Algorithms for computing the optimal Lipschitz constant of interpolants with Lipschitz derivative

Matthew J. Hirn*
Yale University
Department of Mathematics
P.O. Box 208282
New Haven, Connecticut 06520-8283
matthew.hirn@yale.edu

July 15, 2013

Abstract

One classical measure of the quality of an interpolating function is its Lipschitz constant. In this paper we consider interpolants with additional smoothness requirements, in particular that their derivatives be Lipschitz. We show that such a measure of quality can be easily computed, giving two algorithms, one optimal in the dimension of the data, the other optimal in the number of points to be interpolated.

§1 Introduction

For an arbitrary function $g : \mathbb{R}^d \to \mathbb{R}^n$, recall that the *Lipschitz constant* of g is defined as:

$$\mathrm{Lip}(g) \triangleq \sup_{\substack{x,y \in \mathbb{R}^d \\ x \neq y}} \frac{|g(x) - g(y)|}{|x - y|},$$

where $|\cdot|$ is taken to be the standard Euclidean norm. Additionally, set $\nabla g : \mathbb{R}^d \to \mathbb{R}^d$ to be the *gradient* of g, where $\nabla g \triangleq (\frac{\partial g}{\partial x_1}, \ldots, \frac{\partial g}{\partial x_d})$.

Given a finite set $E \subset \mathbb{R}^d$ with $\#(E) = N$ and a function $f : E \to \mathbb{R}$, it is will known that the function f can be extended to a function $F : \mathbb{R}^d \to \mathbb{R}$

*www.math.yale.edu/∼mh644

such that $\mathrm{Lip}(F) = \mathrm{Lip}(f)$ (see the work of Whitney [10] and McShane [8] for the original result). Such a function F is a *minimal Lipschitz extension* of f, since the Lipschitz constant of F cannot be lowered while still interpolating the function f. Thus to compute $\mathrm{Lip}(F)$, we must compute $\mathrm{Lip}(f)$. This can clearly be accomplished in $O(N^2)$ operations. However, using the well separated pairs decomposition [1], one can compute a near approximation of $\mathrm{Lip}(f)$ in only $O(N \log N)$ operations.

In this paper, we address a related problem. We assume that along with the function values, we are also given information about the derivatives at each point in E. We wish to efficiently compute the minimal value of $\mathrm{Lip}(\nabla F)$, where $F : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function whose derivative is Lipschitz that additionally interpolates the given functional and derivative information.

Let $C^{1,1}(\mathbb{R}^d)$ denote the space of functions mapping $\mathbb{R}^d$ to $\mathbb{R}$ whose derivatives are Lipschitz:

$$C^{1,1}(\mathbb{R}^d) \triangleq \{g : \mathbb{R}^d \to \mathbb{R} : \mathrm{Lip}(\nabla g) < \infty\}.$$

Let $\mathcal{P}$ denote the set of first order polynomials (i.e., affine functions) mapping $\mathbb{R}^d$ to $\mathbb{R}$. For $F \in C^{1,1}(\mathbb{R}^d)$, let $J_x F \in \mathcal{P}$ denote the first order *jet* of F centered at x, i.e., $J_x F(z) \triangleq F(x) + \nabla F(x) \cdot (z - x)$. A *Whitney 1-field* $\mathcal{P}_E \triangleq \{P_x \in \mathcal{P} : x \in E\}$ is a set polynomials in $\mathcal{P}$ indexed by the set $E \subset \mathbb{R}^d$.

In this paper we address some of the computational aspects of the following problem:

<u>Jet Interpolation Problem</u>: Suppose we are given a finite set $E \subset \mathbb{R}^d$ and a 1-field $\mathcal{P}_E = \{P_x \in \mathcal{P} : x \in E\}$. Compute a function $F \in C^{1,1}(\mathbb{R}^d)$ such that

1. $J_x F = P_x$ for all $x \in E$.
2. $\mathrm{Lip}(\nabla F)$ is minimal.

There are two theoretical problems tied into the Jet Interpolation Problem. The first of these involves determining the optimal value of the semi-norm $\mathrm{Lip}(\nabla F)$. It is, by definition, given by:

$$\mathcal{L}(\mathcal{P}_E) \triangleq \inf\{\mathrm{Lip}(\nabla F) : F \in C^{1,1}(\mathbb{R}^d) \;\&\; J_x F = P_x \;\; \forall\, x \in E\}.$$

The second problem is to construct a function $F \in C^{1,1}(\mathbb{R}^d)$ that interpolates the 1-field $\mathcal{P}_E$ such that $\mathrm{Lip}(\nabla F) = \mathcal{L}(\mathcal{P}_E)$.

Remarkably, there are solutions to both of these problems. In [7], Le Gruyer gives a closed formula for $\mathcal{L}(\mathcal{P}_E)$, while in [9] Wells gives a construction for the interpolant F.

The two theoretical problems lead to two corresponding computational problems: (1) efficiently computing $\mathcal{L}(\mathcal{P}_E)$ and (2) efficiently computing the interpolant F. The theoretical results of Le Gruyer and Wells give a roadmap by which to accomplish these tasks.

The main result of this paper is to give an algorithm that efficiently computes a number M with the same order of magnitude of $\mathcal{L}(\mathcal{P}_E)$. In a follow up paper, we shall address the problem of efficiently computing an interpolant $F \in C^{1,1}(\mathbb{R}^d)$ for $\mathcal{P}_E$ such that $\mathrm{Lip}(\nabla F) = M$.

Two numbers X, Y that are dependent upon $E, \mathcal{P}_E$, and d are said to have the same *order of magnitude* if there exist universal constants c and C such that $cY \leq X \leq CY$.

By compute we mean develop an algorithm that can run on an idealized computer with standard von Neumann architecture, able to work with exact real numbers. We ignore roundoff, overflow, and underflow errors, and suppose that an exact real number can be stored at each memory address. Additionally, we suppose that it takes one machine operation to add, subtract, multiply, or divide two real numbers x and y, or to compare them (i.e., decide whether $x < y$, $x > y$, or $x = y$).

The *work* of an algorithm is the number of machine operations needed to carry it out, and the *storage* of an algorithm is the number of random access memory addresses required.

Throughout, we shall set $\#(E) = N$ to be the number of points in E.

Some related work on the computation of interpolants in $C^m(\mathbb{R}^d)$ is given in [5, 6, 2, 3]. In particular, this work is most closely related to [5, 6], but by working in $C^{1,1}(\mathbb{R}^d)$, and using the semi-norm $\mathrm{Lip}(\nabla F)$ as opposed to some C^m norm, we are able to achieve order of magnitude constants that do not depend on the dimension.

§2 Computing $\mathcal{L}(\mathcal{P}_E)$

In this section we present two algorithms for computing $\mathcal{L}(\mathcal{P}_E)$. One is an exact computation that is simply a corollary of the results found in [7]; it runs in $O(dN^2)$ time and requires $O(dN)$ storage. The second, which requires more effort to develop, computes the order of magnitude of $\mathcal{L}(\mathcal{P}_E)$ in $O(d^{d/2} N \log N)$ time and requires $O(d^{d/2} N)$ storage.

§2.1 Closed formula for $\mathcal{L}(\mathcal{P}_E)$ and an efficient algorithm in the dimension d

In [7], Le Gruyer gives a closed formula for $\mathcal{L}(\mathcal{P}_E)$, which is immensely useful for its computation. We summarize the results in this section.

For the 1-field $\mathcal{P}_E = \{P_x \in \mathcal{P} : x \in E\}$, define two functionals $A : E \times E \to [0, \infty]$ and $B : E \times E \to [0, \infty]$,

$$A(x, y) \triangleq \frac{|P_x(x) - P_y(x) + P_x(y) - P_y(y)|}{|x - y|^2}, \qquad B(x, y) \triangleq \frac{|\nabla P_x - \nabla P_y|}{|x - y|}.$$

Note that A was originally formulated differently in [7], we have simply rewritten it in a form more useful for our purposes. Additionally, recall that $\mathcal{P}$ is the set of first order polynomials, so for any $P \in \mathcal{P}$, ∇P is a constant vector in $\mathbb{R}^d$.

Using A and B, define Γ as:

$$\Gamma(\mathcal{P}_E) \triangleq \max_{\substack{x,y \in E \\ x \neq y}} \sqrt{A(x, y)^2 + B(x, y)^2} + A(x, y).$$

We then have the following theorem:

Theorem 1 (Le Gruyer, [7])**.** *For any finite $E \subset \mathbb{R}^d$ and any 1-field $\mathcal{P}_E$,*

$$\mathcal{L}(\mathcal{P}_E) = \Gamma(\mathcal{P}_E).$$

Thus the functional $\Gamma(\mathcal{P}_E)$ is the closed form of $\mathcal{L}(\mathcal{P}_E)$. If the number of data points N is reasonable, then it yields an obvious algorithm for computing $\mathcal{L}(\mathcal{P}_E)$ by simply evaluating $A(x, y)$ and $B(x, y)$ for all unique pairs $x, y \in E$ and computing $\Gamma(\mathcal{P}_E)$. We state this as a corollary.

Corollary 2. *There is an algorithm, whose inputs are the set E and the 1-field $\mathcal{P}_E$, that computes $\mathcal{L}(\mathcal{P}_E)$ exactly. It requires $O(dN^2)$ work and $O(dN)$ storage.*

The obvious benefit of this algorithm is that it computes $\mathcal{L}(\mathcal{P}_E)$ exactly. Additionally, the storage is asymptotically optimal both in d and in N, and the work is asymptotically optimal in d. On the other hand, if the number of points N is large, then the $O(dN^2)$ work is at best impractical, and at worst impossible. In order to handle this situation, we turn to the well separated pairs decomposition.

Remark 3. When we say that we input $\mathcal{P}_E$ into the computer, what we mean is that we input $P_x(x) \in \mathbb{R}$ and $\nabla P_x \in \mathbb{R}^d$ for each $x \in E$.

Remark 4. In fact Theorem 1 holds not only for $\mathbb{R}^d$, but for any Hilbert space with real valued inner product. Consequently, Corollary 2 can be applied to work in any Hilbert space (replacing the Euclidean norm with the Hilbert space norm), including infinite dimensional Hilbert spaces, so long as one has a method (or "black box") by which to compute inner products. This is often the case when the set $E \subset \mathbb{R}^d$ but one utilizes a kernel function $k : E \times E \to \mathbb{R}$ such that $k(x, y)$ is the inner product in a Hilbert space $\mathcal{H}$ after some implicit mapping $\varphi : E \to \mathcal{H}$.

§2.2 Well separated pairs decomposition

The well separated pairs decomposition was first introduced by Callahan and Kosaraju in [1]; we shall make use of a modified version that was described in detail in [5].

First, recall the standard definitions of the *diameter* of a set and the *distance* between two sets. Let $S, T \subset \mathbb{R}^d$,

$$\operatorname{diam}(S) \triangleq \sup_{\substack{x,y \in S \\ x \neq y}} |x - y|, \qquad \operatorname{dist}(S, T) \triangleq \inf_{\substack{x \in S \\ y \in T}} |x - y|.$$

Let $\varepsilon > 0$; two sets $S, T \subset \mathbb{R}^d$ are *ε-separated* if

$$\max\{\operatorname{diam}(S), \operatorname{diam}(T)\} < \varepsilon \operatorname{dist}(S, T).$$

We follow the construction detailed by Fefferman and Klartag in [5]. Let $\mathcal{T}$ be a collection of subsets of E. For any $\Lambda \subset \mathcal{T}$, set

$$\cup\Lambda \triangleq \bigcup_{S \in \Lambda} S = \{x : x \in S \text{ for some } S \in \Lambda\}.$$

Let $\mathcal{W}$ be a set of pairs (Λ_1, Λ_2) where $\Lambda_1, \Lambda_2 \subset \mathcal{T}$. For any $\varepsilon > 0$, the pair $(\mathcal{T}, \mathcal{W})$ is an *ε-well separated pairs decomposition* or *ε-WSPD* for short if the following properties hold:

1. $\bigcup_{(\Lambda_1, \Lambda_2) \in \mathcal{W}} \cup\Lambda_1 \times \cup\Lambda_2 = \{(x, y) \in E \times E : x \neq y\}$.
2. If $(\Lambda_1, \Lambda_2), (\Lambda_1', \Lambda_2') \in \mathcal{W}$ are distinct pairs, then $(\cup\Lambda_1 \times \cup\Lambda_2) \cap (\cup\Lambda_1' \times \cup\Lambda_2') = \emptyset$.
3. $\cup\Lambda_1$ and $\cup\Lambda_2$ are ε-separated for any $(\Lambda_1, \Lambda_2) \in \mathcal{W}$.
4. $\#(\mathcal{T}) < C(\varepsilon, d)N$ and $\#(\mathcal{W}) < C(\varepsilon, d)N$.

As shown in [5], there is a data structure representing $(\mathcal{T}, \mathcal{W})$ that satisfies the following additional properties as well:

5. The amount of storage to hold the data structure is $O((\sqrt{d}/\varepsilon)^d N)$.

6. The following tasks require at most $O((\sqrt{d}/\varepsilon)^d N \log N)$ work and $O((\sqrt{d}/\varepsilon)^d N)$ storage:
   (a) Go over all $S \in \mathcal{T}$, and for each S produce a list of elements in S.
   (b) Go over all $(\Lambda_1, \Lambda_2) \in W$, and for each (Λ_1, Λ_2) produce the elements (in $\mathcal{T}$) of Λ_1 and Λ_2.
   (c) Go over all $S \in \mathcal{T}$, and for each S produce the list of all $(\Lambda_1, \Lambda_2) \in W$ such that $S \in \Lambda_1$.
   (d) Go over all $x \in E$, and for each $x \in E$ produce a list of $S \in \mathcal{T}$ such that $x \in S$.

As a result of property 6, it follows that the following properties also hold:

7. For $C(\varepsilon, d) = O((\sqrt{d}/\varepsilon)^d)$,
   (a) $\sum_{(\Lambda_1,\Lambda_2)\in\mathcal{W}}(\#(\Lambda_1) + \#(\Lambda_2)) < C(\varepsilon, d)N \log N$.
   (b) $\sum_{S\in\mathcal{T}} \#(S) < C(\varepsilon, d)N \log N$.

Theorem 5 (Fefferman and Klartag, [5])**.** *There is an algorithm, whose inputs are the parameter $\varepsilon > 0$ and a subset $E \subset \mathbb{R}^d$ with $\#(E) = N$, that outputs a ε-WSPD $(\mathcal{T}, \mathcal{W})$ of E such that properties 1,. . . ,7 hold. The algorithm requires $O((\sqrt{d}/\varepsilon)^d N \log N)$ work and $O((\sqrt{d}/\varepsilon)^d)N)$ storage.*

Remark 6*.* The algorithm presented in [5] is built upon the well separated pairs decomposition algorithm developed by Callahan and Kosaraju in [1]. In fact, $\mathcal{T}$ is a completely balanced binary tree based off the inorder relation derived from the fair split tree presented in [1]. In particular, $\#(\mathcal{T}) < 2N$ and the height of the tree is bounded by $\lceil \log_2 N\rceil + 1$. The list $\mathcal{W}$ has a one-to-one correspondence with the well separated pair list presented in [1], hence $\#(\mathcal{W}) = O((\sqrt{d}/\varepsilon)^d N)$.

§2.3 Efficient computation of $\mathcal{L}(\mathcal{P}_E)$ in the number of points N

In this section we prove the following theorem:

Theorem 7. *There is an algorithm, whose inputs are the set E and the 1-field $\mathcal{P}_E$, that computes the order of magnitude of $\mathcal{L}(\mathcal{P}_E)$. It requires $O(d^{d/2}N \log N)$ work and $O(d^{d/2}N)$ storage.*

The plan for proving Theorem 7 is the following. First we view Le Gruyer's Γ functional from the perspective of the classical Whitney conditions. Once we formalize this concept, we can use the ε-WSPD of Fefferman and Klartag, since they built it to handle interpolants in $C^m(\mathbb{R}^n)$ satisfying Whitney conditions.

Concerning the first part, consider the original Whitney conditions for $C^{1,1}(\mathbb{R}^n)$:

(W_0) $|(P_x - P_y)(x)| \le M|x-y|^2$ for all $x, y \in E$.

(W_1) $|\frac{\partial}{\partial x_i}(P_x - P_y)(x)| \le M|x-y|$ for all $x, y \in E$, $i = 1, \ldots, d$.

Whitney's extension theorem states that if (W_0) and (W_1) hold, then there exists an $F \in C^{1,1}(\mathbb{R}^d)$ that interpolates $\mathcal{P}_E$ such that $\mathrm{Lip}(\nabla F) \le C(d)M$.

The main contribution of [7] is to refine (W_0) and (W_1) such that $C(d) = 1$; this is Γ. Indeed, the functional A corresponds to (W_0), the functional B corresponds to (W_1), and Γ pieces them together. Note there are some small, but significant differences. In particular, the functional A is essentially a symmetric version of (W_0); using one is equivalent to using the other, up to a factor of two. The functional B though, merges all of the partial derivative information into one condition, unlike (W_1). Thus they are equivalent only up to a factor of d, the dimension of the Euclidean space we are working in. For the algorithm in this section, we will use the functional B since it is both simpler and more useful than (W_1), but use (W_0) instead of A. Additionally, we will treat them separately instead of together like in Γ; Lemma 8 contains the details.

For the 1-field $\mathcal{P}_E$, define the functional $\widetilde{A} : E \times E \to [0, \infty]$ (which is essentially the same as (W_0)),

$$\widetilde{A}(x, y) \triangleq \frac{|P_x(x) - P_y(x)|}{|x-y|^2}.$$

Additionally, set

$$\widetilde{\Gamma}(\mathcal{P}_E) \triangleq \max_{\substack{x,y \in E \\ x \neq y}} \Big\{ \max\{\widetilde{A}(x,y), B(x,y)\} \Big\}.$$

The functional $\widetilde{\Gamma}(\mathcal{P}_E)$ is more easily approximated via the ε-WSPD than $\Gamma(\mathcal{P}_E)$. Furthermore, as the following Lemma shows, they have the same order of magnitude.

Lemma 8. *For any finite $E \subset \mathbb{R}^d$ and any 1-field $\mathcal{P}_E$,*

$$\widetilde{\Gamma}(\mathcal{P}_E) \le \Gamma(\mathcal{P}_E) \le 2(1+\sqrt{2})\widetilde{\Gamma}(\mathcal{P}_E).$$

Proof. To bridge the gap between $\Gamma(\mathcal{P}_E)$ and $\widetilde{\Gamma}(\mathcal{P}_E)$, we first consider

$$\Gamma'(\mathcal{P}_E) \triangleq \max_{\substack{x,y \in E \\ x \neq y}} \Big\{ \max\{A(x,y), B(x,y)\} \Big\}.$$

Clearly $\Gamma'(\mathcal{P}_E) \le \Gamma(\mathcal{P}_E)$. Furthermore,

$$\begin{aligned}\Gamma(\mathcal{P}_E) &= \max_{\substack{x,y \in E \\ x \neq y}} \sqrt{A(x,y)^2 + B(x,y)^2} + A(x,y) \\ &\le \sqrt{\Gamma'(\mathcal{P}_E)^2 + \Gamma'(\mathcal{P}_E)^2} + \Gamma'(\mathcal{P}_E) \\ &\le (1+\sqrt{2})\Gamma'(\mathcal{P}_E).\end{aligned}$$

Thus $\Gamma(\mathcal{P}_E)$ and $\Gamma'(\mathcal{P}_E)$ have the same order of magnitude, and in particular,

$$\Gamma'(\mathcal{P}_E) \leq \Gamma(\mathcal{P}_E) \leq (1+\sqrt{2})\Gamma'(\mathcal{P}_E). \tag{1}$$

Now let us consider $\Gamma'(\mathcal{P}_E)$ and $\widetilde{\Gamma}(\mathcal{P}_E)$ (which means considering $A(x,y)$ and $\widetilde{A}(x,y)$). First,

$$\begin{aligned} |P_x(x) - P_y(x) + P_x(y) - P_y(y)| &\leq |P_x(x) - P_y(x)| + |P_x(y) - P_y(y)| \\ &\leq 2\widetilde{\Gamma}(\mathcal{P}_E)|x-y|^2, \end{aligned}$$

and so, $\Gamma'(\mathcal{P}_E) \leq 2\widetilde{\Gamma}(\mathcal{P}_E)$. For a reverse inequality, we note,

$$|P_x(x) - P_y(x) + P_x(y) - P_y(y)| = |2(P_x(x) - P_y(x)) + (\nabla P_y - \nabla P_x)\cdot(x-y)|.$$

Thus,

$$\begin{aligned} 2|P_x(x) - P_y(x)| &\leq \Gamma'(\mathcal{P}_E)|x-y|^2 + |(\nabla P_y - \nabla P_x)\cdot(x-y)| \\ &\leq 2\Gamma'(\mathcal{P}_E)|x-y|^2, \end{aligned}$$

which yields $\widetilde{\Gamma}(\mathcal{P}_E) \leq \Gamma'(\mathcal{P}_E)$. Combining the two inequalities,

$$\widetilde{\Gamma}(\mathcal{P}_E) \leq \Gamma'(\mathcal{P}_E) \leq 2\widetilde{\Gamma}(\mathcal{P}_E). \tag{2}$$

Putting (1) and (2) together completes the proof. □

We will also need the following simple Lemmas.

Lemma 9. *Let $(\mathcal{T}, \mathcal{W})$ be a ε-WSPD, $(\Lambda_1, \Lambda_2) \in \mathcal{W}$, $x, x', x'' \in \cup\Lambda_1$, and $y, y' \in \cup\Lambda_2$. Then,*

$$\begin{aligned} &|x' - x''| \leq \varepsilon|x-y| \\ &|x' - y'| \leq (1+2\varepsilon)|x-y|. \end{aligned}$$

Proof. Use the definition of ε-separated. □

Lemma 10. *Suppose that $P \in \mathcal{P}$, $x \in \mathbb{R}^d$, $\delta > 0$, and $M > 0$ satisfy*

$$\begin{aligned} &|P(x)| \leq M\delta^2 \\ &|\nabla P| \leq M\delta. \end{aligned}$$

Then, for any $y \in \mathbb{R}^d$,

$$|P(y)| \leq M(\delta + |x-y|)^2.$$

Proof. Using Taylor's Theorem,

$$\begin{aligned}|P(y)| &= |P(x) + \nabla P(x) \cdot (y - x)| \\ &\leq |P(x)| + |\nabla P||x - y| \\ &\leq M\delta^2 + M\delta|x - y| \\ &\leq M(\delta + |x - y|)^2.\end{aligned}$$

□

Proof of Theorem 7. In order to simplify notation, let $\widetilde{\Gamma}(x, y)$ denote the quantity maximized in the definition of $\widetilde{\Gamma}(\mathcal{P}_E)$, i.e.,

$$\widetilde{\Gamma}(x, y) = \max\{\widetilde{A}(x, y), B(x, y)\}.$$

Additionally, set

$$\widetilde{A}(\mathcal{P}_E) \triangleq \max_{\substack{x,y\in E\\ x\neq y}} \widetilde{A}(x, y), \qquad B(\mathcal{P}_E) \triangleq \max_{\substack{x,y\in E\\ x\neq y}} B(x, y).$$

Our algorithm works as follows. For now, let $\varepsilon > 0$ be arbitrary and invoke the algorithm from Theorem 5. This gives us an ε-WSPD $(\mathcal{T}, \mathcal{W})$ in $O((\sqrt{d}/\varepsilon)^d N \log N)$ work and using $O(\sqrt{d}/\varepsilon)^d N)$ storage. For each $(\Lambda_1, \Lambda_2) \in \mathcal{W}$, pick at random a representative $(x_{\Lambda_1}, x_{\Lambda_2}) \in \cup\Lambda_1 \times \cup\Lambda_2$. Additionally, for each $S \in \mathcal{T}$, pick at random a representative $x_S \in S$.

Now compute the following:

$$\begin{aligned}\widetilde{\Gamma}_1 &\triangleq \max_{(\Lambda_1,\Lambda_2)\in\mathcal{W}} \widetilde{\Gamma}(x_{\Lambda_1}, x_{\Lambda_2}) \\ \widetilde{\Gamma}_2 &\triangleq \max_{(\Lambda_1,\Lambda_2)\in\mathcal{W}} \max_{i=1,2} \max_{S\in\Lambda_i} \widetilde{\Gamma}(x_{\Lambda_i}, x_S) \\ \widetilde{\Gamma}_3 &\triangleq \max_{S\in\mathcal{T}} \max_{x\in S} \widetilde{\Gamma}(x, x_S) \\ \widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}) &\triangleq \max\{\widetilde{\Gamma}_1, \widetilde{\Gamma}_2, \widetilde{\Gamma}_3\}.\end{aligned}$$

Define $\widetilde{A}(\mathcal{P}_E, \mathcal{T}, \mathcal{W})$ and $B(\mathcal{P}_E, \mathcal{T}, \mathcal{W})$ analogously. Using properties 6 and 7 from Section 2.2, we see that computing $\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W})$ requires $O((\sqrt{d}/\varepsilon)^d N \log N)$ work and $O((\sqrt{d}/\varepsilon)^d N)$ storage.

Now we show that $\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W})$ has the same order of magnitude as $\widetilde{\Gamma}(\mathcal{P}_E)$. Clearly, $\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}) \leq \widetilde{\Gamma}(\mathcal{P}_E)$. For the other inequality, we break $\widetilde{\Gamma}$ into its two parts, noting that $\widetilde{\Gamma}(\mathcal{P}_E) = \max\{\widetilde{A}(\mathcal{P}_E), B(\mathcal{P}_E)\}$ and $\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}) = \max\{\widetilde{A}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}), B(\mathcal{P}_E, \mathcal{T}, \mathcal{W})\}$. Thus we can work with $\widetilde{A}$ and B separately.

The functional B is simply the Lipschitz constant of the mapping $x \mapsto \nabla P_x$. It is known that

$$B(\mathcal{P}_E) \leq (1 + C\varepsilon)B(\mathcal{P}_E, \mathcal{T}, \mathcal{W}). \tag{3}$$

See for example Proposition 2 of [4]. Using the particular construction in this proof, we can take $C = 6$.

We now turn to $\widetilde{A}$. Let $x, y \in E$, $x \neq y$. By properties 1 and 2 of Section 2.2, there is a unique pair $(\Lambda_1, \Lambda_2) \in \mathcal{W}$ such that $(x, y) \in \cup\Lambda_1 \times \cup\Lambda_2$. Additionally, by the definition of $(\mathcal{T}, \mathcal{W})$, there exists a set $S \in \Lambda_1$ such that $x \in S$ and a set $T \in \Lambda_2$ such that $y \in T$.

Let $M = \widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W})$. We then have, using the triangle inequality, the definition of $\widetilde{\Gamma}_3$, and Lemma 9,

$$\begin{aligned} |P_x(x) - P_y(x)| &\leq |P_x(x) - P_{x_S}(x)| + |P_{x_S}(x) - P_y(x)| \\ &\leq \widetilde{\Gamma}_1 |x - x_S|^2 + |P_{x_S}(x) - P_y(x)| \\ &\leq \varepsilon M |x - y|^2 + |P_{x_S}(x) - P_y(x)|. \end{aligned} \tag{4}$$

Continuing with the second term of the right hand side of (4), we use the triangle inequality, Lemma 10, the definition of $\widetilde{\Gamma}_2$, and Lemma 9,

$$\begin{aligned} |P_{x_S}(x) - P_y(x)| &\leq |P_{x_S}(x) - P_{x_{\Lambda_1}}(x)| + |P_{x_{\Lambda_1}}(x) - P_y(x)| \\ &\leq \widetilde{\Gamma}_2 (|x_S - x_{\Lambda_1}| + |x - x_{\Lambda_1}|)^2 + |P_{x_{\Lambda_1}}(x) - P_y(x)| \\ &\leq 4\varepsilon^2 M |x - y|^2 + |P_{x_{\Lambda_1}}(x) - P_y(x)|. \end{aligned} \tag{5}$$

Continuing with the second term of the right hand side of (5), we use the triangle inequality, Lemma 10, the definition of $\widetilde{\Gamma}_3$, and Lemma 9,

$$\begin{aligned} |P_{x_{\Lambda_1}}(x) - P_y(x)| &\leq |P_y(x) - P_{x_T}(x)| + |P_{x_T}(x) - P_{x_{\Lambda_1}}(x)| \\ &\leq \widetilde{\Gamma}_3 (|y - x_T| + |x - y|)^2 + |P_{x_T}(x) - P_{x_{\Lambda_1}}(x)| \\ &\leq (1 + \varepsilon)^2 M |x - y|^2 + |P_{x_T}(x) - P_{x_{\Lambda_1}}(x)|. \end{aligned} \tag{6}$$

Continuing with the second term of the right hand side of (6), we use the triangle inequality, Lemma 10, the definitions of $\widetilde{\Gamma}_1$ and $\widetilde{\Gamma}_2$, as well as Lemma 9,

$$\begin{aligned} |P_{x_T}(x) - P_{x_{\Lambda_1(x)}}(x)| &\leq |P_{x_T}(x) - P_{x_{\Lambda_2}}(x)| + |P_{x_{\Lambda_2}}(x) - P_{x_{\Lambda_1}}(x)| \\ &\leq \widetilde{\Gamma}_2 (|x_T - x_{\Lambda_2}| + |x - x_{\Lambda_2}|)^2 + \widetilde{\Gamma}_1 (|x_{\Lambda_2} - x_{\Lambda_1}| + |x - x_{\Lambda_1}|)^2 \\ &\leq 2(1 + 3\varepsilon)^2 M |x - y|^2. \end{aligned} \tag{7}$$

Putting (4), (5), (6), (7) together, we get:

$$|P_x(x) - P_y(x)| \leq 3M|x - y|^2 + 23\varepsilon M |x - y|^2. \tag{8}$$

Taking $\varepsilon = 1/2$ gives the desired bounds on the work and storage, and in addition yields

$$\widetilde{\Gamma}(\mathcal{P}_E) \leq C \widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}).$$

The proof is completed by applying Lemma 8. $\square$

Remark 11. Examining (3) and (8), we see that $\widetilde{\Gamma}(\mathcal{P}_E)$ and $\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W})$ have the same order of magnitude with constants $c = 1$ and $C = C(\varepsilon) = 3 + 23\varepsilon$. Thus,

$$\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}) \leq \widetilde{\Gamma}(\mathcal{P}_E) \leq C(\varepsilon)\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}),$$

Recalling Lemma 8, we then have

$$\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}) \leq \Gamma(\mathcal{P}_E) \leq C'(\varepsilon)\widetilde{\Gamma}(\mathcal{P}_E, \mathcal{T}, \mathcal{W}),$$

where $C'(\varepsilon) = 2(1 + \sqrt{2})C(\varepsilon) = 2(1 + \sqrt{2})(3 + 23\varepsilon)$. Therefore, as $\varepsilon \to 0$, $C'(\varepsilon) \to 6(1 + \sqrt{2})$.

§3 Acknowledgements

The author would like to thank Charles Fefferman for introducing him to the problem and Hariharan Narayanan for numerous insightful conversations.

§4 References

[1] Paul B. Callahan and S. Rao Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the Association for Computing Machinery*, 42(1):67–90, 1995.

[2] Charles Fefferman. Interpolation by linear programming I. *Discrete and Continuous Dynamical Systems*, 30(2):477–492, June 2011.

[3] Charles Fefferman. Nearly optimal interpolation of data in $C^2(\mathbb{R}^2)$. part I. *Revista Matemática Iberoamericana*, 28(2):415–533, 2012.

[4] Charles Fefferman. Smooth interpolation of data by efficient algorithms. In Travis D. Andrews, Radu Balan, John J. Benedetto, Wojciech Czaja, and Kasso A. Okoudjou, editors, *Excursions in Harmonic Analysis*, volume I, pages 71–84. Springer, 2013.

[5] Charles Fefferman and Bo'az Klartag. Fitting a C^m-smooth function to data I. *Annals of Mathematics*, 169(1):315–346, 2009.

[6] Charles Fefferman and Bo'az Klartag. Fitting a C^m-smooth function to data II. *Revista Matemática Iberoamericana*, 25(1):49–273, 2009.

[7] Erwan Le Gruyer. Minimal Lipschitz extensions to differentiable functions defined on a Hilbert space. *Geometric and Functional Analysis*, 19:1101–1118, 2009.

[8] Edward James McShane. Extension of range of functions. *Bulletin of the American Mathematical Society*, 40(12):837–842, 1934.

[9] John C. Wells. Differentiable functions on Banach spaces with Lipschitz derivatives. *Journal of Differential Geometry*, 8:135–152, 1973.

[10] Hassler Whitney. Analytic extensions of differentiable functions defined in closed sets. *Transactions of the American Mathematical Society*, 36(1):63–89, 1934.