set $\Lambda(\underline{x})$ in (1). Note that $\#(\Lambda(\underline{x})) = 1$ or 2. For each $\ell \in \Lambda(\underline{x})$, we compute $\left(\frac{d}{dt}\right)^k \chi_\ell(t)$ at $t = \epsilon \log \delta(\underline{x})$, for $k = 0, \ldots, m$, by Algorithm 10.1. From the data already computed, we can now compute $J_{\underline{x}}(\chi_\ell(\epsilon \log \delta(\cdot)))$ for each $\ell \in \Lambda(\underline{x})$.

Next, for each $\ell \in \Lambda(\underline{x})$, we compute the set $V(\underline{x}, \ell)$ in (2), using Algorithm 10.5 ("Find-Relevant-Cubes"). Note that $\#(V(\underline{x}, \ell)) \leq C$, as mentioned in our statement of Algorithm 10.5.

For each $\ell \in \Lambda(\underline{x})$, and for each $\nu \in V(\underline{x}, \ell)$, we now compute $J_{\underline{x}}(\theta_\nu^\ell)$ by Algorithm 10.4, and $J_{\underline{x}}(F_\nu^\ell)$ by Algorithm 24.1.

We then return the polynomial $J_{\underline{x}}(F)$, which we compute from (3).

Thus, we have succeeded in computing $J_{\underline{x}}(F)$.

Let us estimate the work required for the above computation.
The binary search requires work at most $C \log N$.
The application of (22.9) to compute $J_{\underline{x}}(\delta(\cdot))$ also takes work at most $C \log N$.
The computation of $\Lambda(\underline{x})$ from $\delta(\underline{x})$ takes work at most C.
The computation of $\left(\frac{d}{dt}\right)^k \chi_\ell(t)$ at $t = \epsilon \log \delta(\underline{x})$ for $k = 0, \dots, m$ and all $\ell \in \Lambda(\underline{x})$ takes work at most C.
The computation of $J_{\underline{x}}(\chi_\ell(\epsilon \log \delta(\cdot)))$ for all $\ell \in \Lambda(\underline{x})$ then takes work at most C.
The applications of Algorithm 10.5 take total work at most C.

For each $\ell \in \Lambda(\underline{x})$ and $\nu \in V(\underline{x}, \ell)$, the application of Algorithm 10.4 takes work at most C, and the application of Algorithm 24.1 takes work at most $C \log(N/\epsilon)$.
Since there are at most C such (ℓ, ν), the total work of all the applications of Algorithms 10.4 and 24.1 is at most $C \log(N/\epsilon)$.
The evaluation of the right-hand side of (3) then requires work at most C.
Altogether, the work of Algorithm 24.2 is at most $C \log(N/\epsilon)$.
This agrees with what we claimed for the work at query time in Section 21.

Our explanations of Algorithms 24.2 and 21.1 are complete.

25. Remarks on the One-Time Work

In this section, we return to the setting of Section 22. For future reference, we set down several remarks on the one-time work of that section, and we show that the one-time work can be simplified a bit. We then prove Theorem 7 (from the Introduction), one of our main results.

Proposition 25.1. *Let $\epsilon, \vec{P} = (P^x)_{x \in E}$, $N = \#(E)$ be as in Section 22. Then the algorithm given in Section 22 performs as follows.*

(a) *It computes the representatives x'_ν, x''_ν $(\nu = 1, \dots, \nu_{\max})$ from the WSPD for E, with κ a small enough controlled constant.*

The x'_ν, x''_ν depend only on E, and not on ϵ or on the polynomials $P^x (x \in E)$. We have $\nu_{\max} \leq CN$.

(b) *It computes the least $\hat{M} \geq 0$ such that*

$$|\partial^\alpha P^x(x)| \leq \hat{M} \quad \text{for } |\alpha| \leq m,\ x \in E \text{ and}$$
$$|\partial^\alpha (P^{x'_\nu} - P^{x''_\nu})(x'_\nu)| \leq |x'_\nu - x''_\nu|^{m-|\alpha|} \cdot \hat{M} \quad \text{for } |\alpha| \leq m-1,\ 1 \leq \nu \leq \nu_{\max}.$$

(c) *It computes a list of cubes $Q^{(\lambda)}$ and sets $S^{(\lambda)} \subseteq E$ $(\lambda = 1, \dots, L)$, with $L \leq \frac{C}{\epsilon} \cdot N$. This list is computed from ϵ and E, without using the polynomials $P^x (x \in E)$. For each λ, the set $S^{(\lambda)}$ arises by applying Algorithm 11.1 to the input data $\epsilon, Q^{(\lambda)}, E$.*

(d) *For each* $\lambda(1 \le \lambda \le L)$*, it applies Algorithm* 20.1 *to the input data* $\epsilon, Q^{(\lambda)}$, $\vec{P}|_{S^{(\lambda)}}$*, to compute a number* $\mathcal{N}_\epsilon^{(\lambda)}(\vec{P})$*. (In particular, these input data are as assumed in Algorithm* 20.1.*) We have*

$$\mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \le (1 + C\epsilon) \cdot \| (\vec{P}|_{S^{(\lambda)}}) \|_{C^m(\mathbb{R}^n)} \quad \textit{for each } \lambda.$$

(e) *For each* $y_0 \in E$*, it applies Algorithm* 15.1 *to the input data* $\epsilon, \vec{P}|_{\{y_0\}}$*, to compute a number* $\mathcal{N}_\epsilon(\vec{P}, y_0)$*. (In particular, these input data are as assumed in Algorithm* 15.1.*) We have*

$$\mathcal{N}_\epsilon(\vec{P}, y_0) \le (1 + C\epsilon) \cdot \| (\vec{P}|_{\{y_0\}}) \|_{C^m(\mathbb{R}^n)}, \quad \textit{for each } y_0 \in E.$$

(f) *It returns the number*

$$\mathcal{N}_\epsilon(\vec{P}) = \max \left\{ \frac{\hat{M}}{2C^*}, \mathcal{N}_\epsilon^{(\lambda)}(\vec{P})\,(1 \le \lambda \le L)\,, \mathcal{N}_\epsilon(\vec{P}, y_0)\,(y_0 \in E) \right\},$$

where C^* *is the controlled constant fixed in Section* 22.

(g) *The work and storage to compute all the* $Q^{(\lambda)}, S^{(\lambda)}$ *for* $\lambda = 1, \dots, L$*, are at most* $\exp(C/\epsilon)N \log N$*, and* $\exp(C/\epsilon)N$*, respectively.*

Proof. Everything asserted in the above Proposition was shown in Section 22. ∎

Let $\mathcal{N}_\epsilon(\vec{P})$ be as above. In Section 23, we exhibited a function $F \in C^m(\mathbb{R}^n)$, and proved the following properties of $\mathcal{N}_\epsilon(\vec{P})$ and F.

(A) F agrees with $\vec{P}$.

(B) $\| F \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon)\mathcal{N}_\epsilon(\vec{P})$.

(C) $\mathcal{N}_\epsilon(\vec{P}) \le (1 + C\epsilon) \cdot \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

From (A) and (B), we conclude that

(1) $\| \vec{P} \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon)\mathcal{N}_\epsilon(\vec{P})$,

by definition of the C^m-norm of a Whitney field. On the other hand, from (22.4), we recall that

(2) $\frac{\hat{M}}{2C^*} \le \frac{1}{2} \| \vec{P} \|_{C^m(\mathbb{R}^n)}$.

Combining (1) and (2), we see that

(3) $\frac{\hat{M}}{2C^*} \le \frac{1+C\epsilon}{2} \cdot \mathcal{N}_\epsilon(\vec{P})$.

Since ϵ is less than a small enough controlled constant, we have $\frac{1+C\epsilon}{2} < 1$.

Hence, comparing (3) with conclusion (f) of Proposition 25.1, we find that

(f′) $\quad \mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_\epsilon^{(\lambda)}(\vec{P})\,(1 \le \lambda \le L),\ \mathcal{N}_\epsilon(\vec{P}, y_0)\ (y_0 \in E)\}.$

In other words, we can delete the term $\frac{\hat{M}}{2C^*}$ from the maximum in (f). Consequently, we needn't bother to compute $\hat{M}$ at all; we may simply omit Step 2 from the algorithm described in Section 22.

From Proposition 25.1 (d) and (e), we recall that

$$\mathcal{N}_\epsilon^{(\lambda)}(\vec{P}) \le (1 + C\epsilon)\ \|\ (\vec{P}|_{S^{(\lambda)}})\ \|_{C^m(\mathbb{R}^n)} \quad \text{for each } \lambda, \text{ and}$$
$$\mathcal{N}_\epsilon(\vec{P}, y_0) \le (1 + C\epsilon) \cdot \|\ (\vec{P}|_{\{y_0\}})\ \|_{C^m(\mathbb{R}^n)}, \quad \text{for each } y_0 \in E.$$

Consequently, (f′) implies the estimate

$$\mathcal{N}_\epsilon(\vec{P}) \le$$
$$\le (1 + C\epsilon) \cdot \max\{\|\ (\vec{P}|_{S^{(\lambda)}})\ \|_{C^m(\mathbb{R}^n)}\ (1 \le \lambda \le L), \|\ (\vec{P}|_{\{y_0\}})\ \|_{C^m(\mathbb{R}^n)}\ (y_0 \in E)\}.$$

Together with (1), this yields

(4) $$\|\ \vec{P}\ \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon)\ \cdot \max\Big\{\ \|\ (\vec{P}|_{S^{(\lambda)}})\ \|_{C^m(\mathbb{R}^n)}\ (1 \le \lambda \le L),\ \|\ (\vec{P}|_{\{y_0\}})\ \|_{C^m(\mathbb{R}^n)}\ (y_0 \in E)\Big\}.$$

We recall that each $S^{(\lambda)}$ arises from the input $\epsilon, E, Q^{(\lambda)}$ by Algorithm 11.1. By the defining properties of that algorithm, we have, for $1 \le \lambda \le L$, that

(5) $\quad S^{(\lambda)} \subset (Q^{(\lambda)})^{**} \cap E$, and

(6) $\quad |y - y'| > c\epsilon e^{-2/\epsilon}\,\delta_{Q^{(\lambda)}}$ for any two distinct points $y, y' \in S^{(\lambda)}$.

Recall that the $Q^{(\lambda)}$ and $S^{(\lambda)}$ are defined for $1 \le \lambda \le L$, with

(7) $\quad L \le \frac{CN}{\epsilon}$, $N = \#(E)$.

It is convenient to define $Q^{(\lambda)}$ and $S^{(\lambda)}$ for $L + 1 \le \lambda \le L + N$, as follows. Let $\{y_1, \dots, y_N\}$ be an enumeration of E. For $i = 1, 2, \dots, N$, we define $S^{(L+i)} = \{y_i\}$, and we take $Q^{(L+i)}$ to be a dyadic cube of side 1 (say), containing y_i.
Then (5), (6) hold also for $L + 1 \le \lambda \le L + N$. Our estimate (4) may be restated as follows.

(8) $$\|\ \vec{P}\ \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon)\ \cdot \max\{\|\ (\vec{P}|_{S^{(\lambda)}})\ \|_{C^m(\mathbb{R}^n)} : 1 \le \lambda \le L + N\}.$$

Let us estimate the work and storage needed to compute (and store) all the $Q^{(\lambda)}$ and $S^{(\lambda)}$, $\lambda = 1, \dots, L + N$.

Trivially, we can compute and store all the $Q^{(\lambda)}$ and $S^{(\lambda)}$ for $L+1 \leq \lambda \leq L+N$ with work and storage at most CN. Hence, Proposition 25.1(g) implies the following.

(9) The cubes $Q^{(\lambda)}$ and the sets $S^{(\lambda)}$, for all $\lambda = 1, \ldots, L+N$, can be computed and stored, with at most $\exp(C/\epsilon)N \log N$ operations, in space at most $\exp(C/\epsilon)N$.

Theorem 7 from the introduction now follows at once from (5)...(9).

26. Minimax Functions

Let V, W be (real) finite-dimensional vector spaces, and let $\lambda_i : V \oplus W \to \mathbb{R}$ $(i = 1, \ldots, I)$ be linear functionals.

Then we can define a function $\Lambda : V \to \mathbb{R}$, by setting

$$\Lambda(v) = \min_{w \in W} \max_{i=1,\ldots,I} |\lambda_i(v, w)| \text{ for } v \in V. \tag{1}$$

(It is an elementary exercise to check that the minimum over all $w \in W$ is attained.)

We call

$$\eta = (V, W, \lambda_1, \ldots, \lambda_I) \tag{2}$$

a "V-descriptor", with V as in (1), (2). Also, we call $\Lambda(v)$ the "minimax function arising from η", and we write this function as $\Lambda(v, \eta)$. For brevity, we may say that η is a "descriptor" (omitting the V), or that Λ is a "minimax function" (omitting the η).

Here, we allow degenerate cases. If $I = 0$, then (1) simply means that $\Lambda(v) = 0$ for all $v \in V$. If W is a single point $\{0\}$, then (1) reduces to

$$\Lambda(v) = \max_{i=1,\ldots,I} |\lambda_i(v)|, \quad \text{for linear functionals } \lambda_1, \ldots, \lambda_I : V \to \mathbb{R}.$$

We define the "dimension" of the descriptor η in (2) to be the dimension of the vector space W. Also, we define the "length" of the descriptor (2) to be the number of linear functionals appearing in (2), i.e., the integer I.

Given a vector $v \in V$, we would like to compute a vector $w \in W$ that nearly achieves the minimum in (1). Therefore, we make the following definition.

Let $v \in V$, $w^0 \in W$, let $\Gamma \geq 1$ be a real number, and let Λ, η be given by (1) and (2). We say that w^0 is a "Γ-optimal" vector for v, η, provided

$$\max_{i=1,\ldots,I} |\lambda_i(v, w^0)| \leq \Gamma \cdot \min_{w \in W} \max_{i=1,\ldots,I} |\lambda_i(v, w)|.$$

(Compare with Algorithm 26.3 below.)

Suppose that V and W in (1), (2) are identified with Euclidean spaces $\mathbb{R}^D$ and $\mathbb{R}^d$, respectively. Then, for $i = 1, \ldots, I$, the functional λ_i is specified by $(D + d)$ coordinates $(\lambda_i^1, \ldots, \lambda_i^{D+d})$.

Thus, the descriptor η can be stored in memory, and it occupies storage at most

$$C \cdot (\dim V + \dim \eta) \cdot \mathsf{length}(\eta) + C. \tag{3}$$

In this section, we perform a few elementary operations involving descriptors and minimax functions. For our algorithms involving descriptors, we suppose that V and W are identified with $\mathbb{R}^D$ and $\mathbb{R}^d$ (respectively), and that any given descriptor η is specified by its coordinates as above.

Proposition 26.1. *Let* $T : V_1 \to V_2$ *be a linear map, and let*

$$\eta = (V_2, W, \lambda_1, \ldots, \lambda_I) \text{ be a } V_2\text{-descriptor.} \tag{4}$$

Define the V_1*-descriptor*

$$\eta \circ T = (V_1, W, \lambda_1 \circ (T \oplus \mathrm{Id}), \ldots, \lambda_I \circ (T \oplus \mathrm{Id}))\,,$$

where Id *denotes the identity map on* W.

Then we have

$$\Lambda(v_1, \eta \circ T) = \Lambda(Tv_1, \eta) \text{ for all } v_1 \in V_1. \tag{5}$$

Moreover,

$$\dim(\eta \circ T) = \dim(\eta) \text{ and } \mathsf{length}(\eta \circ T) = \mathsf{length}\,(\eta). \tag{6}$$

Finally, if w *is a* Γ*-optimal vector for* Tv, η*, then* w *is also a* Γ*-optimal vector for* $v, \eta \circ T$.

Proof. Trivial. ∎

Thus, if $T : V_1 \to V_2$ is linear, and if Λ is a minimax function on V_2, then $\Lambda \circ T$ is a minimax function on V_1.

Proposition 26.2. *For* $a = 1, \ldots, A$*, let*

$$\eta_a = (V, W_a, \lambda_1^a, \ldots, \lambda_{I(a)}^a) \text{ be a } V\text{-descriptor.} \tag{7}$$

Let

$$W = W_1 \oplus \cdots \oplus W_A, \tag{8}$$

and, for $a = 1, \ldots, A$*, let*

$$\pi_a : V \oplus W \to V \oplus W_a \tag{9}$$

be the natural projection.

Let $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_I$ *be an enumeration of the family of linear functionals*

$$\lambda_i^a \circ \pi_a : V \oplus W \to \mathbb{R} \ (1 \le a \le A,\ 1 \le i \le I(a)). \tag{10}$$

Define a V-descriptor $\eta_1 \vee \cdots \vee \eta_A$, by setting

$$(11)\quad \eta_1 \vee \cdots \vee \eta_A = (V, W, \tilde{\lambda}_1, \ldots, \tilde{\lambda}_I),$$

with W and $\tilde{\lambda}_i$ as in (8) *and* (10).

Then, for all $v \in V$, we have

$$(12)\quad \Lambda(v, \eta_1 \vee \cdots \vee \eta_A) = \max_{a=1,\ldots,A} \Lambda(v, \eta_a).$$

Moreover,

$$(13)\quad \mathsf{length}\,(\eta_1 \vee \cdots \vee \eta_A) = \mathsf{length}\,(\eta_1) + \cdots + \mathsf{length}\,(\eta_A),$$

and

$$(14)\quad \dim(\eta_1 \vee \cdots \vee \eta_A) = \dim(\eta_1) + \cdots + \dim(\eta_A).$$

Finally, let $v \in V$, and suppose that $w_a \in W$ is Γ-optimal for v, η_a $(a = 1, \ldots, A)$.

Then $(w_1, \ldots, w_A) \in W_1 \oplus \cdots \oplus W_A$ is Γ-optimal for $v, \eta_1 \vee \cdots \vee \eta_A$.

Thus, if $\Lambda_1, \ldots, \Lambda_A$ are minimax functions on V, then the function

$$\Lambda(v) = \max\{\Lambda_1(v), \ldots, \Lambda_A(v)\}\,(v \in V)$$

is again a minimax function.

If $\{\eta_\alpha, \alpha \in \mathcal{A}\}$ is a finite collection of V-descriptors indexed by $\alpha \in \mathcal{A}$, then we sometimes write $\bigvee_{\alpha \in \mathcal{A}} \eta_\alpha$ to denote $\eta_{\alpha_1} \vee \cdots \vee \eta_{\alpha_A}$, where $(\alpha_1, \ldots, \alpha_A)$ is an enumeration of $\mathcal{A}$.

Proof of Proposition 26.2. Fix $v \in V$. For each $a = 1, \ldots, A$, let $\tilde{w}_a$ be a minimizer for the function

$$w \mapsto \max_{i=1,\ldots,I(a)} |\lambda_i^a(v, w)|\,(w \in W_a)\,.$$

Then, for any $w = (w_1, \ldots, w_A) \in W = W_1 \oplus \cdots \oplus W_A$, and for any a $(1 \le a \le A)$, we have

$$\max_{1 \le i \le I(a)} |\lambda_i^a \circ \pi_a(v, w)| = \max_{1 \le i \le I(a)} |\lambda_i^a(v, w_a)| \ge \Lambda(v, \eta_a)\,,$$

with equality in case $w_a = \tilde{w}_a$.
Consequently, for any $w = (w_1, \ldots, w_A) \in W$, we have

$$\max_{1 \le a \le A} \max_{1 \le i \le I(a)} |\lambda_i^a \circ \pi_a(v, w)| \ge \max_{1 \le a \le A} \Lambda(v, \eta_a)$$

with equality in case $w = (\tilde{w}_1, \ldots, \tilde{w}_A)$. Thus,

$$(15)\quad \min_{w\in W} \max\{|\lambda_i^a \circ \pi_a(v,w)| : 1 \le a \le A\,,\ 1 \le i \le I(a)\} \\ = \max\{\Lambda(v,\eta_a) : 1 \le a \le A\}\,.$$

In view of the definition (11) of $\eta_1 \vee \cdots \vee \eta_A$, the equality (15) tells us that

$$\Lambda(v, \eta_1 \vee \cdots \vee \eta_A) = \max\{\Lambda(v,\eta_1), \ldots, \Lambda(v,\eta_A)\}\,,$$

completing the proof of (12). Conclusions (13) and (14) are obvious.

Finally, suppose w_a is Γ-optimal for v, η_a for each $a = 1, \ldots, A$. Then

$$\max_{i,a} |\lambda_i^a \circ \pi_a(v, (w_1, \ldots, w_A))| = \max_a \max_i |\lambda_i^a(v, w_a)| \\ \le \max_a \Gamma \cdot \Lambda(v, \eta_a) = \Gamma \cdot \Lambda(v, \eta_1 \vee \cdots \vee \eta_A) \text{ by } (12).$$

Thus, $(w_1, \ldots, w_A)$ is Γ-optimal for $v, \eta_1 \vee \cdots \vee \eta_A$. ∎

To implement the above propositions, we set down the following algorithms.

Algorithm 26.1. *Given* $T : \mathbb{R}^{D_1} \to \mathbb{R}^{D_2}$ *(specified as a matrix), and given an* $\mathbb{R}^{D_2}$*-descriptor* η*, we compute the* $\mathbb{R}^{D_1}$*-descriptor* $\eta \circ T$*.*

We recall that $\dim(\eta \circ T) = \dim(\eta)$ *and* $\text{length}\,(\eta \circ T) = \text{length}\,(\eta)$*.*

The work required for the computation is at most

$$C + C \cdot \text{length}\,(\eta) \cdot [\dim V_1 \cdot \dim V_2 + \dim(\eta)].$$

The storage required for the computation (aside from that required to hold the inputs η, T*) is at most*

$$C + C \cdot (\dim V_1 + \dim(\eta)) \cdot \text{length}(\eta)\,.$$

Algorithm 26.2. *Given a list* $\eta_1, \ldots, \eta_A$ *of* $\mathbb{R}^D$*-descriptors, we compute the* $\mathbb{R}^D$*-descriptor* $\eta_1 \vee \cdots \vee \eta_A$*. Recall that*

$$\dim(\eta_1 \vee \cdots \vee \eta_A) = \dim(\eta_1) + \cdots + \dim(\eta_A), \text{ and}$$
$$\text{length}\,(\eta_1 \vee \cdots \vee \eta_A) = \text{length}\,(\eta_1) + \cdots + \text{length}\,(\eta_A)\,.$$

The work and storage required for the computation are at most

$$C + C \cdot [D + \sum_{a=1}^{A} \dim(\eta_a)] \cdot [\sum_{a=1}^{A} \text{length}\,(\eta_a)]\,.$$

We omit the straightforward explanation of Algorithms 26.1 and 26.2.

Given a minimax function $\Lambda(\cdot)$ in the form (1), and given a vector $v \in V$, we would like to find a vector w that nearly achieves the minimum in (1). Under favorable circumstances, we can find such a w, thanks to the following application of the Ellipsoid Algorithm.

Algorithm 26.3. *Given a list* $\lambda_i(\cdot) - b_i$ $(i = 1, \dots, I)$ *of affine functions on* $\mathbb{R}^D$ *(with* $\lambda_i \in (\mathbb{R}^D)^*$ *and* $b_i \in \mathbb{R}$*); given a real number* $\Gamma \geq 1$ *and a vector* $w_0 \in \mathbb{R}^D$ *such that*

$$(\dagger) \quad \max_{i=1,\dots,I} |\lambda_i(w_0) - b_i| \leq \Gamma \cdot \min_{w \in \mathbb{R}^D} \max_{i=1,\dots,I} |\lambda_i(w) - b_i|;$$

and given $\epsilon > 0$*; we compute a vector* $w_1 \in \mathbb{R}^D$*, such that*

$$(\dagger\dagger) \quad \max_{i=1,\dots,I} |\lambda_i(w_1) - b_i| \leq (1 + \epsilon) \cdot \min_{w \in \mathbb{R}^D} \max_{i=1,\dots,I} |\lambda_i(w) - b_i|.$$

The work of the computation is at most $C(D + I)^5 (\log I) (\log \frac{ID\Gamma}{\epsilon})$ *and the storage is at most* $C \cdot (D + I)^2$.

Explanation: Let W be a vector subspace of $\mathbb{R}^D$, complementary to $W_0 := \mathrm{nullspace}(\lambda_1) \cap \cdots \cap \mathrm{nullspace}(\lambda_I)$, and let $\pi : \mathbb{R}^D \to W$ be the natural projection arising from the direct sum decomposition $\mathbb{R}^D = W \oplus W_0$.

Note that $\lambda_i(\pi w) = \lambda_i(w)$ for any $w \in \mathbb{R}^D$ and $i = 1, \dots, I$. Hence, in (†), we may replace w_0 by πw_0; and then, in the statement of our problem, we may pass from $\mathbb{R}^D$ to the subspace W. Consequently, we may assume without loss of generality that $\mathrm{nullspace}(\lambda_1) \cap \cdots \cap \mathrm{nullspace}(\lambda_I) = \{0\}$.

Thus, the quadratic form $\sum_{i=1}^{I} (\lambda_i(w))^2$ on $\mathbb{R}^D$ may be assumed to be strictly positive-definite.

Now let

$$(16) \quad \hat{M} = \max_{i=1,\dots,I} |\lambda_i(w_0) - b_i|,$$

and let

$$(17) \quad K = \{(M, w) \in \mathbb{R} \oplus \mathbb{R}^D : |\lambda_i(w) - b_i| \leq M \text{ for } i = 1, \dots, I,\ \Gamma^{-1}\hat{M} \leq M \leq 10\hat{M}\}.$$

Note that

$$\{(M, w) \in \mathbb{R} \oplus \mathbb{R}^D : |\lambda_i(w) - b_i| \leq M \text{ for } i = 1, \dots, I,\ M < \Gamma^{-1}\hat{M}\}$$

is empty, by hypothesis (†). Note also that on

$$\{(M, w) \in \mathbb{R} \oplus \mathbb{R}^D : |\lambda_i(w) - b_i| \leq M \text{ for } i = 1, \dots, I,\ M > 10\hat{M}\}$$

we evidently have $M > 10\hat{M}$, whereas the point $(\hat{M}, w_0)$ belongs to K. The above remarks show that

$$(18) \quad \min_{w \in \mathbb{R}^D} \max_{i=1,\dots,I} |\lambda_i(w) - b_i| = \min\{M : (M, w) \in K\}.$$

For any $(M, w) \in K$, we have $0 \leq M \leq 10\hat{M}$, and

$$|\lambda_i(w - w_0)| \leq |\lambda_i(w) - b_i| + |\lambda_i(w_0) - b_i| \leq M + \hat{M} \leq 11\hat{M}$$

for $i = 1, \ldots, I$; and therefore

$$(19) \quad K \subset \{(M, w) \in \mathbb{R} \oplus \mathbb{R}^D : M^2 + \sum_{i=1}^{I} (\lambda_i(w - w_0))^2 \leq (100 + 121\, I)\, \hat{M}^2\} \equiv E.$$

On the other hand, suppose $(M, w) \in \mathbb{R} \oplus \mathbb{R}^D$, with

$$(M - 3\hat{M})^2 + \sum_{i=1}^{I} (\lambda_i(w - w_0))^2 \leq \hat{M}^2.$$

Then $|M - 3\hat{M}| \leq \hat{M}$; and for $i = 1, \ldots, I$, we have $|\lambda_i(w - w_0)| \leq \hat{M}$, hence

$$|\lambda_i(w) - b_i| \leq |\lambda_i(w_0) - b_i| + \hat{M} \leq 2\hat{M} \leq M \text{ (since } |M - 3\hat{M}| \leq \hat{M}).$$

Consequently,

$$(20) \quad \{(M, w) \in \mathbb{R} \oplus \mathbb{R}^D : (M - 3\hat{M})^2 + \sum_{i=1}^{I} (\lambda_i(w - w_0))^2 \leq \hat{M}^2\} \subset K.$$

Trivially, we have

$$(21) \quad \Gamma^{-1}\hat{M} \leq M \leq 10\,\hat{M} \text{ for } (M, w) \in K, \text{ and}$$

$$(22) \quad |M| \leq (100 + 121I)^{1/2}\,\hat{M} \text{ for } (M, w) \in E \text{ (see (19))}.$$

Thanks to (19), (20), (22), the Ellipsoid Algorithm produces a point

$$(23) \quad (M_1, w_1) \in K,$$

such that

$$(24) \quad M_1 \leq \min_{(M,w)\in K} M + \epsilon\Gamma^{-1}\,\hat{M}.$$

From (17) we obtain trivially

$$(25) \quad \hat{M} \leq \Gamma \min_{(M,w)\in K} M.$$

Combining (24) and (25), we find that

$$M_1 \le (1+\epsilon)\cdot \min\{M : (M,w)\in K\}.$$

We have also $|\lambda_i(w_1)-b_i| \le M_1$ for $i=1,\dots,I$, thanks to (23). Therefore,

$$\begin{aligned}\max_{i=1,\dots,I} |\lambda_i(w_1)-b_i| &\le (1+\epsilon)\min\{M : (M,w)\in K\}\\ &= (1+\epsilon)\min_{w\in\mathbb{R}^D}\max_{i=1,\dots,I}|\lambda_i(w)-b_i|,\end{aligned}$$

thanks to (18). Thus, w_1 satisfies (††).

Let us review how we computed w_1.

- By linear algebra, we reduced matters to the case $\bigcap_{i=1}^{I} \mathsf{nullspace}\,(\lambda_i) = \{0\}$.
- We computed $\hat{M}$ from formula (16).
- We defined the convex set K in (17).
- Using the ellipsoid method, starting with (17), (19), (20), (22), we computed the point $(M_1, w_1)\in K$.

This gives w_1.

We can easily estimate the work and storage used in the above computation. We may assume that $D, I \ge 1$, since our problem is trivial otherwise.

- The linear algebra in our computation takes space at most CDI and work at most $CDI\cdot(D+I)$.
- The computation of $\hat{M}$ from (16) requires work at most CDI and space at most C (aside from the space used to hold the input data).
- Exhibiting the constraints defining K in (17) requires work and space at most CDI.
- Exhibiting the ellipsoids in (19), (20) requires work at most CD^2I in space at most CD^2.
- The quantities playing the rôles of ϵ, λ, D, L in the Ellipsoid Algorithm here are $\epsilon' = c\epsilon\Gamma^{-1}(100+121\,I)^{-1/2}$, $\lambda' = (100+121\,I)^{-1/2}$, $D+1$, and $2I+1$, respectively. Consequently, our application of the ellipsoid method uses work at most $CD^4I\log(100+121\,I)\log(\frac{\Gamma}{\epsilon}\cdot D\cdot(100+121\,I))$ in space at most CDI.

Altogether, the work and storage used by Algorithm 26.3 are at most

$$C\cdot(D+I)^5(\log I)\left(\log\frac{\Gamma D I}{\epsilon}\right) \quad\text{and}\quad C\cdot(D+I)^2, \quad\text{respectively.}$$

The explanation of Algorithm 26.3 is complete.

27. Minimax Functions of Whitney Fields

In this section, we study the quantities $\mathcal{N}_\epsilon(\vec{P})$, $\mathcal{N}_\epsilon(\vec{P}, Q)$, computed by Algorithms 14.1...20.1, as functions of the Whitney field $\vec{P}$. We shall see that these functions are all well-approximated by minimax functions, and we will provide algorithms to compute the descriptors η of those minimax functions. Moreover, given a Whitney field $\vec{P}$, we compute a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta$, with η as above.

We begin with some notation, definitions and remarks.

Recall that, for $E \subset \mathbb{R}^n$, $\mathcal{W}h(E)$ denotes the vector space of Whitney fields on E. If E is the disjoint union of two sets E_1 and E_2, then we identify $\mathcal{W}h(E)$ with $\mathcal{W}h(E_1) \oplus \mathcal{W}h(E_2)$ in the obvious way, and we write

$$\pi^E_{E_1} : \mathcal{W}h(E) \to \mathcal{W}h(E_1) \tag{1}$$

to denote the obvious projection.

We identify $\mathcal{W}h(E)$ with $\mathbb{R}^D$, $D = \dim[\mathcal{W}h(E)]$, by identifying $\vec{P} = (P^x)_{x\in E}$ with the coordinates $\partial^\alpha P^x(x)$ ($|\alpha| \leq m, x \in E$). In all our algorithms involving $\mathcal{W}h(E)$-descriptors, we assume that this identification has been made.

We recall from the preceding section that $\vec{P} \mapsto \Lambda(\vec{P}, \eta)$ ($\vec{P} \in \mathcal{W}h(S)$) is the minimax function arising from a given $\mathcal{W}h(S)$-descriptor η.

As usual, given two real numbers $X, Y \geq 0$ and a real number $\Gamma \geq 1$, we say that X and Y "differ by at most a factor of Γ" provided $\Gamma^{-1}X \leq Y \leq \Gamma X$.

The following algorithms accomplish the goals of this section.

Algorithm 27.1. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube Q, assumed to satisfy*

$$\delta_Q \leq \tilde{c}\epsilon^{-1}, \text{ with } \tilde{c} \text{ as in Algorithm } 14.1; \tag{2}$$

and given a set

$$S \subset Q^{**}, \tag{3}$$

assumed to satisfy

$$|y - y'| > e^{-3/\epsilon}\,\delta_Q \text{ for any two distinct points } y, y' \in S; \tag{4}$$

we compute a $\mathcal{W}h(S)$-descriptor η, with the following properties.

(5) *For any $\vec{P} \in \mathcal{W}h(S)$, the number $\mathcal{N}_\epsilon(\vec{P}, Q)$ computed from $\epsilon, Q, \vec{P}$ by Algorithm 14.1 differs by at most a factor $(1 + C\epsilon)$ from $\Lambda(\vec{P}, \eta)$.*

(6) *The length and dimension of η are at most $\exp(C/\epsilon)$.*

The work and storage used to compute η are at most $\exp(C/\epsilon)$.

Explanation: We recall Proposition 14.1. Let $B(x_0, r)$, S^+, $O(\epsilon, y)$ ($y \in S^+$) be as in that proposition. We can compute these objects with work and storage at most $\exp(C/\epsilon)$, by Proposition 14.1 (f). We set

(7) $\eta = (\mathcal{W}h(S), \mathcal{W}h(S^+ \setminus S), \lambda_1, \ldots, \lambda_I)$,

where $(\lambda_1, \ldots, \lambda_I)$ is an enumeration of the following family of linear functionals on $\mathcal{W}h(S) \oplus \mathcal{W}h(S^+ \setminus S) = \mathcal{W}h(S^+)$.

(8) The functionals $\vec{P} = (P^x)_{x \in S^+} \mapsto (1 + \hat{A}\epsilon)^{-1} \cdot \lambda(P^y)$ for $y \in S^+$, $\lambda \in O(\epsilon, y)$;

together with

(9) The functionals $\vec{P} = (P^x)_{x \in S^+} \mapsto [\hat{A}\, e^{4m/\epsilon} r^{-1} |y - y'|^{m+1-|\alpha|}]^{-1} \cdot \partial^\alpha (P^y - P^{y'})(y)$ for $|\alpha| \leq m$, $y, y' \in S^+$, $y \neq y'$.

Here, $\hat{A}$ is as in Proposition 14.1; recall that $\hat{A}$ is a controlled constant.

We will check that the descriptor η satisfies (5) and (6), and that it is computed from ϵ, Q, S using work and storage at most $\exp(C/\epsilon)$. In fact, both (6) and the desired estimate for work and storage are obvious by inspection of (7), (8), (9), since $\#(S^+) \leq \exp(C/\epsilon)$ and $\#(O(\epsilon, y)) \leq \exp(C/\epsilon)$ for each $y \in S^+$. (See Proposition 14.1 (b), (c).)

It remains only to check (5). Let $\vec{P} = (P^y)_{y \in S} \in \mathcal{W}h(S)$ be given. We recall that each $O(\epsilon, y)$ ($y \in S^+$) is symmetric about the origin. (See Proposition 14.1 (c).) Let $\xi^0 = [M^0, (P^0_y)_{y \in S^+}]$ be in Proposition 14.1 (d), (e); and let $\vec{P}' = (P^0_y)_{y \in S^+ \setminus S} \in \mathcal{W}h(S^+ \setminus S)$.

Comparing (7), (8), (9) with Proposition 14.1 (d), we learn that

(10) $\max_{i=1,\ldots,I} |\lambda_i(\vec{P}, \vec{P}')| \leq M^0$,

and that

(11) $M^0 \leq (1 + C\epsilon) \cdot \max_{i=1,\ldots,I} |\lambda_i(\vec{P}, \vec{P}'')|$ for any $\vec{P}'' \in \mathcal{W}h(S^+ \setminus S)$.

Comparing (10), (11) with the definition of $\Lambda(\vec{P}, \eta)$, we see that

(12) M^0 differs from $\Lambda(\vec{P}, \eta)$ by at most a factor of $(1 + C\epsilon)$,

and also

(13) $\vec{P}'$ is a $(1 + C\epsilon)$-optimal vector for $\vec{P}$, η.

Conclusion (5) is now immediate from (12), together with Proposition 14.1(e). This completes our explanation of Algorithm 27.1.

Algorithm 27.2. *Given* ϵ, Q, S *as in Algorithm* 27.1; *and given* $\vec{P} \in \mathcal{Wh}(S)$*; we compute a* $(1 + C\epsilon)$*-optimal vector* w^0 *for* $\vec{P}, \eta$*, where* η *is the* $\mathcal{Wh}(S)$*-descriptor computed from* ϵ, Q, S *by Algorithm* 27.1.

The work and storage used to compute w^0 *are at most* $\exp(C/\epsilon)$.

Explanation: We compute $\xi^0 = [M^0, (P^0_y)_{y \in S^+}]$ as in Proposition 14.1 (d), and we set

$$w^0 = \vec{P}'' = (P^0_y)_{y \in S^+ \smallsetminus S} \in \mathcal{Wh}(S^+ \smallsetminus S)\,.$$

According to (13), the vector w^0 is $(1 + C\epsilon)$-optimal for $\vec{P}, \eta$.

Thanks to Proposition 14.1 (f), the work and storage used to compute w^0 are at most $\exp(C/\epsilon)$.

This completes our explanation of Algorithm 27.2.

Algorithm 27.3. *Given* $\epsilon > 0$*, assumed to be less than a small enough controlled constant; and given a point* $y_0 \in \mathbb{R}^n$*; we compute a* $\mathcal{Wh}(\{y_0\})$*-descriptor* η*, such that:*

(a) *For any* $\vec{P} \in \mathcal{Wh}(\{y_0\})$*, the number* $\mathcal{N}_\epsilon(\vec{P})$ *computed from* $\epsilon, \vec{P}$ *by Algorithm* 15.1 *differs by at most a factor of* $(1 + C\epsilon)$ *from* $\Lambda(\vec{P}, \eta)$*;*

and

(b) *The length and dimension of* η *are at most* $\exp(C/\epsilon)$.

The work and storage used to compute η *are at most* $\exp(C/\epsilon)$.

Explanation Let Q be the cube computed as in Proposition 15.1, and let η be the descriptor computed from $\epsilon, Q, \{y_0\}$ by Algorithm 27.1. For any $\vec{P} \in \mathcal{Wh}(\{y_0\})$, Proposition 15.1 shows that the number $\mathcal{N}_\epsilon(\vec{P})$ computed by Algorithm 15.1 is equal to the number $\mathcal{N}_\epsilon(\vec{P}, Q)$ computed from $\epsilon, Q, \vec{P}$ by Algorithm 14.1. Hence (a) follows from (5). Evidently, (b) follows from (6).

The work and storage used to compute Q are at most C, by Proposition 15.1; and the work and storage used to compute η from $\epsilon, Q, \{y_0\}$ are at most $\exp(C/\epsilon)$, as we see from Algorithm 27.1.

This completes our explanation of Algorithm 27.3.

Algorithm 27.4. *Given* ϵ, y_0 *as in Algorithm* 27.3*, and given* $\vec{P} \in \mathcal{Wh}(\{y_0\})$*, we compute a* $(1 + C\epsilon)$*-optimal vector* w^0 *for* $\vec{P}, \eta$*, where* η *is the descriptor computed from* ϵ, y_0 *by Algorithm* 27.3.

The work and storage used to compute w^0 *are at most* $\exp(C/\epsilon)$.

Explanation: Let Q be the cube computed from ϵ, y_0 as in Proposition 15.1, let η be the $Wh(\{y_0\})$-descriptor computed from $\epsilon, Q, \{y_0\}$ by Algorithm 27.1, and let w^0 be the vector computed from $\epsilon, Q, \{y_0\}$, $\vec{P}$ by Algorithm 27.2. Thus, w^0 is a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta$, by the defining property of w^0 in Algorithm 27.2. Recall from our explanation of Algorithm 27.3 that our present η is precisely the descriptor computed from ϵ, y_0 by Algorithm 27.3. Thus, w^0 has the required property. Moreover, the estimates for work and storage given in Proposition 15.1 and Algorithm 27.2 show that our present computation of w^0 from $\epsilon, y_0, \vec{P}$ uses work and storage at most $\exp(C/\epsilon)$.

This completes our explanation of Algorithm 27.4.

Algorithm 27.5. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube, assumed to satisfy*

(14) $\delta_Q \leq e^{-1/(4\epsilon)}$;

and given a set

(15) $S \subset Q^{**}$,

assumed to satisfy

(16) $|y - y'| > \epsilon^2 e^{-2/\epsilon} \delta_Q$ *for any two distinct points* $y, y' \in S$;

we compute a $Wh(S)$-descriptor η, with the following properties.

(17) *For any $\vec{P} \in Wh(S)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, Q, \vec{P}$ by Algorithm* 16.1 *differs by at most a factor of $(1 + C\epsilon)$ from $\Lambda(\vec{P}, \eta)$.*

(18) *The length and dimension of η are at most $\exp(C/\epsilon)$.*

The work and storage used to compute η are at most $\exp(C/\epsilon)$.

Explanation: We recall Proposition 16.1. Let y_0, Q_{00} be as in that proposition. In particular, we can compute y_0, Q_{00} with work and storage at most C. For any $\vec{P} \in Wh(S)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from ϵ, Q, $\vec{P}$ by Algorithm 16.1 is given by

(19) $\mathcal{N}_\epsilon(\vec{P}) = \max(\mathcal{N}_\epsilon(\vec{P}, Q_{00}), \mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}, y_0))$,

where $\mathcal{N}_\epsilon(\vec{P}, Q_{00})$ is the number computed from $\epsilon, Q_{00}, \vec{P}$ by Algorithm 14.1, and $\mathcal{N}_\epsilon(\vec{P}', y_0)$ is the number computed from ϵ, $\vec{P}'$ by Algorithm 15.1 (with $\vec{P}' = \vec{P}|_{\{y_0\}}$ here).

By applying Algorithm 27.1 to ϵ, Q_{00}, S, we obtain a $Wh(S)$-descriptor η_1, such that:

(20) $\mathcal{N}_\epsilon(\vec{P}, Q_{00})$ differs from $\Lambda(\vec{P}, \eta_1)$ by at most a factor $(1 + C\epsilon)$, for any $\vec{P} \in Wh(S)$;

and

(21) $\mathsf{length}\,(\eta_1)$, $\dim(\eta_1) \leq \exp(C/\epsilon)$.

The work and storage used to compute η_1 are at most $\exp(C/\epsilon)$.

Also, by applying Algorithm 27.3 to ϵ, y_0, we obtain a $\mathcal{W}\mathcal{H}(\{y_0\})$-descriptor $\bar{\eta}_2$, such that:

(22) $\mathcal{N}_\epsilon(\vec{P}', y_0)$ differs from $\Lambda(\vec{P}', \bar{\eta}_2)$ by at most a factor $(1 + C\epsilon)$, for any $\vec{P}' \in \mathcal{W}\mathcal{H}(\{y_0\})$; and

(23) $\mathsf{length}\,(\bar{\eta}_2)$, $\dim(\bar{\eta}_2) \leq \exp(C/\epsilon)$.

The work and storage used to compute η_2 are at most $\exp(C/\epsilon)$. In (22), $\mathcal{N}_\epsilon(\vec{P}', y_0)$ denotes the number computed from ϵ, $\vec{P}'$ by Algorithm 15.1.

Note that

(24) $\#(S) \leq \exp(C/\epsilon)$,

thanks to (15), (16).

By applying Algorithm 26.1 to the $\mathcal{W}\mathcal{H}(\{y_0\})$-descriptor $\bar{\eta}_2$ and the linear map $\pi^S_{\{y_0\}} : \mathcal{W}\mathcal{H}(S) \to \mathcal{W}\mathcal{H}(\{y_0\})$, we compute a $\mathcal{W}\mathcal{H}(S)$-descriptor η_2, given by

(25) $\eta_2 = \bar{\eta}_2 \circ \pi^S_{\{y_0\}}$.

The work and storage used to apply Algorithm 26.1 are at most $\exp(C/\epsilon)$, thanks to (23), (24). Moreover, we have

(26) $\mathsf{length}\,(\eta_2) = \mathsf{length}\,(\bar{\eta}_2) \leq \exp(C/\epsilon)$ and $\dim(\eta_2) = \dim(\bar{\eta}_2) \leq \exp(C/\epsilon)$,

as we see from Algorithm 26.1 and (23).

Proposition 26.1 and (22) yield

(27) $\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}, y_0)$ differs from $\Lambda(\vec{P}, \eta_2)$ by at most a factor $(1 + C\epsilon)$, for any $\vec{P} \in \mathcal{W}\mathcal{H}(S)$.

From (19), (20), (27), we obtain the following result.

(28) $\mathcal{N}_\epsilon(\vec{P})$ differs from $\max(\Lambda(\vec{P}, \eta_1), \Lambda(\vec{P}, \eta_2))$ by at most a factor $(1 + C\epsilon)$, for any $\vec{P} \in \mathcal{W}\mathcal{H}(S)$.

In (28), $\mathcal{N}_\epsilon(\vec{P})$ denotes the number computed from $\epsilon, Q, \vec{P}$ by Algorithm 16.1.

We now compute the $\mathcal{W}\mathcal{H}(S)$-descriptor

(29) $\eta = \eta_1 \vee \eta_2$,

by applying Algorithm 26.2.

We have

$$(30)\quad \mathsf{length}\,(\eta) \;=\; \mathsf{length}\,(\eta_1) \;+\; \mathsf{length}\,(\eta_2) \;\le\; \exp(C/\epsilon)$$

and

$$(31)\quad \dim(\eta) \;=\; \dim(\eta_1) \;+\; \dim(\eta_2) \;\le\; \exp(C/\epsilon),$$

thanks to (21), (26) and Algorithm 26.2.

Moreover, Proposition 26.2 gives

$$(32)\quad \Lambda(\vec{P}, \eta_1 \vee \eta_2) \;=\; \max(\Lambda(\vec{P}, \eta_1),\, \Lambda(\vec{P}, \eta_2)), \text{ for any } \vec{P} \in \mathcal{W}\mathfrak{h}(S).$$

The work and storage used by Algorithm 26.2 are at most $\exp(C/\epsilon)$, thanks to (21), (26) and (24).

Comparing (28) with (32), we see that (17) holds for the descriptor η in (29). Moreover, (18) holds for that descriptor, as we see in (30), (31). Thus, η has the desired properties. Finally, the total work and storage used in all the above steps are at most $\exp(C/\epsilon)$.

This completes our explanation of Algorithm 27.5.

Algorithm 27.6. *Given ϵ, Q, S as in Algorithm* 27.5*, and given $\vec{P} \in \mathcal{W}\mathfrak{h}(S)$, we compute a $(1 + C\epsilon)$-optimal vector w^0 for $\vec{P}, \eta$, where η is the descriptor computed from ϵ, Q, S by Algorithm* 27.5*.*

The work and storage used to compute w^0 are at most $\exp(C/\epsilon)$*.*

Explanation: We repeat the explanation of Algorithm 27.5, and then continue as follows.

Applying Algorithm 27.2 to $\epsilon, Q, S, \vec{P}$, we compute a $(1 + C\epsilon)$-optimal vector w^1 for $\vec{P}, \eta_1$.

The work and storage used to compute w^1 are at most $\exp(C/\epsilon)$.

Next, applying Algorithm 27.4 to $\epsilon, y_0, \vec{P}|_{\{y_0\}}$, we compute a $(1 + C\epsilon)$-optimal vector w^2 for $\vec{P}|_{\{y_0\}}, \bar{\eta}_2$.

The work and storage used to compute w^2 are at most $\exp(C/\epsilon)$.

By Proposition 26.1, w^2 is also a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta_2$, thanks to (25). Now Proposition 26.2 shows that $w^0 = (w^1, w^2)$ is a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta_1 \vee \eta_2$. In view of (29), our vector w^0 is a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta$, where η is the descriptor computed from ϵ, Q, S by Algorithm 27.5. We have seen that the work and storage used to compute w^0 are at most $\exp(C/\epsilon)$.

This concludes our explanation of Algorithm 27.6.

Algorithm 27.7. *Given* $\epsilon > 0$*, assumed to be less than a small enough controlled constant; and given a cube* Q*, assumed to satisfy*

(33) $e^{-1/(2\epsilon)} \leq \delta_Q \leq c^{\#}\epsilon^{-1}$ *with* $c^{\#}$ *as in Algorithm* 17.1;

and given a set

(34) $S \subset Q^{**}$,

assumed to satisfy

(35) $|y - y'| > \epsilon^2\, e^{-2/\epsilon}\, \delta_Q$ *for any two distinct points* $y, y' \in S$*;*

we compute a $Wh(S)$*-descriptor* η*, such that:*

(36) *For any* $\vec{P} \in Wh(S)$*, the number* $\mathcal{N}_\epsilon(\vec{P})$ *computed from* $\epsilon, Q, \vec{P}$ *by Algorithm* 17.1 *differs by at most a factor of* $(1 + C\epsilon)$ *from* $\Lambda(\vec{P}, \eta)$*; and*

(37) $\mathsf{length}\,(\eta)$, $\dim(\eta) \leq \exp(C/\epsilon)$.

The work and storage used to compute η *are at most* $\exp(C/\epsilon)$.

Explanation: Obvious from Proposition 17.1 and Algorithm 27.1.

Algorithm 27.8. *Given* ϵ, Q, S *as in Algorithm* 27.7*, and given a Whitney field* $\vec{P} \in Wh(S)$*, we compute a* $(1 + C\epsilon)$*-optimal vector* w^0 *for* $\vec{P}, \eta$*, where* η *is the descriptor computed from* ϵ, Q, S *by Algorithm* 27.7.

The work and storage used to compute w^0 *are at most* $\exp(C/\epsilon)$.

Explanation: Obvious from Proposition 17.1 and Algorithm 27.2.

Algorithm 27.9. *Given* $\epsilon > 0$*, assumed to be less than a small enough controlled constant; and given a dyadic cube* Q*, assumed to satisfy*

(38) $\frac{1}{2}\, c^{\#}\, \epsilon^{-1} \leq \delta_Q \leq e^{10/\epsilon}$ *with* $c^{\#}$ *as in Algorithm* 17.1;

and given a set

(39) $S \subset Q^{**}$,

assumed to satisfy

(40) $|y - y'| > \epsilon^2\, e^{-2/\epsilon}\, \delta_Q$ *for any two distinct points* $y, y' \in S$*;*

we compute a $Wh(S)$*-descriptor* η*, with the following properties.*

(41) *For any* $\vec{P} \in Wh(S)$*, the number* $\mathcal{N}_\epsilon(\vec{P})$ *computed from* $\epsilon, Q, \vec{P}$ *by Algorithm* 18.1 *differs by at most a factor of* $(1 + C\epsilon)$ *from* $\Lambda(\vec{P}, \eta)$.

(42) *The length and dimension of* η *are at most* $\exp(C/\epsilon)$.

The work and storage used to compute η *are at most* $\exp(C/\epsilon)$.

Explanation: From (39), (40), we have

(43) $\#(S) \leq \exp(C/\epsilon)$.

We recall Proposition 18.1. Let $s, \mathcal{G}, Q_\nu^{\langle s \rangle}$ (all $\nu \in \mathcal{G}$) and S_ν (all $\nu \in \mathcal{G}$) be as in that proposition. In particular, we can compute these objects using work and storage at most $\exp(C/\epsilon)$, using only ϵ, Q, S (not $\vec{P}$).

According to Proposition 18.1, for any $\vec{P} \in \mathcal{W}h(S)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, Q, \vec{P}$ by Algorithm 18.1 is given by

(44) $\mathcal{N}_\epsilon(\vec{P}) = \max_{\nu \in \mathcal{G}} \mathcal{N}_\epsilon(\vec{P}|_{S\nu})$,

where $\mathcal{N}_\epsilon(\vec{P}|_{S_\nu})$ denotes the number computed from $\epsilon, Q_\nu^{\langle s \rangle}, \vec{P}|_{S\nu}$ by Algorithm 17.1.

For each $\nu \in \mathcal{G}$, we apply Algorithm 27.7 to $\epsilon, Q_\nu^{\langle s \rangle}, S_\nu$, to compute a $\mathcal{W}h(S_\nu)$-descriptor $\bar{\eta}_\nu$, with the following properties.

(45) For any $\vec{P}' \in \mathcal{W}h(S_\nu)$, the number $\mathcal{N}_\epsilon(\vec{P}')$ computed from $\epsilon, Q_\nu^{\langle s \rangle}$, $\vec{P}'$ by Algorithm 17.1 differs by at most a factor $(1 + C\epsilon)$ from $\Lambda(\vec{P}', \bar{\eta}_\nu)$.

(46) The length and dimension of $\bar{\eta}_\nu$ are at most $\exp(C/\epsilon)$.

The work and storage used to compute a single $\bar{\eta}_\nu$ are at most $\exp(C/\epsilon)$. Since $\#(\mathcal{G}) \leq \exp(C/\epsilon)$ by Proposition 18.1, we conclude that the total work and storage used to compute all the $\bar{\eta}_\nu (\nu \in \mathcal{G})$ are at most $\exp(C/\epsilon)$.

Next, for each $\nu \in \mathcal{G}$, we apply Algorithm 26.1 to the $\mathcal{W}h(S_\nu)$-descriptor $\bar{\eta}_\nu$ and the linear map $\pi^S_{S_\nu} : \mathcal{W}h(S) \to \mathcal{W}h(S_\nu)$, to compute a $\mathcal{W}h(S)$-descriptor η_ν, with the following properties.

(47) $\Lambda(\vec{P}, \eta_\nu) = \Lambda(\vec{P}|_{S_\nu}, \bar{\eta}_\nu)$ for any $\vec{P} \in \mathcal{W}h(S)$.

(48) The length and dimension of η_ν are at most $\exp(C/\epsilon)$.

(To derive (48), we use (46) and refer to Algorithm 26.1.)

For each $\nu \in \mathcal{G}$, the application of Algorithm 26.1 uses work and storage at most $\exp(C/\epsilon)$, thanks to (43) and (46). Since $\#(\mathcal{G}) \leq \exp(C/\epsilon)$, it follows that the total work and storage used to compute all the η_ν $(\nu \in \mathcal{G})$ from the corresponding $\bar{\eta}_\nu$ are at most $\exp(C/\epsilon)$.

From (44), (45), (47), we conclude that

(49) $\mathcal{N}_\epsilon(\vec{P})$ differs by at most a factor $(1 + C\epsilon)$ from $\max_{\nu \in \mathcal{G}} \Lambda(\vec{P}, \eta_\nu)$, for any $\vec{P} \in \mathcal{W}h(S)$;

where again $\mathcal{N}_\epsilon(\vec{P})$ denotes the number computed from $\epsilon, Q, \vec{P}$ by Algorithm 18.1.

We now apply Algorithm 26.2 to the family of $\mathcal{W}\mathcal{H}(S)$-descriptors η_ν ($\nu \in \mathcal{G}$). Thus, we compute the $\mathcal{W}\mathcal{H}(S)$ descriptor

$$\eta = \bigvee_{\nu \in \mathcal{G}} \eta_\nu, \tag{50}$$

using work and storage at most $\exp(C/\epsilon)$, as we see from (43), (48) and the fact that $\#(\mathcal{G}) \leq \exp(C/\epsilon)$.

By Proposition 26.2, the descriptor η in (50) has the following properties.

$$\Lambda(\vec{P}, \eta) = \max_{\nu \in \mathcal{G}} \Lambda(\vec{P}, \eta_\nu) \text{ for any } \vec{P} \in \mathcal{W}\mathcal{H}(S). \tag{51}$$

$$\text{length}\,(\eta) = \sum_{\nu \in \mathcal{G}} \text{length}\,(\eta_\nu) \leq \exp(C/\epsilon). \tag{52}$$

$$\dim(\eta) = \sum_{\nu \in \mathcal{G}} \dim(\eta_\nu) \leq \exp(C/\epsilon). \tag{53}$$

(Here again, we use (48) and the fact that $\#(\mathcal{G}) \leq \exp(C/\epsilon)$.)
Comparing (49) with (51), we learn that

(54) $\mathcal{N}_\epsilon(\vec{P})$ differs by at most a factor $(1 + C\epsilon)$ from $\Lambda(\vec{P}, \eta)$, for any $\vec{P} \in \mathcal{W}\mathcal{H}(S)$.

Again in (54), $\mathcal{N}_\epsilon(\vec{P})$ denotes the number computed from $\epsilon, Q, \vec{P}$ by Algorithm 18.1.

The desired properties (41), (42) of our descriptor η are equivalent to our results (52), (53), (54). Moreover, we have seen that the total work and storage used to compute η from ϵ, Q, S are at most $\exp(C/\epsilon)$.

This concludes our explanation of Algorithm 27.9.

Algorithm 27.10. *Given ϵ, Q, S as in Algorithm 27.9, and given a Whitney field $\vec{P} \in \mathcal{W}\mathcal{H}(S)$, we compute a $(1 + C\epsilon)$-optimal vector w^0 for $\vec{P}, \eta$, where η is the descriptor computed from ϵ, Q, S by Algorithm 27.9.*

The work and storage used to compute w^0 are at most $\exp(C/\epsilon)$.

Explanation: We repeat the explanation of Algorithm 27.9, and then continue as follows.

Applying Algorithm 27.8 to $\epsilon, Q_\nu^{\langle s \rangle}$, S_ν, $\vec{P}|_{S_\nu}$, we obtain, for each $\nu \in \mathcal{G}$, a $(1 + C\epsilon)$-optimal vector w^ν for $\vec{P}|_{S_\nu}$, $\bar{\eta}_\nu$, with $\bar{\eta}_\nu$ as in the explanation of Algorithm 27.9.

The work and storage used to compute a single w^ν are at most $\exp(C/\epsilon)$ (see Algorithm 27.8), and we know that $\#(\mathcal{G}) \leq \exp(C/\epsilon)$. Hence, the total work and storage used to compute all the $w^\nu (\nu \in \mathcal{G})$ are at most $\exp(C/\epsilon)$.

For each $\nu \in \mathcal{G}$, Proposition 26.1 and the definition of η_ν together show that w^ν is a $(1 + C\epsilon)$-optimal vector for $\vec{P}$, η_ν. Consequently, Proposition 26.2 shows that the vector

$$w^0 = (w^\nu)_{\nu \in \mathcal{G}}$$

is a $(1 + C\epsilon)$-optimal vector for $\eta = \bigvee_{\nu \in \mathcal{G}} \eta_\nu$. (See (50).)

The work and storage used to compute w^0 are at most $\exp(C/\epsilon)$.

This completes our explanation of Algorithm 27.10.

Algorithm 27.11. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a dyadic cube Q, assumed to satisfy*

(55) $\delta_Q \geq e^{5/\epsilon}$;

and given a set

(56) $S \subset Q^{**}$,

assumed to satisfy

(57) $|y - y'| > \epsilon^2 e^{-2/\epsilon} \delta_Q$ *for any two distinct points $y, y' \in S$;*

we compute a $\mathcal{W}h(S)$-descriptor η, with the following properties.

(58) *For any $\vec{P} \in \mathcal{W}h(S)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, Q, \vec{P}$ by Algorithm 19.1 differs by at most a factor of $(1 + C\epsilon)$ from $\Lambda(\vec{P}, \eta)$.*

(59) *The length and dimension of η are at most $\exp(C/\epsilon)$.*

The work and storage used to compute η are at most $\exp(C/\epsilon)$.

Explanation: By (56) and (57), we have

(60) $\#(S) \leq \exp(C/\epsilon)$.

Recall Proposition 19.1. Thus,

(61) For any $\vec{P} \in \mathcal{W}h(S)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, Q, \vec{P}$ by Algorithm 19.1 is equal to $\max\{\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}}) : y_0 \in S\}$,

where

(62) $\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})$ is the number computed from $\epsilon, \vec{P}|_{\{y_0\}}$ by Algorithm 15.1.

For each $y_0 \in S$, we apply Algorithm 27.3 to ϵ, y_0, to compute a $\mathcal{W}h(\{y_0\})$-descriptor $\bar{\eta}_{y_0}$, with the following properties.

(63) For any $\vec{P}' \in \mathcal{W}h(\{y_0\})$, the number $\mathcal{N}_\epsilon(\vec{P}')$ computed from $\epsilon, \vec{P}'$ by Algorithm 15.1 differs by at most a factor of $(1 + C\epsilon)$ from $\Lambda(\vec{P}', \bar{\eta}_{y_0})$.

(64) The length and dimension of $\bar{\eta}_{y_0}$ are at most $\exp(C/\epsilon)$.

The work and storage used to compute a single $\bar{\eta}_{y_0}$ are at most $\exp(C/\epsilon)$. Hence, by (60), the total work and storage used to compute all the $\bar{\eta}_{y_0}$ ($y_0 \in S$) are at most $\exp(C/\epsilon)$.

Next, for each $y_0 \in S$, we apply Algorithm 26.1 to the $\mathcal{W}\!h(\{y_0\})$-descriptor $\bar{\eta}_{y_0}$ and the linear map $\pi^S_{\{y_0\}} : \mathcal{W}\!h(S) \to \mathcal{W}\!h(\{y_0\})$.

Thus, for each $y_0 \in S$, we obtain a $\mathcal{W}\!h(S)$-descriptor η_{y_0}, with the following properties.

(65) For any $\vec{P} \in \mathcal{W}\!h(S)$, we have $\Lambda(\vec{P}|_{\{y_0\}}, \bar{\eta}_{y_0}) = \Lambda(\vec{P}, \eta_{y_0})$.

(66) The length and dimension of η_{y_0} are at most $\exp(C/\epsilon)$.

The work and storage used to compute a single η_{y_0} are at most $\exp(C/\epsilon)$, thanks to (60), (64), and our estimate for the work and storage used by Algorithm 26.1. Hence, by (60), the total work and storage used to compute all the $\eta_{y_0}(y_0 \in S)$ are at most $\exp(C/\epsilon)$.

From (61), (62), (63), (65) we obtain the following result.

(67) For any $\vec{P} \in \mathcal{W}\!h(S)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, Q, \vec{P}$ by Algorithm 19.1 differs by at most a factor of $(1+C\epsilon)$ from $\max\{\Lambda(\vec{P}, \eta_{y_0}) : y_0 \in S\}$.

We now apply Algorithm 26.2 to the family of $\mathcal{W}\!h(S)$-descriptors $\{\eta_{y_0} : y_0 \in S\}$. Thus, we compute the descriptor

$$\eta = \bigvee_{y_0 \in S} \eta_{y_0}, \tag{68}$$

and we know from Proposition 26.2 that

$$\max\{\Lambda(\vec{P}, \eta_{y_0}) : y_0 \in S\} = \Lambda(\vec{P}, \eta) \text{ for any } \vec{P} \in \mathcal{W}\!h(S). \tag{69}$$

Moreover,

(70) η has length and dimension at most $\exp(C/\epsilon)$,

thanks to (60), (66) and our formulas for the length and dimension of $\eta_1 \vee \cdots \vee \eta_A$ in Algorithm 26.2. From (60), (66) and Algorithm 26.2, we learn also that the work and storage used to compute η are at most $\exp(C/\epsilon)$.

The desired properties (58), (59) of η are immediate from (67), (69) and (70). We have seen that the above computation uses work and storage at most $\exp(C/\epsilon)$.

This completes our explanation of Algorithm 27.11.

Algorithm 27.12. *Given* ϵ, Q, S *as in Algorithm* 27.11*, and given a Whitney field* $\vec{P} \in Wh(S)$*, we compute a* $(1 + C\epsilon)$*- optimal vector* w^0 *for* $\vec{P}, \eta$*, where* η *is the* $Wh(S)$*-descriptor computed from* ϵ, Q, S *by Algorithm* 27.11*.*

The work and storage used to compute w^0 *are at most* $\exp(C/\epsilon)$*.*

Explanation: We repeat the explanation of Algorithm 27.11, and then add the following.

For each $y_0 \in S$, we apply Algorithm 27.4 to $\epsilon, y_0, \vec{P}|_{\{y_0\}}$, to compute a $(1 + C\epsilon)$-optimal vector w^{y_0} for $\vec{P}|_{\{y_0\}}$, $\bar{\eta}_{y_0}$ (with $\bar{\eta}_{y_0}$ as in the explanation of Algorithm 27.11).

According to our estimate for the work and storage of Algorithm 27.4, we can compute a single w^{y_0} using work and storage at most $\exp(C/\epsilon)$. Hence, by (60), the total work and storage used to compute all the w^{y_0} $(y_0 \in S)$ are at most $\exp(C/\epsilon)$.

Recall from our explanation of Algorithm 27.11 that $\eta_{y_0} = \bar{\eta}_{y_0} \circ \pi^S_{\{y_0\}}$ for each $y_0 \in S$. Hence, Proposition 26.1 shows that w^{y_0} is a $(1 + C\epsilon)$-optimal vector for $\vec{P}$, η_{y_0}, for any $y_0 \in S$.

Also, recall from our explanation of Algorithm 27.11 that $\eta = \bigvee_{y_0 \in S} \eta_{y_0}$, where η is the descriptor computed from ϵ, Q, S by that algorithm. Proposition 26.2 therefore shows that

$w^0 = (w^{y_0})_{y_0 \in S}$

is a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta$. We have computed w^0 using work and storage at most $\exp(C/\epsilon)$.

This completes our explanation of Algorithm 27.12.

Algorithm 27.13. *Given* $\epsilon > 0$*, assumed to be less than a small enough controlled constant; and given a dyadic cube* Q*; and given a set*

(71) $S \subset Q^{**}$*,*

assumed to satisfy

(72) $|y - y'| > \epsilon^2 e^{-2/\epsilon} \delta_Q$ *for any two distinct points* $y, y' \in Q$*;*

we compute a $Wh(S)$*-descriptor* η*, with the following properties.*

(73) *For any* $\vec{P} \in Wh(S)$*, the number* $\mathcal{N}_\epsilon(\vec{P})$ *computed from* $\epsilon, Q, \vec{P}$ *by Algorithm* 20.1 *differs by at most a factor of* $(1 + C\epsilon)$ *from* $\Lambda(\vec{P}, \eta)$*.*

(74) *The length and dimension of* η *are at most* $\exp(C/\epsilon)$*.*

The work and storage used to compute η *are at most* $\exp(C/\epsilon)$*.*

Explanation: We proceed by cases, depending on the size of δ_Q, as in our explanation of Algorithm 20.1.

This reduces matters to Algorithms 27.5, 27.7, 27.9, and 27.11.

Algorithm 27.14. *Given* ϵ, Q, S *as in Algorithm* 27.13*, and given a Whitney field* $\vec{P} \in Wh(S)$*, we compute a* $(1 + C\epsilon)$*-optimal vector* w^0 *for* $\vec{P}, \eta$*, where* η *is the* $Wh(S)$*-descriptor computed from* ϵ, Q, S *by Algorithm* 27.13.

The work and storage used to compute w^0 *are at most* $\exp(C/\epsilon)$.

Explanation: We proceed by cases, depending on the size of δ_Q, as in our explanations of Algorithms 20.1. and 27.13.

This reduces matters to Algorithms 27.6, 27.8, 27.10, and 27.12.

28. Minmax Functions and the Main Algorithm

In this section, we will see that the number $\mathcal{N}_\epsilon(\vec{P})$ computed by the Main Algorithm (Algorithm 21.1) is well-approximated by a minimax function. In addition, we compute a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta$, where η is the descriptor of that minimax function.

Algorithm 28.1. *Given* $\epsilon > 0$*, assumed to be less than a small enough controlled constant, and given a set* $E \subset \mathbb{R}^n$*, with* $\#(E) = N$*,* $2 \le N < \infty$*; we compute a* $Wh(E)$*-descriptor* η*, with the following properties.*

(1) *For any* $\vec{P} \in Wh(E)$*, the number* $\mathcal{N}_\epsilon(\vec{P})$ *computed from* $\epsilon, \vec{P}$ *by Algorithm* 21.1 *differs by at most a factor of* $(1 + C\epsilon)$ *from* $\Lambda(\vec{P}, \eta)$.

(2) *The length and dimension of* η *are at most* $\exp(C/\epsilon) \cdot N$.

Moreover, the computation of η *uses work and storage at most* $\exp(C/\epsilon)N^2$.

Explanation: We make some preliminary remarks. Recall from Section 26 that a V-descriptor η with length L and dimension D is stored as a matrix, requiring storage $C + C \cdot [\dim V + D] \cdot L$. In view of (2), it will take storage $\exp(C/\epsilon)N^2$ simply to hold the matrix representing η. Hence, it is natural to expect that the work and storage of our algorithm will be comparable to $\exp(C/\epsilon)N^2$. In fact, the matrix representing η is sparse, and almost all the work of Algorithm 28.1 consists of repeatedly writing the number zero. Unfortunately, it is not clear how to take advantage of this fact, once we apply Algorithm 28.1 below.

Let us begin our explanation of Algorithm 28.1. We recall Proposition 25.1, with conclusion (f) there replaced by the sharper conclusion (f′), as explained in Section 25.

We compute the list of cubes and sets $Q^{(\lambda)}$, $S^{(\lambda)}$, $\lambda = 1, \dots, L$, as in that proposition. Thus, the following hold.

(3) $L \le \frac{C}{\epsilon}N$.

(4) The total work and storage to compute all the $Q^{(\lambda)}$ and $S^{(\lambda)}$ ($\lambda = 1, \dots, L$) are at most $\exp(C/\epsilon)\, N \log N$ and $\exp(C/\epsilon)N$, respectively.

(5) For each $\lambda = 1, \dots, L$, the set $S^{(\lambda)} \subseteq E$ arises by applying Algorithm 11.1 to the input data ϵ, $Q^{(\lambda)}$, E.

(6) For any $\vec{P} \in \mathcal{W}h(E)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, \vec{P}$ by Algorithm 21.1 is given by

(7) $\mathcal{N}_\epsilon(\vec{P}) = \max\{\mathcal{N}_\epsilon(\vec{P}|_{S^{(\lambda)}})\ (1 \le \lambda \le L),\ \mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})\ (y_0 \in E)\}$, where

(8) $\mathcal{N}_\epsilon(\vec{P}|_{S^{(\lambda)}})$ is the number computed from $\epsilon, Q^{(\lambda)}, \vec{P}|_{S^{(\lambda)}}$ by Algorithm 20.1, and

(9) $\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})$ is the number computed from $\epsilon, \vec{P}|_{\{y_0\}}$ by Algorithm 15.1.

By (5) and the defining properties of Algorithm 11.1, we know that, for each $\lambda = 1, \dots, L$.

(10) $S^{(\lambda)} \subseteq E \cap (Q^{(\lambda)})^{**}$, and

(11) $|y - y'| \ge c\epsilon\, e^{-2/\epsilon}\, \delta_{Q^{(\lambda)}}$ for any distinct points $y, y' \in S^{(\lambda)}$.

Hence, $S^{(\lambda)}$ and $Q^{(\lambda)}$ satisfy conditions (71) and (72) in Section 27. Consequently, we may apply Algorithm 27.13 to $\epsilon, Q^{(\lambda)}$, $S^{(\lambda)}$, to compute a $\mathcal{W}h(S^{(\lambda)})$-descriptor $\bar{\eta}^{(\lambda)}$, with the following properties.

(12) For any $\vec{P}' \in \mathcal{W}h(S^{(\lambda)})$, the number $\mathcal{N}_\epsilon(\vec{P}')$ computed from $\epsilon, Q^{(\lambda)}$, $\vec{P}'$ by Algorithm 20.1 differs by at most a factor of $(1 + C\epsilon)$ from $\Lambda(\vec{P}', \bar{\eta}^{(\lambda)})$.

(13) The length and dimension of $\bar{\eta}^{(\lambda)}$ are at most $\exp(C/\epsilon)$.

We compute descriptors $\bar{\eta}^{(\lambda)}$ satisfying (12) and (13), for each $\lambda = 1, \dots, L$.

The work and storage used to compute a single $\bar{\eta}^{(\lambda)}$ are at most $\exp(C/\epsilon)$, as we see from Algorithm 27.13. Hence, the total work and storage used to compute all the $\bar{\eta}^{(\lambda)}$ ($\lambda = 1, \dots, L$) (given the $S^{(\lambda)}$ and $Q^{(\lambda)}$) are at most $\exp(C/\epsilon)N$; see (3).

Next, for each $\lambda = 1, \dots, L$, we apply Algorithm 26.1 to $\mathcal{W}h(S^{(\lambda)})$-descriptor $\bar{\eta}^{(\lambda)}$ and the linear map $\pi^E_{S^{(\lambda)}} : \mathcal{W}h(E) \to \mathcal{W}h(S^{(\lambda)})$. Thus, for each λ, we obtain a $\mathcal{W}h(E)$-descriptor $\eta^{(\lambda)}$, with the following properties.

(14) For any $\vec{P} \in \mathcal{W}h(E)$, $\Lambda(\vec{P}|_{S^{(\lambda)}}, \bar{\eta}^{(\lambda)}) = \Lambda(\vec{P}, \eta^{(\lambda)})$.

(15) The length and dimension of $\eta^{(\lambda)}$ are at most $\exp(C/\epsilon)$.

Moreover, the work and storage used to compute a single $\eta^{(\lambda)}$ from the corresponding $\bar{\eta}^{(\lambda)}$ are at most $\exp(C/\epsilon) \cdot N$. Hence, by (3), the total work and storage used to compute $\eta^{(1)}, \dots, \eta^{(L)}$ from $\bar{\eta}^{(1)}, \dots, \bar{\eta}^{(L)}$ are at most $\exp(C/\epsilon) \cdot N^2$.

Comparing (7), (8) with (12), (14), we learn that

(16) For any $\vec{P} \in \mathcal{W}\mathfrak{h}(E)$, the number $\mathcal{N}_\epsilon(\vec{P}|_{S^{(\lambda)}})$ in (7) differs by at most a factor of $(1 + C\epsilon)$ from $\Lambda(\vec{P}, \eta^{(\lambda)})$.

Property (16) holds for each $\lambda = 1, \ldots, L$.

Next, for each $y_0 \in E$, we apply Algorithm 27.3 (to ϵ and y_0), to compute a $\mathcal{W}\mathfrak{h}(\{y_0\})$-descriptor $\bar{\eta}^{(y_0)}$, with the following properties.

(17) For any $\vec{P}' \in \mathcal{W}\mathfrak{h}(\{y_0\})$, the number $\mathcal{N}_\epsilon(\vec{P}')$ computed from $\epsilon, \vec{P}'$ by Algorithm 15.1 differs by at most a factor $(1 + C\epsilon)$ from $\Lambda(\vec{P}', \bar{\eta}^{(y_0)})$.

(18) The length and dimension of $\bar{\eta}^{(y_0)}$ are at most $\exp(C/\epsilon)$.

For each $y_0 \in E$, the work and storage used to compute $\bar{\eta}^{(y_0)}$ are at most $\exp(C/\epsilon)$. Hence, the total work and storage used to compute all the $\bar{\eta}^{(y_0)}$ $(y_0 \in E)$ are at most $\exp(C/\epsilon)N$.

For each $y_0 \in E$, we now apply Algorithm 26.1 to the $\mathcal{W}\mathfrak{h}(\{y_0\})$-descriptor $\bar{\eta}^{(y_0)}$ and the linear map $\pi^E_{\{y_0\}} : \mathcal{W}\mathfrak{h}(E) \to \mathcal{W}\mathfrak{h}(\{y_0\})$.

Thus, for each $y_0 \in E$, we compute a $\mathcal{W}\mathfrak{h}(E)$-descriptor $\eta^{(y_0)}$, with the following properties.

(19) For any $\vec{P} \in \mathcal{W}\mathfrak{h}(E)$, $\Lambda(\vec{P}|_{\{y_0\}}, \bar{\eta}^{(y_0)}) = \Lambda(\vec{P}, \eta^{(y_0)})$.

(20) The length and dimension of $\eta^{(y_0)}$ are at most $\exp(C/\epsilon)$.

Moreover, for each fixed $y_0 \in E$, the work and storage used to compute $\eta^{(y_0)}$ from $\bar{\eta}^{(y_0)}$ are at most $\exp(C/\epsilon)N$. Consequently, the total work and storage used to compute all the $\eta^{(y_0)}$ $(y_0 \in E)$ from the $\bar{\eta}^{(y_0)}$ $(y_0 \in E)$ are at most $\exp(C/\epsilon) \cdot N^2$.

Comparing (7), (9) with (17), (19), we see that the following holds, for each $y_0 \in E$.

(21) For any $\vec{P} \in \mathcal{W}\mathfrak{h}(E)$, the number $\mathcal{N}_\epsilon(\vec{P}|_{\{y_0\}})$ in (7) differs by at most a factor of $(1 + C\epsilon)$ from $\Lambda(\vec{P}, \eta^{(y_0)})$.

From (6), (7), (16) and (21), we learn that

(22) For any $\vec{P} \in \mathcal{W}\mathfrak{h}(E)$, the number $\mathcal{N}_\epsilon(\vec{P})$ computed from $\epsilon, \vec{P}$ by Algorithm 21.1 differs by at most a factor of $(1 + C\epsilon)$ from $\max\{\Lambda(\vec{P}, \eta^{(\lambda)})$ $(\lambda = 1, \ldots, L)$, $\Lambda(\vec{P}, \eta^{(y_0)})$ $(y_0 \in E)\}$.

We now apply Algorithm 26.2 to compute the $\mathcal{W}\mathfrak{h}(E)$-descriptor

(23) $$\eta = (\eta^{(1)} \vee \cdots \vee \eta^{(L)}) \vee \bigvee_{y_0 \in E} \eta^{(y_0)}.$$

According to Proposition 26.2, we have

(24) $\Lambda(\vec{P},\eta) = \max\{\Lambda(\vec{P},\eta^{(\lambda)})\ (\lambda = 1,\dots,L),\ \Lambda(\vec{P},\eta^{(y_0)})\ (y_0 \in E)\}$ for any $\vec{P} \in Wh(E)$.

Moreover, that same proposition gives

$$\text{length}\,(\eta) = \sum_{\lambda=1}^{L} \text{length}\,(\eta^{(\lambda)}) + \sum_{y_0\in E} \text{length}\,(\eta^{(y_0)}) \text{ and}$$

$$\dim(\eta) = \sum_{\lambda=1}^{L} \dim(\eta^{(\lambda)}) + \sum_{y_0\in E} \dim(\eta^{(y_0)})\,.$$

Thanks to (15), (20) and (3), it follows that

(25) The length and dimension of η are at most $\exp(C/\epsilon)N$.

Also, the estimate for work and storage given in Algorithm 26.2 shows that it takes work and storage at most $\exp(C/\epsilon)N^2$ to compute η from (23).

The desired properties (1) and (2) of the descriptor η are now immediate from (22), (24), (25). We have computed η using total work and storage at most $\exp(C/\epsilon)N^2$.

This completes our explanation of Algorithm 28.1.

Algorithm 28.2. *Given ϵ, E, N as in Algorithm* 28.1*, and given a Whitney field $\vec{P} \in Wh(E)$, we compute a $(1+C\epsilon)$-optimal vector w^0 for $\vec{P},\eta$, where η is the $Wh(E)$-descriptor computed from ϵ, E by Algorithm* 28.1.

The work and storage used to compute w^0 are at most $\exp(C/\epsilon)N^2$.

Explanation: We repeat our explanation of Algorithm 28.1, and then continue as follows.

For each $\lambda = 1,\dots,L$, we apply Algorithm 27.14 to $\epsilon, Q^{(\lambda)}, S^{(\lambda)}, \vec{P}|_{S^{(\lambda)}}$. This produces a $(1+C\epsilon)$-optimal vector $w^{(\lambda)}$ for $\vec{P}|_{S^{(\lambda)}}, \bar{\eta}^{(\lambda)}$, with $\bar{\eta}^{(\lambda)}$ as in the explanation of Algorithm 28.1.

The work and storage used to compute a single $w^{(\lambda)}$ are at most $\exp(C/\epsilon)$, as we see from Algorithm 27.14. Hence, by (3), the total work and storage used to compute all the $w^{(\lambda)}$ $(\lambda = 1,\dots,L)$ are at most $\exp(C/\epsilon)N$.

For each $\lambda = 1,\dots,L$, we recall that $\eta^{(\lambda)} = \bar{\eta}^{(\lambda)} \circ \pi^E_{S^{(\lambda)}}$. Consequently, Proposition 26.1 shows that

(26) For each $\lambda = 1,\dots,L$, $w^{(\lambda)}$ is a $(1+C\epsilon)$-optimal vector for $\vec{P}, \eta^{(\lambda)}$.

Next, for each $y_0 \in E$, we apply Algorithm 27.4 to $\epsilon, y_0, \vec{P}|_{\{y_0\}}$. This produces a $(1+C\epsilon)$-optimal vector $w^{(y_0)}$ for $\vec{P}|_{\{y_0\}}, \bar{\eta}^{(y_0)}$, with $\bar{\eta}^{(y_0)}$ as in the explanation of Algorithm 28.1.

The work and storage used to compute a single $w^{(y_0)}$ are at most $\exp(C/\epsilon)$, as we see from Algorithm 27.4. Hence, the total work and storage used to compute all the $w^{(y_0)}$ $(y_0 \in E)$ are at most $\exp(C/\epsilon)\cdot N$.

For each $y_0 \in E$, we recall that $\eta^{(y_0)} = \bar{\eta}^{(y_0)} \circ \pi^E_{\{y_0\}}$. Consequently, Proposition 26.1 shows that

(27) For each $y_0 \in E$, $w^{(y_0)}$ is a $(1 + C\epsilon)$-optimal vector for $\vec{P}$, $\eta^{(y_0)}$.

Now (23), (26), (27) and Proposition 26.2 together show that

$$w^0 = (w^{(1)}, \ldots, w^{(L)}, (w^{(y_0)})_{y_0 \in E})$$

is a $(1 + C\epsilon)$-optimal vector for $\vec{P}, \eta$, with η as in Algorithm 28.1.

We have seen that the work and storage used to compute w^0 are most $\exp(C/\epsilon)N$, plus the work and storage needed to repeat Algorithm 28.1. Thus, our algorithm uses total work and storage at most $\exp(C/\epsilon)N^2$.

This completes our explanation of Algorithm 28.2.

Remark 28.1. *To carry out Algorithm* 28.2*, we needn't repeat all the steps in Algorithm* 28.1*. This allows us easily to reduce the work and storage of Algorithm* 28.2 *to* $\exp(C/\epsilon)N \log N$ *and* $\exp(C/\epsilon)N$*, respectively. Unfortunately, these improvements will not help us when we apply Algorithm* 28.2 *in the next section.*

29. From Whitney Fields to Functions

In this section, we pass from Whitney fields to functions, and give the proof of Theorem 2. We begin by introducing some definitions and notation.

Given a finite set $E \subset \mathbb{R}^n$, we write $\mathcal{Fns}(E)$ to denote the vector space of all functions $f : E \to \mathbb{R}$. We identify $f \in \mathcal{Fns}(E)$ with the Whitney field $[f] = (\tilde{P}^x)_{x \in E}$, where for each $x \in E$, $\tilde{P}^x$ denotes the constant polynomial $\tilde{P}^x(y) = f(x)$ (all $y \in \mathbb{R}^n$). Thus, $\mathcal{Fns}(E)$ is identified with a subspace of $\mathcal{Wh}(E)$. Also, we write $\mathcal{Wh}_0(E)$ to denote the space of all Whitney fields $\vec{P} = (P^x)_{x \in E} \in \mathcal{Wh}(E)$ such that $P^x(x) = 0$ for all $x \in E$.

Thus,

(1) $\mathcal{Wh}(E) = \mathcal{Fns}(E) \oplus \mathcal{Wh}_0(E)$.

In this section we write $\vec{P}$ to denote an element of $\mathcal{Wh}_0(E)$, f to denote an element of $\mathcal{Fns}(E)$, and (f, P) to denote an element of $\mathcal{Wh}(E)$, as in (1).

Let us record some of our earlier definitions and results using the above notation.

Let $(f, \vec{P}) \in \mathcal{Wh}(E)$, with $\vec{P} = (P^x)_{x \in E} \in \mathcal{Wh}_0(E)$. Then the C^m-norm of (f, P) is given by

(2) $\| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)} = \inf \{ \| F \|_{C^m(\mathbb{R}^n)}$:
$F \in C^m(\mathbb{R}^n)$, $F = f$ on E, $\partial^\alpha F(x) = \partial^\alpha P^x(x)$ for $0 < |\alpha| \leq m$, $x \in E\}$.

For $f \in Fns(E)$, the C^m-norm is given by

$$\| f \|_{C^m(\mathbb{R}^n)} = \inf\{\| F \|_{C^m(\mathbb{R}^n)} : F \in C^m(\mathbb{R}^n),\ F = f \text{ on } E\}. \tag{3}$$

Comparing (2) and (3), we see that

$$\| f \|_{C^m(\mathbb{R}^n)} = \inf\{\| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)} : \vec{P} \in Wh_0(E)\}, \tag{4}$$

for any $f \in Fns(E)$.

In terms of our new notation, Algorithm 21.1 takes the following form.

Algorithm 29.1. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a Whitney field $(f, \vec{P}) \in Wh(E)$, with $\vec{P} = (P^x)_{x \in E} \in Wh_0(E)$, and with $\#(E) = N$, $2 \le N < \infty$; we compute a number $\mathcal{N}_\epsilon(f, \vec{P}) \ge 0$, and a function $F \in C^m(\mathbb{R}^n)$, with the following properties.*

(5) $F = f$ *on* E.

(6) $\partial^\alpha F(x) = \partial^\alpha P^x(x)$ *for* $0 < |\alpha| \le m$, $x \in E$.

(7) $\| F \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(f, \vec{P})$.

(8) $\mathcal{N}_\epsilon(f, \vec{P}) \le (1 + C\epsilon)\ \| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)}$.

Moreover,

(9) *The one-time work of the algorithm is at most* $\exp(C/\epsilon)\, N \log N$, *the query work is at most* $C \log(N/\epsilon)$, *and the storage used is at most* $\exp(C/\epsilon) N$.

Note that (5)...(8) and (2) imply

$$(1 - C\epsilon)\ \| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)} \le \mathcal{N}_\epsilon(f, \vec{P}) \le (1 + C\epsilon)\ \| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)}, \tag{10}$$

and

$$\| F \|_{C^m(\mathbb{R}^n)} \le (1 + C\epsilon) \cdot \| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)}. \tag{11}$$

Next, we prepare to express Algorithms 28.1 and 28.2 in our new notation. A $Wh(E)$-descriptor η takes the form

$$\eta = (Fns(E) \oplus Wh_0(E), W, \lambda_1, \dots, \lambda_I), \tag{12}$$

where W is a finite-dimensional vector space, and $\lambda_1, \dots, \lambda_I : Fns(E) \oplus Wh_0(E) \oplus W \to \mathbb{R}$ are linear functionals. We write $(f, \vec{P}, w)$ to denote an element of $Fns(E) \oplus Wh_0(E) \oplus W$, and we write $\lambda_i(f, \vec{P}, w)$ to denote the value of the linear functional λ_i applied to the vector $(f, \vec{P}, w)$.
Let η be a $Wh(E)$-descriptor in the form (12).

Then, for $(f, \vec{P}) \in Fns(E) \oplus Wh_0(E)$, we have

$$\Lambda((f, \vec{P}), \eta) = \min_{w \in W} \max_{i=1,\dots,I} |\lambda_i(f, \vec{P}, w)|. \tag{13}$$

Some of our computations will involve $\mathcal{Fns}(E)$-descriptors $\check{\eta}$. We identify $\mathcal{Fns}(E)$ with $\mathbb{R}^N$ ($N = \#(E)$), by taking the co-ordinates of $f \in \mathcal{Fns}(E)$ to be the function values $f(x)$ (all $x \in E$). Thus, a $\mathcal{Fns}(E)$-descriptor $\check{\eta}$ can be stored in the computer as a matrix, as described in Section 26.

The reformulations of Algorithms 28.1 and 28.2 in our new notation are as follows.

Algorithm 29.2. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given a set $E \subset \mathbb{R}^n$, with $\#(E) = N$, $2 \leq N < \infty$; we compute a $\mathcal{Wh}(E)$-descriptor η in the form* (12)*, with the following properties.*

(14) *Let $(f, \vec{P}) \in \mathcal{Wh}(E)$. Then the number $\mathcal{N}_\epsilon(f, \vec{P})$ computed from $\epsilon, (f, \vec{P})$ by Algorithm* 29.1 *differs by at most a factor of $(1+C\epsilon)$ from $\Lambda((f, \vec{P}), \eta)$; see* (13).

(15) *In* (12)*, we have $I \leq \exp(C/\epsilon)N$ and $\dim W \leq \exp(C/\epsilon)N$.*

Moreover,

(16) *The work and storage used to compute η are at most $\exp(C/\epsilon) \cdot N^2$.*

Algorithm 29.3. *Given ϵ, E, N as in Algorithm* 29.2*, and given a Whitney field $(f, \vec{P}) \in \mathcal{Wh}(E)$, we compute a $(1 + C\epsilon)$-optimal vector $w^0 \in W$ for $(f, \vec{P})$ and η, where η is the $\mathcal{Wh}(E)$-descriptor computed from ϵ, E by Algorithm* 29.2*.*

The work and storage used to compute w^0 are at most $\exp(C/\epsilon) \cdot N^2$.

As a simple consequence of Theorem 4 from the introduction (see Fefferman-Klartag [22, 23]), we have the following algorithm.

Algorithm 29.4. *Given $f \in \mathcal{Fns}(E)$, with $E \subset \mathbb{R}^n$, $\#(E) = N$, $2 \leq N < \infty$, we compute $\vec{P}^\# \in \mathcal{Wh}_0(E)$ such that*

(17) $\| (f, \vec{P}^\#) \|_{C^m(\mathbb{R}^n)} \leq C \| f \|_{C^m(\mathbb{R}^n)}$.

The computation of $\vec{P}^\#$ uses work at most $CN\log N$, and storage at most CN.

Explanation: By Theorem 4, we can compute a function $F \in C^m(\mathbb{R}^n)$, such that $F = f$ on E, and $\| F \|_{C^m(\mathbb{R}^n)} \leq C \| f \|_{C^m(\mathbb{R}^n)}$. The computation of F involves one-time work at most $CN \log N$, storage at most CN, and query work at most $C \log N$.

We take

$$\vec{P}^\# = (P^\#_x)_{x \in E}, \text{ where } P^\#_x = J_x(F) - f(x) \text{ for } x \in E.$$

Thus, $\vec{P}^{\#} \in \mathcal{Wh}_0(E)$, and the Whitney field $(f, \vec{P}^{\#})$ agrees with F, in the sense that $F(x) = f(x)$ for $x \in E$, and

$$\partial^\alpha F(x) = \partial^\alpha P^{\#}_x(x) \quad \text{for } 0 < |\alpha| \leq m\,,\ x \in E\,.$$

Hence, (2) yields $\| (f, \vec{P}^{\#}) \|_{C^m(\mathbb{R}^n)} \leq \| F \|_{C^m(\mathbb{R}^n)} \leq C \| f \|_{C^m(\mathbb{R}^n)}$. This proves (17). The work and storage of our algorithm are as asserted.

This completes our explanation of Algorithm 29.4.

Algorithm 29.5. *Given $\epsilon > 0$, assumed to be less than a small enough controlled constant; and given $f \in \mathcal{Fns}(E)$, with $E \subset \mathbb{R}^n$, $\#(E) = N$, $2 \leq N < \infty$; we compute a number $\mathcal{N}_\epsilon(f) \geq 0$ and a function $F \in C^m(\mathbb{R}^n)$, with the following properties.*

(18) $F = f$ *on* E.

(19) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \mathcal{N}_\epsilon(f)$.

(20) $\mathcal{N}_\epsilon(f) \leq (1 + C\epsilon) \cdot \| f \|_{C^m(\mathbb{R}^n)}$.

The algorithm uses one-time work at most $\exp(C/\epsilon)N^5(\log N)^2$*, query work at most* $C \log(N/\epsilon)$*, and storage at most* $\exp(C/\epsilon)N^2$.

Explanation: The algorithm proceeds in several steps.

Step 1: Applying Algorithm 29.2 to ϵ, E, we compute a $\mathcal{Wh}(E)$-descriptor

(21) $\eta = (\mathcal{Fns}(E) \oplus \mathcal{Wh}_0(E), W, \lambda_1, \ldots, \lambda_I)$,

satisfying (14) and (15).

The work and storage of Step 1 are at most $\exp(C/\epsilon)N^2$, by (16).

In (21), each λ_i is a linear functional on $[\mathcal{Fns}(E) \oplus \mathcal{Wh}_0(E)] \oplus W$. We may instead regard λ_i as a linear functional on $\mathcal{Fns}(E) \oplus [\mathcal{Wh}_0(E) \oplus W]$.

This allows us to carry out the next step below.

Step 2: We compute the $\mathcal{Fns}(E)$-descriptor $\check{\eta}$, defined by

(22) $\check{\eta} = (\mathcal{Fns}(E), \mathcal{Wh}_0(E) \oplus W, \lambda_1, \ldots, \lambda_I)$.

Thanks to (15), we see easily that

(23) The length and dimension of $\check{\eta}$ are at most $\exp(C/\epsilon)N$.

Moreover, the task of computing $\check{\eta}$ from η is trivial; it requires work and storage at most $\exp(C/\epsilon)N^2$.

Let us discuss the relationship of η and $\check{\eta}$ to C^m-norms.
For any $\vec{P} \in \mathcal{Wh}_0(E)$, (10) and (14) together show that

$$(24)\quad (1 - C\epsilon) \| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)} \leq \min_{w \in W} \max_{i=1,\ldots,I} |\lambda_i(f, \vec{P}, w)| \leq (1 + C\epsilon) \| (f, \vec{P}) \|_{C^m(\mathbb{R}^n)}\,.$$

(Here we have also used (13).)

Taking the infimum over $\vec{P}$ in (24), and recalling (4), we see that

$$(25)\quad (1-C\epsilon)\parallel f\parallel_{C^m(\mathbb{R}^n)} \le \inf_{(\vec{P},w)\in \mathcal{WH}_0(E)\oplus W} \max_{i=1,\dots,I} |\lambda_i(f,\vec{P},w)| \le (1+C\epsilon)\parallel f\parallel_{C^m(\mathbb{R}^n)}.$$

That is,

$$(26)\quad (1-C\epsilon)\parallel f\parallel_{C^m(\mathbb{R}^n)} \le \Lambda(f,\check{\eta}) \le (1+C\epsilon)\parallel f\parallel_{C^m(\mathbb{R}^n)},$$

thanks to (22) and the definition of $\Lambda(f,\check{\eta})$.

In view of (26), we would like to compute $\Lambda(f,\check{\eta})$ up to a factor $(1+C\epsilon)$, using Algorithm 26.3.

One of the inputs to that algorithm is a vector, assumed to satisfy condition (†) in Section 26. To produce such a vector, we proceed as follows.

Step 3: Applying Algorithm 29.4 to f, we compute $\vec{P}^{\#} \in \mathcal{WH}_0(E)$ such that

$$(27)\quad \parallel (f,\vec{P}^{\#})\parallel_{C^m(\mathbb{R}^n)} \le C\parallel f\parallel_{C^m(\mathbb{R}^n)}.$$

The work and storage used for Step 3 are at most $CN\log N$ and CN, respectively.

Step 4: Applying Algorithm 29.3 to ϵ, E, N and $(f,\vec{P}^{\#})$, we compute a $(1+C\epsilon)$-optimal vector $w^{\#}\in W$ for $(f,\vec{P}^{\#}),\eta$. Thus,

$$(28)\quad \max_{i=1,\dots,I} |\lambda_i(f,\vec{P}^{\#},w^{\#})| \le (1+C\epsilon)\cdot \min_{w\in W}\max_{i=1,\dots I} |\lambda_i(f,\vec{P}^{\#},w)|.$$

The work and storage used for Step 4 are at most $\exp(C/\epsilon)N^2$.

Observe that

$$(29)\quad \max_{i=1,\dots,I} |\lambda_i(f,\vec{P}^{\#},w^{\#})| \le (1+C\epsilon)\cdot \min_{w\in W}\max_{i=1,\dots,I} |\lambda_i(f,P^{\#},w)| \text{ (by (28))}$$

$$\le (1+C'\epsilon)\parallel (f,\vec{P}^{\#})\parallel_{C^m(\mathbb{R}^n)} \text{ (by (24))}$$

$$\le C\parallel f\parallel_{C^m(\mathbb{R}^n)} \text{ (by (27)).}$$

On the other hand, for any $\vec{P}\in \mathcal{WH}_0(E)$, we have

$$\parallel f\parallel_{C^m(\mathbb{R}^n)} \le \parallel (f,\vec{P})\parallel_{C^m(\mathbb{R}^n)} \text{ (by (4))}$$

$$\le (1+C\epsilon)\cdot \min_{w\in W}\max_{i=1,\dots,I} |\lambda_i(f,\vec{P},w)| \text{ (by (24)).}$$

Consequently,

$$(30)\quad \parallel f\parallel_{C^m(\mathbb{R}^n)} \le (1+C\epsilon)\cdot \min_{(\vec{P},w)\in \mathcal{WH}_0(E)\oplus W} \max_{i=1,\dots,I} |\lambda_i(f,\vec{P},w)|.$$

(The minimum is achieved, thanks to an elementary remark from Section 26.)

Combining (29) and (30), we learn that

$$(31)\quad \max_{i=1,\dots,I} |\lambda_i(f,\vec{P}^{\#},w^{\#})| \le C \min_{(\vec{P},w)\in \mathcal{W}\mathcal{H}_0(E)\oplus W} \max_{i=1,\dots,I} |\lambda_i(f,\vec{P},w)|.$$

Let us compare (22) and (31) with condition (†) in the statement of Algorithm 26.3. We find that the vector $(\vec{P}^{\#},w^{\#})$ satisfies (†) for the list of affine functions $\mathcal{W}\mathcal{H}_0(E)\oplus W \ni (\vec{P},w)\mapsto \lambda_i(f,\vec{P},w)$ $(i=1,\dots,I)$, with Γ in (†) equal to C in (31).

Thus, we are in position to apply Algorithm 26.3.

Step 5: We apply Algorithm 26.3 to the list of affine functions $(\vec{P},w)\mapsto \lambda_i(f,\vec{P},w)$ $(i=1,\dots,I)$, with $(\vec{P}^{\#},w^{\#})$ here playing the rôle of the vector w^0 in Algorithm 26.3. We recall that we can take $\Gamma = C$ in Algorithm 26.3.

Thus, we compute a vector

$$(32)\quad (\vec{P}^0,w^0)\in \mathcal{W}\mathcal{H}_0(E)\oplus W,$$

such that

$$(33)\quad \max_{i=1,\dots,I} |\lambda_i(f,\vec{P}^0,w^0)| \le (1+C\epsilon)\cdot \min_{(\vec{P},w)\in \mathcal{W}\mathcal{H}_0(E)\oplus W} \max_{i=1,\dots,I} |\lambda_i(f,\vec{P},w)|.$$

Thanks to (23), we have $I,D\le \exp(C/\epsilon)N$ in Algorithm 26.3. Therefore, the work consumed by Step 4 is at most $\exp(C/\epsilon)N^5(\log N)^2$, while the storage used is at most $\exp(C/\epsilon)N^2$. (These quantities dominate the work and storage used in Algorithm 29.5. Thus, virtually all the work goes into solving one big linear programming problem.)
We now observe that

$$\begin{aligned}
\|f\|_{C^m(\mathbb{R}^n)} &\le \|(f,\vec{P}^0)\|_{C^m(\mathbb{R}^n)} && \text{(by (4))}\\
&\le (1+C\epsilon)\cdot \min_{w\in W}\max_{i=1,\dots,I} |\lambda_i(f,\vec{P}^0,w)| && \text{(by (24))}\\
&\le (1+C\epsilon)\cdot \max_{i=1,\dots,I} |\lambda_i(f,\vec{P}^0,w^0)|\\
&\le (1+C'\epsilon)\cdot \min_{(\vec{P},w)\in \mathcal{W}\mathcal{H}_0(E)\oplus W}\max_{i=1,\dots,I} |\lambda_i(f,\vec{P},w)| && \text{(by (33))}\\
&\le (1+C''\epsilon)\cdot \|f\|_{C^m(\mathbb{R}^n)} && \text{(by (25)).}
\end{aligned}$$

In particular,

$$(34)\quad \|f\|_{C^m(\mathbb{R}^n)} \le \|(f,\vec{P}^0)\|_{C^m(\mathbb{R}^n)} \le (1+C''\epsilon)\cdot \|f\|_{C^m(\mathbb{R}^n)}.$$

This reduces matters to Algorithm 29.1. We proceed as follows.

Step 6: We apply Algorithm 29.1 to ϵ and $(f, \vec{P}^0)$.

Thus, we compute a number $\mathcal{N}_\epsilon(f, \vec{P}^0) \geq 0$, and a function $F \in C^m(\mathbb{R}^n)$, with the following properties.

(35) $F = f$ on E.

(36) $\partial^\alpha F(x) = \partial^\alpha P^{0,x}(x)$ for $0 < |\alpha| \leq m$, $x \in E$; where $\vec{P}^0 = (P^{0,x})_{x \in E}$.

(37) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon)\, \mathcal{N}_\epsilon(f, \vec{P}^0)$.

(38) $\mathcal{N}_\epsilon(f, \vec{P}^0) \leq (1 + C\epsilon) \| (f, \vec{P}^0) \|_{C^m(\mathbb{R}^n)}$.

The one-time work of Step 6 is at most $\exp(C/\epsilon) N \log N$, while the query work is at most $C \log(N/\epsilon)$, and the storage used is at most $\exp(C/\epsilon) \cdot N$. Finally, we mop up as follows.

Step 7: We set $\mathcal{N}_\epsilon(f) = \mathcal{N}_\epsilon(f, \vec{P}^0)$, and take F as in Step 6. Let us check that $\mathcal{N}_\epsilon(f)$ and F have the desired properties (18), (19), (20). In fact, we have already proven (18); see (35). Also, (19) is immediate from (37), since we have just set $\mathcal{N}_\epsilon(f) = \mathcal{N}_\epsilon(f, \vec{P}^0)$. To check (20), we note that

$$\begin{aligned} \mathcal{N}_\epsilon(f) = \mathcal{N}_\epsilon(f, \vec{P}^0) &\leq (1 + C\epsilon) \| (f, \vec{P}^0) \|_{C^m(\mathbb{R}^n)} \\ &\leq (1 + C'''\epsilon) \| f \|_{C^m(\mathbb{R}^n)}, \end{aligned}$$

by (38) and (34). Thus, (18), (19), (20) hold.

We have seen that the one-time work of Steps 1...7 above is at most $\exp(C/\epsilon) N^5 (\log N)^2$, while the query work (which occurs only in Step 6) is at most $C \log(N/\epsilon)$, and the storage used is at most $\exp(C/\epsilon) N^2$.

This completes our explanation of Algorithm 29.5.

Note that (18), (19), (20) and the definition of $\| f \|_{C^m(\mathbb{R}^n)}$ show that

(39) $(1 - C\epsilon) \| f \|_{C^m(\mathbb{R}^n)} \leq \mathcal{N}_\epsilon(f) \leq (1 + C\epsilon) \| f \|_{C^m(\mathbb{R}^n)}$,

and

(40) $\| F \|_{C^m(\mathbb{R}^n)} \leq (1 + C\epsilon) \cdot \| f \|_{C^m(\mathbb{R}^n)}$, $F = f$ on E.

Thus, (39) and (40) hold for the number $\mathcal{N}_\epsilon(f)$ and the function $F \in C^m(\mathbb{R}^n)$ computed by Algorithm 29.5.

Theorem 2 from the introduction is now obvious; in the case $\#(E) \geq 2$, we just apply Algorithm 29.5 to ϵ', f, with $\epsilon' = c \cdot \min(\epsilon, 1)$ for a small enough c.

The case $\#(E) = 1$ follows; details are left to the reader.

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 subsets of $\mathbb{R}^n$. In *Linear and Complex Analysis*, 279–281. Lect. Notes in Math., Springer-Verlag, 1994.

[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] Dyer, M.: A class of convex programs with applications to computational geometry. In *Proceedings of the Eighth Annual Symposium on Computational Geometry (1992)*, 9–15.

[13] Fefferman, C.: Interpolation and extrapolation of smooth functions by linear operators. *Rev. Mat. Iberoamericana* **21** (2005), no. 1, 313–348.

[14] Fefferman, C.: A sharp form of Whitney's extension theorem. *Ann. of Math. (2)* **161** (2005), 509–577.

[15] Fefferman, C.: Whitney's extension problem for C^m. *Ann. of Math. (2)* **164** (2006), no. 1, 313–359.

[16] Fefferman, C.: Whitney's extension problem in certain function spaces. (preprint).

[17] Fefferman, C.: A generalized sharp Whitney theorem for jets. *Rev. Mat. Iberoamericana* **21** (2005), no. 2, 577–688.

[18] Fefferman, C.: Extension of $C^{m,\omega}$ smooth functions by linear operators. *Rev. Mat. Iberoamericana* **25** (2009), no. 1, 1–48.

[19] Fefferman, C.: C^m extension by linear operators *Ann. of Math. (2)* **166** (2007), no. 3, 779–835.

[20] Fefferman, C. and Klartag, B.: Fitting a C^m-smooth function to data III. *Annals of Math. (2)*, to appear.

[21] Fefferman, C.: The C^m norm of a function with prescribed jets I, preprint.

[22] Fefferman, C. and Klartag, B.: Fitting a C^m-smooth function to data I. *Annals of Math. (2)*, to appear.

[23] Fefferman, C. and Klartag, B.: Fitting a C^m-smooth function to data II. *Rev. Mat. Iberoamericana* **25** (2009), no. 1, 49–273.

[24] Fefferman, C. and Klartag, B.: An example related to Whitney extension with almost minimal C^m norm. *Rev. Mat. Iberoamericana* **25** (2009), no. 2, 423–446.

[25] Glaeser, G.: Étude de quelques algèbres tayloriennes. *J. Analyse Math.* **6** (1958), 1–124.

[26] 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.

[27] John, F.: Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday*, 187–204. Interscience Publishers, Inc., New York, 1948.

[28] Karmarkar, N.: A new polynomial-time algorithm for linear programming. *Combinatorica* **4** (1984), 302–311.

[29] Khachiyan, L. G.: A polynomial-time algorithm in linear programming. *Dokl. Akad. Nauk SSSR* **244** (1979), 1093–1096; translated in *Soviet Math. Dokl.* **20** (1979), 191–194.

[30] Knuth, D.: *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd edition. Addison-Wesley, 1997.

[31] Malgrange, B.: *Ideals of Differentiable Functions*. Oxford Univ. Press, 1966.

[32] Megiddo, N.: *Linear programming in linear time when the dimension is fixed. J. ACM* **31** (1984), no. 1, 114–127.

[33] Preparata, F. P. and Shamos, M. I.: *Computational Geometry: An introduction*, 2nd edition. Texts and Monographs in Computer Science. Springer-Verlag, New York, 1985.

[34] 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.

[35] 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.

[36] 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.

[37] Shvartsman, P.: Lipschitz selections of set-valued functions and Helly's theorem. *J. Geom. Anal.* **12** (2002), no. 2, 289–324.

[38] Stein, E. M.: *Singular Integrals and Differentiability Properties of Functions*. Princeton Univ. Press, 1970.

[39] 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.

[40] Webster, R.: *Convexity*. Oxford Science Publications, 1994.

[41] Whitney, H.: Analytic extensions of differentiable functions defined in closed sets. *Trans. Amer. Math. Soc.* **36** (1934), 63–89.

[42] Whitney, H.: Differentiable functions defined in closed sets I. *Trans. Amer. Math. Soc.* **36** (1934), 369–389.

[43] Whitney, H.: Functions differentiable on the boundaries of regions. *Ann. of Math.* **35** (1934), 482–485.

[44] Zobin, N.: Whitney's problem on extendability of functions and an intrinsic metric. *Adv. Math.* **133** (1998), no. 1, 96–132.

[45] Zobin, N.: Extension of smooth functions from finitely connected planar domains. *J. Geom. Anal.* **9** (1999), no. 3, 489–509.

Recibido: 30 de mayo de 2007

Charles Fefferman
Department of Mathematics
Princeton University
Fine Hall, Washington Road
Princeton, New Jersey 08544
cf@math.princeton.edu

Partially supported by NSF Grants #DMS0245242 and #DMS0601025.