On Computable Normed Almost Linear Spaces

Hiten Dalmia
Department of Mathematics
Kirori Mal College, University of Delhi
Delhi - 110007
hitendalmia04@gmail.com

Keywords: Computable functional analysis, Effective descriptive set theory, Normed almost linear spaces
2010 MSC: 03F60, 46S30

Abstract

We study the notion of a computable normed almost linear space and examine some results of the theory of operators in normed almost linear spaces in the context of computability. Specifically, we establish the effectivity of the Inverse Mapping Theorem in the normed almost linear spaces. We use the Type-2 Theory of Effectivity (TTE) or the representations based approach to computability in this paper.

1 Introduction

The normed almost linear spaces (nals) are a framework for studying the theory of best simultaneous approximation in a normed linear spaces (nls) [3]. An almost linear space is a generalization of a linear space. A linear space is an abelian group with a group action, scalar multiplication, on a field. An almost linear space has all the axioms of a regular linear space except the existence of additive (right) inverses. We usually only define the almost linear space (als) on the $\mathbb{R}$ field. In [2], Brattka defines computability in Banach spaces and other normed spaces using the notion of separability, this is because we need a dense subset of the space to define an admissible representation on it, and working with separable spaces gives us a very natural way to define one.

Formally we say, an almost linear space is a set X with the mappings $s : X \times X \to X$ (addition, where $s(x, y), x, y \in X$ is denoted by $x + y$) and $m : \mathbb{R} \times X \to X$ (scalar multiplication, where $m(\lambda, x), \lambda \in \mathbb{R}, x \in X$ is denoted by λx) satisfying the following axioms for all $x, y, z \in X, \lambda, \mu \in \mathbb{R}$. (L1) $x + (y + z) = (x + y) + z$ (L2) $x + y = y + x$ (L3) $\exists 0 \in X$ such that $0 + x = x$ (L4) $1x = x$ (L5) $0x = 0$ (L6) $\lambda(x + y) = \lambda x + \lambda y$ (L7) $(\lambda\mu)x = \lambda(\mu x)$ (L8) $(\lambda + \mu)x = \lambda x + \mu x$ for $\lambda \geq 0, \mu \geq 0$

We define a norm on an als X, $|| \cdot || : X \to \mathbb{R}$. This is the same norm we are familiar with, when an almost linear space is equipped with it we call it a normed almost linear space. A norm satisfies the following axioms for all $x, y \in X, w \in W_X$ and $\lambda \in \mathbb{R}$.

(N1) $||x|| = 0$ iff $x = 0$

(N2) $||\lambda x|| = |\lambda| ||x||$

(N3) $||x + y|| \leq ||x|| + ||y||$

(N4) $||x|| \leq ||x + w||$

We will define two important subsets for a nals.

$$V_X = \{x \in X \mid x - x = 0\} \tag{1}$$

$$W_X = \{x \in X \mid x = -x\} = \{x - x \mid x \in X\} \tag{2}$$

V_X is a linear subspace of X, and W_X is an almost linear subspace of X. Of course, we have $W_X \cap V_X = \{0\}$. An als X is a linear space iff $W_X = \{0\}$.

We will now very briefly summarise the basic facts about the representation based approach to computability in the reals. This approach is based on inducing computability on objects through their representations, which are ways of representing mathematical objects like real numbers or real-valued functions using infinite strings. However, we only work with a certain class of representations, which allow for the property that finite approximations of the output of a computation are determined by finite portions of the input. In classical computability we work with the set of all finite words Σ^*, here Σ is the alphabet, in computable analysis we extend this idea to work with the set of all infinite words, Σ^ω. Formally, a representation is a surjective function $\delta :\subseteq \Sigma^\omega \to M$, where M is a set of objects such as $\mathbb{R}$ or $\mathcal{C}[0, 1]$. $\langle \cdot, \cdot \rangle : \Sigma^a \times \Sigma^b \to \Sigma^c$ is a tupling function where a, b, c are $*$ or ω depending on the context and c depends on a, b. For any object $q \in M$, if $\sigma : \Sigma^* \to M$ and $w \in \Sigma^*$ such that $\sigma(w) = q$, we write $\overline{w} := \sigma(w)$. We work only with admissible representations for the convenient fact that continuity induced by them on functions on the represented sets is equivalent to the topological continuity of these functions, for more details refer [7].

2 Basic definitions

We will now look at a few definitions which are necessary for our construction of a computable normed almost linear space. A represented space is a 2-tuple (X, δ) where X is a set and $\delta :\subseteq \Sigma^\omega \to X$ is a representation of X. We call a *word function*, $F :\subseteq \Sigma^\omega \to \Sigma^\omega$ computable if there exists a Type-2 Turing machine (TM) [7] which on reading input p continually writes $F(p)$ on its output tape and for input $p \notin \text{dom}(F)$ the TM *loops* or *diverges*, i.e., computes infinitely. We state a few more definitions, from [2].

Definition 1 (Computable Function)**.** Let (X, δ) and (Y, δ') be represented spaces. Then a function $f :\subseteq X \to Y$ is called (δ, δ')–computable if there exists a computable word function $F : \Sigma^\omega \to \Sigma^\omega$ such that

$$f \circ \delta(p) = \delta' \circ F(p)$$

for all $p \in \text{dom}(f \circ \delta)$.

A computable metric space is simply one which has a computable metric and in which an admissible representation can be defined, that is, the existence of a dense subset and a computable notation for the subset. Notations are naming systems which have their domain in Σ^*, in contrast to representations which have their domain in Σ^ω.

Definition 2 (Computable Metric Space)**.** A tuple (X, d, α) is called a computable metric space, if

1. $d : X \times X \to \mathbb{R}$ is a metric on X,
2. $\alpha : \mathbb{N} \to X$ is a sequence dense in X,
3. $d \circ \alpha \times \alpha : \mathbb{N}^2 \to \mathbb{R}$ is a computable double sequence in $\mathbb{R}$.

Given a metric space we define its *cauchy representation* as $\delta_X :\subseteq \Sigma^\omega \to X$ such that

$$\delta_X(01^{n_1}01^{n_2}01^{n_3}\ldots) := \lim_{i\to\infty} \alpha(n_i)$$

such that the limit exists and $d(\alpha(n_i), \alpha(n_j)) < 2^{-i}, \forall j \geq i$. The defined cauchy representation is admissible, for this, the existence of a dense subset is essential. This is something we will also need in our construction of a computable metric almost linear space.

2.1 Computable Normed Almost Linear Space

In [2], Brattka defines a computable linear space by any linear space which has computable addition and scalar multiplication and where 0 is a computable element of the linear space. Inspired by this, we give the following definition.

Definition 3 (Computable Almost Linear Space)**.** A represented space (X, δ) is called a computable almost linear space (over $\mathbb{R}$), if $(X, +, \cdot, 0)$ is an als such that the following conditions hold:

1. $+ : X \times X \to X, (x, y) \mapsto x + y$ is computable,
2. $\cdot : \mathbb{R} \times X \to X, (a, x) \mapsto a \cdot x$ is computable,
3. $0 \in X$ is a computable element.

We return to the theory of normed almost linear spaces and define some objects and introduce some theorems necessary for our main result.

Definition 4 (Semi-metric)**.** A function $\rho : X \times X \to \mathbb{R}$ is said to be a semi-metric if it satisfies the following conditions for all $x, y, z \in X$

1. $\rho(x, y) = \rho(y, x) \geq 0$
2. $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$
3. $x = y \implies \rho(x, y) = 0$

A semi-metric satisfies all the conditions that a metric does except $\rho(x, y) = 0 \implies x = y$. As such, it generates a topology on the set it is defined.

A *cone* in an als X is a set $C \subseteq X$ such that $\lambda x \in C, \forall x \in X, \lambda \geq 0$. We say V is a *convex* subset if $\lambda x + (1-\lambda)y \in V, \forall \lambda \in [0, 1]$ and $x, y \in V$. Let X, Y be almost linear spaces, $T : X \to Y$ is called a *linear operator* if $T(\alpha x + \beta y) = \alpha T(x) + \beta T(y)$ where $\alpha, \beta \in \mathbb{R}$ and $x, y \in X$. We now mention an important theorem of normed almost linear spaces ([4], Theorem 3.2).

Theorem 1. For any nals $(X, \|\cdot\|)$ there exist a normed linear space $(E_X, \|\cdot\|_{E_X})$ and a mapping $\omega_X : X \to E_X$ with the following properties:

(i) The set $X_1 = \omega_X(X)$ is a convex cone of E_X such that $E_X = X_1 - X_1$, and X_1 can be organized as an als where the addition and the multiplication by non-negative reals are the same as in E_X.

(ii) For each $z \in E_X$ we have:

$$\|z\|_{E_X} = \inf\{\|x_1\| + \|x_2\| : x_1, x_2 \in X, z = \omega_X(x_1) - \omega_X(x_2)\} \tag{3}$$

and the als X_1 together with this norm is a nals.

(iii) The mapping ω_X from X onto the nals X_1 is a linear operator and $\|\omega_X(x)\|_{E_X} = \|x\|$ for each $x \in X$.

We will skip using the subscript X for E_X, ω_X for the rest of the text. We have some important consequences of this theorem, $\omega(W_X) = W_{X_1}$ and $\omega(V_X) = V_{X_1}$. And, if $\omega : X \to X_1$ is one-one then $\omega^{-1} : X_1 \to X$ is also linear. Using this theorem we can also define a semi-metric for all nals ([4], Corollary 3.3).

Corollary 1. For any nals $(X, \|\cdot\|)$ the function

$$\rho(x, y) = \|\omega(x) - \omega(y)\| \quad (x, y \in X)$$

is a semi-metric on X and we have:

$$\rho(-1x, -1y) = \rho(x, y) \quad (x, y \in X) \tag{4}$$

This semi-metric ρ is a metric when ρ is one-one. We can now talk about topological properties like continuity, openness, closeness, etc. since the semi-metric for a nals generates a topology. Note, if A is a closed subset of a nals X, then $\omega(A)$ is a closed subset of X_1 [5].

To define an admissible representation on a topological space, we need a second countable T_0 space. The semi-metric on a nals given by corollary 1 generates a topology that is not T_0 in general, see that $\rho(x, y) = 0 \nRightarrow x = y$. This means that we can not define a computable semi-metric space in the lines of a computable metric space, since a semi-metric space is not guaranteed to be a T_0 space and can not necessarily have an admissible representation defined on it.

Let X, Y be two almost linear spaces and C a convex cone of Y [5].

Definition 5 (Almost Linear Operator)**.** A mapping $T : X \to Y$ is called an almost linear operator with respect to C if the following three conditions hold:

5.1 $T(x_1 + x_2) = T(x_1) + T(x_2) \quad (x_1, x_2 \in X)$

5.2 $T(\lambda x) = \lambda T(x) \quad (x \in X, \lambda \geq 0)$

5.3 $T(W_X) \subseteq C$

We denote by $\mathcal{L}(X, (Y, C))$ the set of all $T : X \to Y$ satisfying (5.1)–(5.3). We organize $\mathcal{L}(X, (Y, C))$ as an als in the following way: for $T_1, T_2 \in \mathcal{L}(X, (Y, C))$ and $\lambda \in \mathbb{R}$ we define $T_1 + T_2 \in \mathcal{L}(X, (Y, C))$ and $\lambda \circ T \in \mathcal{L}(X, (Y, C))$ by

$$\begin{aligned} (T_1 + T_2)(x) &= T_1(x) + T_2(x) \qquad (x \in X) \\ (\lambda T)(x) &= T(\lambda x) \qquad (x \in X) \end{aligned}$$

The additive identity $0 \in \mathcal{L}(X, (Y, C))$ is the operator which is identically zero. The space $\mathcal{L}(X, (Y, C))$ forms a nals when the convex set C has a special property (P).

Definition 6. The convex cone C has **property (P)** in X if the relations $x, y \in X$, $x + y \in C$ and $c \in C$ imply that

$$\max \{\|x\|, \|y\|\} \leq \max \{\|x + c\|, \|y + c\|\}$$

For $T \in \mathcal{L}(X, (Y, C))$, we define

$$\|T\| = \sup\{\|T(x)\| \mid \|x\| \leq 1\}$$

and $L(X, (Y, C)) = \{T \in \mathcal{L}(X, (Y, C)) \mid \|T\| < \infty\}$. T is *bounded* if $T \in L(X, (Y, C))$, i.e., $\|T\| < \infty$. Also note

$$\|T\| = \sup\{\|T(x)\| \mid \|x\| \leq 1\} = \sup\{\|T(\frac{x}{\|x\|})\| \mid x \in X\} = \sup\{\frac{1}{\|x\|}\|T(x)\| \mid x \in X\}.$$

and hence $\forall x \in X$

$$\|T(x)\| \leq \|T\|\|x\|.$$

We can now finally define a computable normed almost linear space. Again, we look for a definition that is analogous to the definition of a computable normed linear space [2], that suits our purposes.

Definition 7 (Computable Normed Almost Linear Space)**.** A tuple $(X, \|\cdot\|, e)$ is called a computable normed almost linear space, if

1. $\|\cdot\| : X \to \mathbb{R}$ is a norm on X,
2. $e : \mathbb{N} \to X$ is a complete (fundamental) sequence, i.e. its linear span is dense in X,
3. (X, d, α_e) with $\rho(x, y) := \|\omega(x) - \omega(y)\|$ where ω is one-one and $\alpha_e\langle k, \langle n_0, \ldots, n_k\rangle\rangle := \sum_{i=0}^{k} \alpha_{\mathbb{R}}(n_i)e_i$, is a computable metric space with Cauchy representation δ_X,
4. (X, δ_X) is a computable almost linear space over $\mathbb{R}$.

2.2 Hyperspace of Open sets

We will need to work with open sets in our proof of the computable version of the Banach Inverse Mapping Theorem in nals. Hence, we define a representation for the hyperspace of open sets of a set X [7, 2].

Definition 8 (Hyperspace of open subsets)**.** Let (X, d, α) be a computable metric space. For the hyperspace $\mathcal{O}(X) := \{U \subseteq X : U \text{ open}\}$ of open subsets, we give the representation $\delta_{\mathcal{O}(X)}$, defined by

$$\delta_{\mathcal{O}(X)}(01^{\langle n_0, k_0\rangle}01^{\langle n_1, k_1\rangle}01^{\langle n_2, k_2\rangle}\ldots) := \bigcup_{i=0}^{\infty} B(\alpha(n_i), \overline{k}_i).$$

Every representation of the hyperspace of open sets will induce a representation on the hyperspace of closed sets, however we will now not describe any representation for the hyperspace of closed sets, since our proof only uses open sets.

3 Main Results

We will first state the Inverse Mapping Theorem (Bounded Inverse Theorem) that we are familiar with from functional analysis, for more details refer any standard textbook on functional analysis like [6].

Theorem 2 (Inverse Mapping Theorem)**.** If a bounded linear operator T from a Banach space X onto a Banach space Y is bijective, its inverse T^{-1} is also a bounded linear operator.

We have the following theorem as the generalized version of the Inverse Mapping Theorem ([5], Theorem 6.3).

Theorem 3 (Generalized Inverse Mapping Theorem)**.** Let X, Y be two complete normed almost linear spaces such that both ω_X and ω_Y are one-one. If $T \in L(X, (Y, W_Y))$ is one-one and onto Y and $T(W_X) = W_Y$, then the inverse operator $T^{-1} \in L(Y, (X, W_X))$.

Similar to what Brattka does in [2, 1], an effective version of the Generalized Inverse Mapping Theorem can ask two questions,

1. does T computable $\implies T^{-1}$ computable?
2. is the mapping $T \mapsto T^{-1}$ computable?

The answer to the second question is negative in the context of the Inverse Mapping Theorem [2, 1], the same stands true for the Generalized Inverse Mapping Theorem. Take X, Y Banach spaces where ω_X, ω_Y are identity maps and T is a linear operator. We will have $W_X = W_Y = \{0\}$ and theorem 3 will reduce to the Inverse Mapping Theorem. Hence the second question is not true for the Generalized Inverse Mapping Theorem either, since it reduces to the Inverse Mapping Theorem in the conditions described.

Our proof for the following theorem is based on the proof for the Computable Inverse Mapping Theorem ([1], Corollary 5.3). We first state a Lemma required in the proof from ([1], Proposition 3.5).

Lemma 1. Let X be a computable metric space. The map $Z : \mathcal{C}(X) \to \mathcal{O}(X), f \mapsto X \setminus f^{-1}\{0\}$ is computable and admits a computable right-inverse $\mathcal{O}(X) \rightrightarrows \mathcal{C}(X)$.

We need another lemma from ([5], Lemma 5.1). Let $X_1 = \omega_X(X), Y_1 = \omega_Y(Y)$ and $C_1 = \omega_Y(C)$.

Lemma 2. For each $T \in L(X, (Y, C))$ there exists a unique $\tilde{T} \in L(X_1, (Y_1, C_1))$ such that $\tilde{T}\omega_X = \omega_Y T$ and $\|\tilde{T}\| = \|T\|$.

Theorem 4 (Effective Generalized Inverse Mapping Theorem)**.** Let X, Y be two computable complete normed almost linear spaces such that both ω_X and ω_Y are one-one. If $T \in L(X, (Y, W_Y))$ is computable, one-one and onto Y and $T(W_X) = W_Y$, then the inverse operator $T^{-1} \in L(Y, (X, W_X))$ and is computable.

Proof. Note that since ω_X, ω_Y are one-one, X, Y also form metric spaces, with the metric (given by corollary 1) for both X, Y being represented by ρ, however, it will be clear from the context which metric we wish to denote. Let $\alpha : \mathbb{N} \to X$ and $\beta : \mathbb{N} \to Y$ be the complete sequences, $\delta_X : \Sigma^\omega \to X$ and $\delta_Y : \Sigma^\omega \to Y$ be the Cauchy representations of X and Y respectively. Let us define the following function

$$O(T^{-1}) : O(Y) \to O(X), \quad V \mapsto T^{-1}(V)$$

Now since, T is a computable function, it is also continuous [7] and hence, $O(T^{-1})$ is well-defined. Now take some continuous $f : Y \to \mathbb{R}$ such that $V = Y \setminus f^{-1}(\{0\})$. Now,

$$T^{-1}(V) = T^{-1}(Y \setminus f^{-1}(\{0\})) = X \setminus T^{-1}(f^{-1}(\{0\})) = X \setminus (f \circ T)^{-1}(\{0\})$$

Since T is continuous, $T \in \mathcal{C}(X, Y)$ and the composition $\circ : \mathcal{C}(Y, \mathbb{R}) \times \mathcal{C}(X, Y) \to \mathcal{C}(Y, \mathbb{R})$, $(f, T) \mapsto f \circ T$ is computable. From lemma 1, $O(T^{-1})$ is computable. Let $O(S) := O(T^{-1})$, $O(S)$ is well-defined and computable, whence, S is open. Let M be the Turing machine which computes a realization of $O(S)$. Since S is bounded (from theorem 3), there exists a rational bound $s > 0$ such that $\|Sy\| \leq s\|y\|$ for all $y \in Y$ and some $j \in \mathbb{N}$ such that $2^j > s$. We construct a Turing machine M' which computes a (δ_Y, δ_X)–realization of S. Given some input $p = 01^{n_0}01^{n_1}01^{n_2}0...$ with $y := \delta_Y(p)$, machine M' works in steps $i = 0, 1, 2, ...$ as follows: in step i machine M' starts machine M with input $q = 01^{\langle n_{i+j+2}, k_i \rangle}01^{\langle n_{i+j+2}, k_i \rangle}01^{\langle n_{i+j+2}, k_i \rangle}0...$ where $\overline{k}_i := 2^{-i-j-2}$ and simulates M until it writes the first word $01^{\langle n,k \rangle}0$. Then M' writes 01^n on its output tape and continues with the next step $i + 1$.

Since $\delta_{\mathcal{O}(Y)}(q) = B(\beta(n_{i+j+2}), \overline{k}_i)$, M produces an output r with

$$\delta_{\mathcal{O}(X)}(r) = \mathcal{O}(S)(\delta_{\mathcal{O}(Y)}(q)) = S(B(\beta(n_{i+j+2}), \overline{k}_i)).$$

Thus, for any subword 01^n which is written by M' in step i on its output tape, we obtain $\alpha(n) \in S(B(\beta(n_{i+j+2}), \overline{k}_i))$. From lemma 2, theorem 1 and $\rho(y, \beta(n_{i+j+2})) \leq \overline{k}_i$, it follows

$$\begin{aligned}
\rho(S\beta(n_{i+j+2}), Sy) &= \|\omega_X(S\beta(n_{i+j+2})) - \omega_X(Sy)\| \\
&= \|\tilde{S}(\omega_Y(\beta(n_{i+j+2}))) - \tilde{S}(\omega_Y(y))\| \\
&\leq \|\tilde{S}\| \|\omega_Y(\beta(n_{i+j+2})) - \omega_Y(y)\| \\
&\leq s\rho(\beta(n_{i+j+2}), y) = s\overline{k_i}.
\end{aligned}$$

Then

$$\rho(\alpha(n), Sy) \leq \rho(\alpha(n), S\beta(n_{i+j+2})) + \rho(S\beta(n_{i+j+2}), Sy) \leq 2s\overline{k}_i < 2^{-i-1}$$

and hence $\delta_X(t) = Sy$ holds for the infinite output t of M'. Hence, $S = T^{-1}$ is computable. □

We also hold that the result that any two comparable computable complete norms are computably equivalent from ([1], Theorem 5.10) is also true in nals. The proof for the following theorem is very similar to the original result and is omitted.

Theorem 5. Let $(X, \|\cdot\|, e)$ and $(X, \|\cdot\|', e')$ be computable nals and let δ, δ' be the corresponding Cauchy representations of X. If $\delta \leq \delta'$ then $\delta \equiv \delta'$.

References

[1] Vasco Brattka. "A computable version of Banach's Inverse Mapping Theorem". en. In: *Ann. Pure Appl. Logic* 157.2-3 (Feb. 2009), pp. 85–96.

[2] Vasco Brattka. *Computability of Banach space principles*. en. Vol. 286. Informatik-Berichte. Hagen: FernUniversität in Hagen, 2001. URL: `https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00000784`.

[3] G Godini. "A framework for best simultaneous approximation: Normed almost linear spaces". en. In: *J. Approx. Theory* 43.4 (Apr. 1985), pp. 338–358.

[4] G Godini. “On normed almost linear spaces”. en. In: *Math. Ann.* 279.3 (Jan. 1988), pp. 449–455.

[5] G. Godini. *Operators in normed almost linear spaces.* Vol. 14. Rendiconti del Circolo Matematico di Palermo, II. Ser. Suppl. Proceedings of the 14th Winter School, Srní, Czechoslovakia, 1986. Circolo Matematico di Palermo, 1987, pp. 309–328. URL: https://eudml.org/doc/221445.

[6] Erwin Kreyszig. *Introductory functional analysis with applications.* John Wiley & Sons, 1991.

[7] Klaus Weihrauch. *Computable Analysis.* en. 2000th ed. Texts in Theoretical Computer Science. An EATCS Series. Berlin, Germany: Springer, Sept. 2000.