Part II

The Computability Theory of Banach Spaces

Chapter 2
Computability Structures on a Banach Space

Introduction

In this chapter we introduce the concept, "computability structure on a Banach space". This concept will play a fundamental role throughout the remainder of the book.

The notion of a computability structure arose in response to the following question. Which processes in analysis and physics preserve computability and which do not? We first make this question precise, and then we answer it.

How will the term "process" be understood? Among the processes of physics we expect to include are those associated with the wave equation, the heat equation, Laplace's equation and many others. Among the processes of analysis we expect to consider are Fourier series, Fourier transform and others. A moment's thought will convince the reader that the solution operator associated with each of these processes is a linear operator which maps functions into functions. Hence we will be concerned with linear operators on Banach spaces of functions. They will be our "processes".

How will the term "computable" be understood? Of course, since we will be dealing with Fourier series and transforms, solutions of the wave and heat equations etc., the appropriate notion is "computable function of a real/complex variable". The classical definition of this was spelled out in Chapter 0. In order to cover the variety of processes mentioned above, we will have to extend this definition to computability on an arbitrary Banach space. The extended definition plays a key role in the statement, proof and applications of the three principal results of this book. One of them, the *First Main Theorem*, provides corollaries which answer the question proposed above: which "processes" do and which do not preserve computability. The other two are the *Second Main Theorem* and the *Eigenvector Theorem*. These two results, together with their associated theorems in Chapter 4, determine precisely the "computability relationships" between an operator and its eigenvalues/spectrum/eigenvectors in a general setting. As is well known, linear operators on Hilbert space—more particularly the unbounded ones—are of considerable importance in physical theory.

The concept of a "computability structure on a Banach space" will be given axiomatically. This is a natural approach when one wants to encompass a variety of applications.

What concept should be axiomatized? Note that it is not sufficient to axiomatize the notion of a "computable point". The reason is obvious: topological notions are required in the solution of problems in analysis. Since a Banach space is a metric space, the topology can be given by convergent sequences. This suggests that an appropriate concept for axiomatization is "computable sequence of points". (A computable point x is a point such that the sequence $x, x, x \ldots$ is computable.) Of course, the topology can also be given by other, equivalent methods—e.g. open sets. However the sequence approach is very natural in the study of computability. Indeed one of the fundamental intuitions of recursive function theory is the notion of a "computable sequence of integers"—a sequence $a(0), a(1), \ldots$ obtained by a Turing Machine computation of the recursive function a. We will see that the concept, "computability structure on a Banach space", which is in fact an axiomatization of "computable sequence of points $x_0, x_1, x_2, \ldots$ of a Banach space", will provide a useful and flexible tool for solving problems.

Once the decision is made to axiomatize "computable sequence of points", the specific axioms follow naturally. Since a Banach space is linear, we need an axiom for linearity (Axiom 1, below). Since a Banach space is complete, we require an axiom for completeness (Axiom 2). Since a Banach space is normed, we need an axiom for the norm (Axiom 3). In fact, these are the only axioms which will be postulated. A detailed presentation of the axioms will be given in the next section.

When viewed in this light, the axioms appear to be minimal: they are just sufficient to relate the concept of computability to the basic concepts of Banach space theory. We will see later that under very general conditions which are satisfied in practical situations, the axioms are not merely minimal but also maximal. Thus it is not possible to postulate axioms which, when added to the axioms of a computability structure, give a more intuitive concept of computability.

The following definitions are both natural and useful. They are concerned with an effectivization of the notion of a separable Banach space. A computability structure $\mathscr{S}$ on a Banach space X is *effectively separable* if it contains a sequence $\{e_n\}$ whose linear span is dense in X. (Recall that the linear span of a set V consists of all finite linear combinations with real/complex coefficients of the elements of V.) Such a sequence $\{e_n\}$ is called an *effective generating set* for $\mathscr{S}$.

Although it is natural for applications to define a computability structure axiomatically, an axiomatic approach may be surprising to the expert in recursion theory. There is something "genetic" about computability. Witness, for example, the definition of computable function from $\mathbb{N}$ to $\mathbb{N}$. As is well-known, early research in this area consisted in attempting to capture the intuitive concept of computability by an appropriate definition. Definitions of this concept were presented by Turing, Post, Herbrand/Gödel, Markov, etc. The fact that the insights of Post, Turing, etc. led to the same class of functions was important in concluding that the "right" definition of "computable function from $\mathbb{N}$ to $\mathbb{N}$" had been found. Recursion theory on $\mathbb{N}$ is the study of the consequences of this definition.

By contrast, the axiomatic approach to a computability structure allows for a multitude of computability structures on a Banach space. (As mentioned earlier this is essential for applications.) Not all of these structures can be "right". Yet, surprisingly enough, the axiomatic approach appears to capture the concept of computability rather well. There are several reasons for this, which we now consider.

Associated with each of the usual Banach spaces—e.g. $C[0, 1]$, $L^p[0, 1]$, l^p $(1 \leqslant p < \infty)$ etc.—there is an intrinsic notion of computability which satisfies the axioms of a "computability structure". This will be discussed in greater detail later in the chapter (Sections 2, 3, 4). At this point we merely remark that the intrinsic computability structure is unique. For the space $C[0, 1]$, we will see that it is given by the "computable sequence of computable functions" defined in Chapter 0. We note further that, for each of the spaces mentioned above, intrinsic computability can be presented in many ways. The different ways originate, in part, from the different branches of analysis. This happens as follows. The specific branch of analysis focuses upon a particular, well-understood sequence of functions which is dense in the Banach space. The computability structure is obtained by taking the effective closure of this sequence. For example, if one is concerned with Fourier series, it is natural to choose the trigonometric polynomials with rational (complex rational) coefficients as the sequence and take its effective closure. If, however, one is concerned with problems in measure and integration on L^p-space $(1 \leqslant p < \infty)$, it is more natural to consider a sequence composed of certain computable step functions —i.e. those step functions with rational jump points and values. As we will see later, each of these sequences is an example of an "effective generating set". (This concept was discussed above.) We will show that the effective closure of those effective generating sets which arise naturally from basic constructs in analysis all lead to the same computability structure—intrinsic computability. For this reason, intrinsic computability plays a fundamental role in a wide variety of applications. This fact is of considerable importance, not only for applications but also for understanding the nature of computability on a Banach space.

The genetic quality of the definition of a computability structure can also be viewed in the more general context of a separable Banach space. Recall that the axioms of a computability structure on any Banach space—whether separable or not—appear to be minimal. They are just sufficient to impact the concept of computability with the basic notions of Banach space theory—linearity, limit and norm. For an effectively separable Banach space something stronger holds. The axioms are also maximal. More specifically it is not possible to formulate axioms which, when added to the axioms of a computability structure, yield a more intuitive characterization of computability. This is a consequence of the following theorem which will appear in Section 5.

Stability Lemma. *Let $\{e_n\}$ be a sequence whose linear span is dense in the Banach space X. Let $\mathscr{S}'$ and $\mathscr{S}''$ be computability structures on X such that $\{e_n\} \in \mathscr{S}'$ and $\{e_n\} \in \mathscr{S}''$. Then $\mathscr{S}' = \mathscr{S}''$.*

Thus we see that if a computability structure $\mathscr{S}$ contains an effective generating set, then it can have no proper extension. This rigidity in the structure of $\mathscr{S}$ implies the aforementioned redundancy of any further axioms. (Note also that $\mathscr{S}$ cannot have any proper restriction containing the effective generating set.)

The fact that for an effectively separable Banach space, the axioms of a computability structure are both maximal and minimal seems to indicate that, at least for those spaces, the concept of a computability structure does capture the intuitive notion which we need—computability on a Banach space.

Although the Stability Lemma is very general, it does require that the two structures $\mathscr{S}'$ and $\mathscr{S}''$ share a common effective generating set $\{e_n\}$. This condition is satisfied in most practical cases. However, for the purpose of constructing counterexamples, we may sometimes consider computability structures which do not contain the usual effective generating sets. Such artificial computability structures do exist, and we call them "ad hoc structures".

It should be pointed out that ad hoc computability structures play an important role in our research. They are *not* useless by-products of the axiomatic approach, as we will see. One way in which they are used is as follows. There are some natural techniques in analysis—particularly for obtaining counterexamples—which are noncomputable. They lead readily to ad hoc computability structures. In some instances it is possible to translate the ad hoc computability structure into the intrinsic one. In the process of translation, the analytic intuition which led to the construction is so thoroughly disguised that, for all practical purposes, it is lost. The counterexamples would be difficult to discover without first passing through an ad hoc structure. The proof of the Eigenvector Theorem (Theorem 6, Chapter 4) provides an example of an ad hoc structure which is used in this way.

Before concluding this section, the following should be remarked. We have NOT defined a computable Banach space. We began with a preexisting Banach space and defined a computability structure on it. Furthermore, when reasoning about Banach space theory, we employ classical logic, as physicists and analysts do. In particular we make free use of nonconstructive methods of proof to characterize the set of computable sequences within the larger class of all sequences. Thus our work is firmly within the tradition of recursive function theory.

1. *The Axioms for a Computability Structure*

We begin with a technical note. Virtually everything we do applies mutatis mutandis to either real or complex Banach spaces. The same holds for explicit function spaces such as $C[a, b]$, $L^p[a, b]$, etc. We shall take this as being understood, and shall rarely mention it in what follows.

Let X be a Banach space. The undefined term, which the following axioms will govern, is "computable sequence $\{x_n\}$", $x_n \in X$. We denote the set of all computable sequences by $\mathscr{S}$. Thus a Banach space with a computability structure is really a pair $\langle X, \mathscr{S}\rangle$. By a conventional "abuse of notation", we shall sometimes refer merely to X (or $\mathscr{S}$), the other member of the pair being understood.

A double sequence $\{x_{nm}\}$ is called "computable" if it is mapped onto a computable sequence by one of the standard recursive pairing functions from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. Similarly for triple or k-fold sequences.

An element $x \in X$ is called "computable" if the sequence $\{x, x, x, \ldots\}$ is computable.

The following simple definition will be used not only in the axioms, but throughout the remainder of the book. It is an obvious adaptation of the notion of effective convergence (which was discussed in Chapter 0) to Banach spaces.

Definition. The double sequence $\{x_{nk}\}$ converges to the sequence $\{x_n\}$ as $k \to \infty$, effectively in k and n, if there exists a recursive function e such that

$$\|x_{nk} - x_n\| \leqslant \frac{1}{2^N}$$

for all $k \geqslant e(n, N)$.

Now we state the axioms. The first one covers finite linear combinations.

Axiom 1 (Linear Forms). Let $\{x_n\}$ and $\{y_n\}$ be computable sequences in X, let $\{\alpha_{nk}\}$ and $\{\beta_{nk}\}$ be computable double sequences of real or complex numbers, and let d: $\mathbb{N} \to \mathbb{N}$ be a recursive function. Then the sequence

$$s_n = \sum_{k=0}^{d(n)} (\alpha_{nk} x_k + \beta_{nk} y_k)$$

is computable in X.

Axiom 2 (Limits). Let $\{x_{nk}\}$ be a computable double sequence in X such that $\{x_{nk}\}$ converges to $\{x_n\}$ as $k \to \infty$, effectively in k and n. Then $\{x_n\}$ is a computable sequence in X.

Axiom 3 (Norms). If $\{x_n\}$ is a computable sequence in X, then the norms $\{\|x_n\|\}$ form a computable sequence of real numbers.

We assume that at least one computable sequence in X exists, whence the sequence $\{0, 0, 0, \ldots\}$ is computable.

As corollaries we derive the following, which were given as axioms in the authors' papers [1983b, 1987].

Proposition. *The following are consequences of the axioms.*

(1) (*Composition property*). *If $\{x_n\}$ is a computable sequence in X, and a: $\mathbb{N} \to \mathbb{N}$ is a recursive function, then $\{x_{a(n)}\}$ is a computable sequence in X.*

(2) (*Insertion property*). *If $\{x_n\}$ and $\{y_n\}$ are computable sequences in X, then the sequence $\{x_0, y_0, x_1, y_1, x_2, y_2, \ldots\}$ is computable.*

Proofs. For (1): Let $y_n = 0$ for all n. Let $d(n) = a(n)$, and let $\{\alpha_{nk}\}$ be the computable double sequence of reals given by:

$$\alpha_{nk} = 1 \qquad \text{if } k = a(n), \qquad 0 \text{ otherwise.}$$

Then

$$s_n = \sum_{k=0}^{d(n)} \alpha_{nk} x_k = x_{a(n)}.$$

Hence by the Linear Forms Axiom, $\{x_{a(n)}\}$ is computable.

For (2): Let $\{x_n\}$ and $\{y_n\}$ be the given sequences. Let $d(n) = n$, and let $\{\alpha_{nk}\}$ and $\{\beta_{nk}\}$ be the computable real sequences given by:

$$\alpha_{nk} = 1 \quad \text{if } n \text{ is even and } k = n/2, \quad 0 \text{ otherwise;}$$
$$\beta_{nk} = 1 \quad \text{if } n \text{ is odd and } k = (n-1)/2, \quad 0 \text{ otherwise.}$$

Then

$$s_n = \sum_{k=0}^{d(n)} (\alpha_{nk} x_k + \beta_{nk} y_k) = \begin{cases} x_{n/2} & \text{if } n \text{ is even,} \\ y_{(n-1)/2} & \text{if } n \text{ is odd.} \end{cases}$$

Again by the Linear Forms Axiom, the interspersed sequence $\{x_0, y_0, \ldots\}$ is computable. □

We conclude this section by recalling some definitions which will play a key role in many of the results which follow.

Let $\{e_n\}$ be any sequence of vectors. We recall that the *linear span* of $\{e_n\}$ is the set of all finite (real/complex) linear combinations of the e_n. For example, if $\{e_n\}$ is the sequence of monomials $1, x, x^2, x^3, \ldots$, then the linear span of $\{e_n\}$ is the set of all polynomials.

Definition. Let X be a Banach space with a computability structure $\mathscr{S}$. We say that $\langle X, \mathscr{S} \rangle$ is *effectively separable* if there exists a computable sequence $\{e_n\}$, $e_n \in X$, such that the linear span of $\{e_n\}$ is dense in X. Such a sequence $\{e_n\}$ is called an *effective generating set* for $\langle X, \mathscr{S} \rangle$.

By an abuse of notation we shall sometimes refer to the Banach space X itself as being effectively separable.

Remark. We do not assume above that $\{e_n\}$ is "effectively dense"—merely that it is dense. Actually the effective density condition follows automatically. This will be proved in the Effective Density Lemma (Theorem 1, Section 5). The effective density condition was redundantly made part of the definition in the authors' paper [1983b].

2. *The Classical Case: Computability in the Sense of Chapter* 0

It is not hard to show that the computable sequences of computable functions of Chapter 0 satisfy the axioms of a computability structure. For let a and b be recursive reals, and let $C[a, b]$ be the Banach space of all continuous functions on $[a, b]$ with the uniform norm. Suppose further that $\mathscr{S}$ is the collection of all

computable sequences of computable functions as defined in Chapter 0. It is trivial to verify that Axiom 1 (Linear Forms) holds for $\mathscr{S}$. Furthermore the Limit Axiom (Axiom 2) is Theorem 4, and the Norm Axiom (Axiom 3) is Theorem 7 of that chapter.

It is just as easy to prove that $C[a, b]$ is effectively separable with respect to $\mathscr{S}$. For the monomials $1, x, x^2, \ldots$ form a computable sequence whose linear span is dense in $C[a, b]$. Thus $\{x^n\}$ is an effective generating set.

Another effective generating set is the collection of all piecewise linear functions in $C[a, b]$ with rational (complex rational) coordinates for the "corners".

A third example applies to Fourier series. Here, for convenience, let $[a, b] = [0, 2\pi]$. In this section, where the norm is that of uniform convergence, we must restrict attention to continuous functions f such that $f(0) = f(2\pi)$. [In Section 3 below, where we deal with L^p, the condition $f(0) = f(2\pi)$ can, of course, be dropped.] Now the effective generating set which applies most naturally to this situation is the sequence of trigonometric functions:

$$1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots.$$

It is worth noting that by Theorem 6 of Chapter 0 the effective generating set $\{x^n\}$ has a stronger property: every computable sequence $\{f_n\}$ is the *effective* limit of a computable sequence of polynomials of the form

$$p_{nk} = \sum_{j=0}^{d(n,k)} a_{nkj} x^j,$$

where $\{a_{nkj}\}$ is a computable triple sequence of rationals and d is a recursive function. We will see in Section 5 below, that this property generalizes to arbitrary effectively separable Banach spaces. For suppose $\{e_n\}$ is an effective generating set for $(X, \mathscr{S})$. Then, by the Effective Density Lemma, every computable sequence $\{x_n\} \in X$ is the effective limit of a computable sequence of polynomials

$$p_{nk} = \sum_{j=0}^{d(n,k)} a_{nkj} e_j.$$

3. *Intrinsic L^p-computability*

The definition of intrinsic L^p-computability is a natural generalization of computability as defined in Chapter 0.

We begin with a brief comment on notation. Let $1 \leqslant p < \infty$ be a computable real; let a and b be computable reals.

We now turn to $L^p[a, b]$. For $p = \infty$ we take the space $C[a, b]$ with the uniform norm. Recall that classically $C[a, b]$ is dense in $L^p[a, b]$. Thus the following definition suggests itself.

Definition. (a) A function $f \in L^p[a, b]$ is L^p-computable if there exists a sequence $\{g_k\}$ of continuous functions which is computable (in the sense of Chapter 0) and such that the L^p-norms $\|g_k - f\|_p$ converge to zero effectively as $k \to \infty$.

(b) A sequence $\{f_n\}$ of L^p-functions is L^p-computable if there exists a double sequence $\{g_{nk}\}$, which is computable (in the sense of Chapter 0) and such that $\|g_{nk} - f_n\|_p$ converges to zero as $k \to \infty$, effectively in n and k.

Computability for $L^p(I^q)$—where I^q is a computable rectangle of $\mathbb{R}^q$ (see Section 3, Chapter 0)—is defined similarly.

It is not hard to verify that the computable sequences of $L^p[a, b]$ defined above satisfy the axioms of a computability structure. Furthermore, this structure is effectively separable. Three effective generating sets are inherited from the computability structure on $C[a, b]$. They are: the monomials $1, x, x^2, \ldots$; the collection of all piecewise linear functions in $C[a, b]$ with rational coordinates for the "corners"; and the trigonometric polynomials. (See the preceding section.)

There is, however, a fourth effective generating set for $L^p[a, b]$, with $p \neq \infty$. This is the collection of all step functions with rational values and jump points. We observe that step functions could not have been used in conjunction with $C[a, b]$, since they are not continuous. In fact, it has been part of the accepted "folk wisdom" of this subject that a discontinuous function cannot be computable. This holds, however, only so long as we view the computation of a function as being carried out pointwise. Now, for L^p-functions, pointwise evaluation is not well-defined, since an L^p-function is determined only almost everywhere. By contrast, for functions in $C[a, b]$, where we use the uniform norm, pointwise evaluation makes perfectly good sense. Indeed uniform convergence implies pointwise convergence.

We now turn to noncompact domains. For $p \neq \infty$, $L^p(\mathbb{R})$ is our prototype. For $p = \infty$ our space is $C_0(\mathbb{R})$, the collection of all continuous functions which approach zero as $|x| \to \infty$.

The definition of L^p-computability on $\mathbb{R}$ is an extension of the definition of L^p-computability on $[a, b]$. We require that the functions g_k and g_{nk} have compact supports which vary effectively in n and k. Without loss of generality, we can assume that the k^{th} function is supported on a disk of radius k. This gives

Definition. A sequence $\{f_n\} \in L^p(\mathbb{R})$ is computable if there exists a sequence $\{g_{nk}\}$, computable in the sense of Chapter 0, such that

(a) the support of g_{nk} is included in $[-k, k]$,

(b) $\|f_n - g_{nk}\|_p \to 0$ as $k \to \infty$, effectively in n and k.

Computability for $L^p(\mathbb{R}^q)$ is defined similarly.

Note. Suppose $p = \infty$. If $f \in C[a, b]$, then the first definition gives the computable functions and sequences on $C[a, b]$ as defined in Chapter 0. If $f \in C_0(R)$, then the second definition shows that f is computable if and only if f is computable as in Section 3 of Chapter 0, and $f(x) \to 0$ effectively as $|x| \to \infty$.

Once again it is easy to see that intrinsic computability on $L^p(\mathbb{R})$, $1 \leqslant p \leqslant \infty$, satisfies the axioms of a computability structure. Furthermore, the computability

structure is effectively separable. Note however, that the monomials do not form an effective generating set, as $x^n \notin L^p(\mathbb{R})$. Similarly for trigonometric polynomials. However if we take those continuous functions with compact support which are piecewise linear and have rational coordinates at their "corners", then we obtain an effective generating set. For $p \neq \infty$, the step functions with rational jump points and values also are an effective generating set.

4. *Intrinsic l^p-computability*

As in functional analysis, let l^p (for $1 \leqslant p < \infty$) denote the space of all real (or complex) sequences $\{c_k\}$ such that the norm $(\Sigma|c_k|^p)^{1/p}$ is finite. Of course, we assume that p is a computable real. For $p = \infty$ (which corresponds to the sup norm), our space is l_0^∞, the space of all sequences $\{c_k\}$ which converge to zero as $k \to \infty$.

Definition. (a) Let $\{c_k\} \in l^p$. Then $\{c_k\}$ is l^p-computable if $\{c_k\}$ is a computable sequence of real or complex numbers and $\Sigma|c_k|^p$ converges effectively.

Let $\{c_k\} \in l_0^\infty$. Then $\{c_k\}$ is l_0^∞-computable if $\{c_k\}$ is a computable sequence of real or complex numbers which converges effectively to zero as $k \to \infty$.

(b) Let $\{x_n\}$ be a sequence of elements in l^p (i.e. $x_n = \{c_{nk}\}$, where each c_{nk} is a number). Then $\{x_n\}$ is l^p-computable if (i) $\{c_{nk}\}$ is a computable double sequence of numbers, (ii) for each n, $\Sigma_k|c_{nk}|^p$ converges effectively in both k and n. Similarly if $\{x_n\}$ is a sequence of l_0^∞, then $\{x_n\}$ is l_0^∞-computable if $\{c_{nk}\}$ is a computable sequence, such that $\{c_{nk}\}$ converges to zero as $k \to \infty$, effectively in k and n.

It is simple to verify that the axioms of a computability structure hold for these theories. Furthermore all of these spaces are effectively separable. As an effective generating set, we can take the sequence of vectors $e_n = \{0, 0, 0, \ldots, 0, 1, 0, 0, \ldots\}$ with a 1 in the n^{th} place.

5. *The Effective Density Lemma and the Stability Lemma*

In this section we prove two basic lemmas. The first of these, the Effective Density Lemma, states roughly that an effective generating set is not merely dense but "effectively dense". This fact is quite useful. It allows us to conclude without further proof, that certain (classical) approximation theorems have effective versions. For example, by the (classical) Weierstrass approximation theorem, the polynomials with rational coefficients are dense in $C[0, 1]$. Since these polynomials form an effective generating set, we know by the Effective Density Lemma that the Effective Weier-

strass Theorem (Theorem 6 of Chapter 0) holds. The reader may recall the long and somewhat messy proof given in that chapter.

The second lemma, the Stability Lemma, was discussed briefly in the introduction to this chapter. As a consequence of this lemma, we have a kind of "uniqueness" for computability structures on a Banach space. The lemma actually shows that the specification of an effective generating set gives a computability structure (= collection of computable sequences) which has no proper extension or restriction. Thus it is not possible to invent new axioms which, together with the axioms of a computability structure, give rise to a more intuitive notion of computability.

The Stability Lemma is an immediate corollary of the Effective Density Lemma.

Theorem 1 (Effective Density Lemma). *Suppose $\{e_n\}$ is an effective generating set for X. Then the sequence $\{x_n\}$, $x_n \in X$, is computable if and only if there is a computable double sequence p_{nk} such that*

(i) $$p_{nk} = \sum_{j=0}^{d(n,k)} \alpha_{nkj} e_j,$$

where $\{\alpha_{nkj}\}$ is a computable triple sequence of rationals/complex rationals, and $d(n, k)$ is a recursive function—

(ii) $$p_{nk} \to x_n \quad as \quad k \to \infty, \text{ effectively in } k \text{ and } n.$$

Proof. The "if" part follows immediately from the Linear Forms Axiom and the Limit Axiom. To prove the "only if" part we argue in the following way. Let $\{p_i\}$ be an effective listing of all finite linear combinations of the e_n with rational (complex rational) coefficients. By the Linear Forms Axiom, $\{p_i\}$ is computable in X. Thus $p_i - x_n$ is a computable double sequence, and by the Norm Axiom, $\{\|p_i - x_n\|\}$ is computable in $\mathbb{R}$. But the linear span of $\{e_n\}$ is dense in X. So there exists an i such that $\|p_i - x_n\| < 2^{-k}$. In order to compute p_{nk}, fix x_n and generate the p_i until one satisfying $\|p_i - x_n\| < 2^{-k}$ is found. Then let $p_{nk} = p_i$. Since the index $i = i(n, k)$ is a recursive function of n and k, we have, by the Composition Property (see Section 1 of this chapter) that $\{p_{nk}\}$ is computable in X. □

Corollary 1a (Effective Weierstrass Theorem). *Let I^q be a q-dimensional rectangle with computable coordinates for the "corners". Let X be the Banach space $C(I^q)$ with the uniform norm. Then $f \in C(I^q)$ is computable if and only if there is a computable sequence of polynomials $p_k(x_1, \ldots, x_q)$ with rational coeffficients which converges uniformly to f, effectively in k.*

Similarly $\{f_n\}$ is a computable sequence of $C(I^q)$ if and only if there is a computable double sequence $p_{nk}(x_1, \ldots, x_q)$ with rational coefficients which converges uniformly to f_n as $k \to \infty$, effectively in k and n.

Proof. An immediate corollary of the Effective Density Lemma. The sequence of monomials $\{x_1^{n_1}, x_2^{n_2}, \ldots, x_q^{n_q}\}$ is an effective generating set for $C(I^q)$. □

As deeper applications in the same vein, we give effective versions of two famous approximation theorems: the Stieltjes, Hamburger, Carleman Theorem and the Wiener Tauberian Theorem.

For the first of these: fix a computable real number $\alpha > 0$. Let $G^\alpha(\mathbb{R})$ denote the Banach space of all continuous function f on $(-\infty, \infty)$ such that $\lim_{x \to \pm\infty} f(x)e^{-|x|^\alpha} = 0$, endowed with the norm

$$\|f\| = \sup_x |f(x)|e^{-|x|^\alpha}.$$

We put a computability structure on $G^\alpha(\mathbb{R})$ by defining a sequence of continuous functions $\{f_n\}$ to be *computable* if:

(i) the sequence $\{f_n\}$ is computable in the sense of Chapter 0 and

(ii) $\lim_{x \to \pm\infty} f_n(x)e^{-|x|^\alpha} = 0$, effectively as $x \to \pm\infty$ and effectively in n.

Corollary 1b (Effective Stieltjes, Hamburger, Carleman Theorem). *Let α be a computable real, $\alpha \geqslant 1$. Let f be a continuous function on $(-\infty, \infty)$ which is computable in $G^\alpha(\mathbb{R})$. Then there exists a computable sequence of polynomials which converges effectively to f in the norm $\|h\| = \sup_x |h(x)|e^{-|x|^\alpha}$.*

Proof. If we ignore the question of effective convergence, this is a famous classical theorem (cf. Shohat, Tamarkin [1943]). But the effectiveness of the convergence follows immediately from the Effective Density Lemma.

Similarly, and with no extra effort, we obtain:

Corollary 1c (Effective Wiener Tauberian Theorem). *Let g be a computable function in $L^1(\mathbb{R})$ whose Fourier transform $\hat{g}(t) \neq 0$ for all real t. Let f be an arbitrary computable function in $L^1(\mathbb{R})$. Then there exists a computable sequence of rational linear combinations of translates of g which converges effectively to f in L^1 norm.*

Proof. An immediate consequence of the classical (noneffective) Wiener Tauberian Theorem, together with the Effective Density Lemma.

Theorem 2 (Stability Lemma). *Let $\{e_n\}$ be a sequence whose linear span is dense in the Banach space X. Let $\mathscr{S}'$ and $\mathscr{S}''$ be computability structures on X such that $\{e_n\} \in \mathscr{S}'$ and $\{e_n\} \in \mathscr{S}''$. Then $\mathscr{S}' = \mathscr{S}''$.*

Proof. This is an immediate consequence of the Effective Density Lemma. For that lemma applies to *any* computability structure $\mathscr{S}$. It gives—in terms of the sequence $\{e_n\}$—a precise characterization of the collection of computable sequences $\{x_n\} \in \mathscr{S}$. □

As a corollary we obtain the following proposition.

Corollary 2a. *Let $\{e_n\}$ be a sequence whose linear span is dense in the Banach space X. Let $\mathscr{S}, \mathscr{S}'$ be computability structures on X such that $\{e_n\} \in \mathscr{S}'$ and $\mathscr{S}' \subseteq \mathscr{S}$. Then $\mathscr{S}' = \mathscr{S}$.*

6. *Two Counterexamples: Separability Versus Effective Separability and Computability on* $L^\infty[0, 1]$

Throughout this book, the notion of effective separability will play an important role. Since most of the Banach spaces we deal with are obviously separable, one might wonder whether the hypothesis of effective separability is redundant. Our first example shows it is not redundant: we exhibit a Banach space with a computability structure which is separable but not effectively separable.

The second example involves a Banach space, $L^\infty[0, 1]$, which has no natural computability structure at all.

Recall that a Banach space X is effectively separable if it contains an effective generating set $\{e_n\}$—i.e. a computable sequence $\{e_n\}$ whose linear span is dense in X.

Example 1. There exists a Hilbert space H with a computability structure $\mathscr{S}$ such that:

(1) The space H is separable.
(2) The computable elements of H are dense in H.
(3) The pair $\langle H, \mathscr{S}\rangle$ is not effectively separable.

We mention that condition (2) above eliminates trivial and inconsequential examples such as the following: Take a closed subspace H_0 of H, and define "computability" only for sequences $\{x_n\}$ in H_0. Conditions (2) and (3) together mean that there is something "seriously wrong" with the computability structure $\mathscr{S}$.

Proof. Now we develop the example. For H we take the Hilbert space $L^2[0, 2\pi]$. It is the computability structure $\mathscr{S}$ which is nonstandard. In brief, we shall use a definition of "computability" which—compared to the standard definition—is too stringent.

We define a sequence of functions $\{f_n\}$ in $L^2[0, 2\pi]$ to be "ultra-computable" if and only if $\{f_n\}$ is a sequence of trigonometric polynomials of *uniformly bounded degrees* (and, of course, with computable coefficients). More precisely, we say that $\{f_n\}$ is "ultra-computable" if, for some integer N (which may depend on the sequence $\{f_n\}$):

$$f_n(x) = \sum_{m=-N}^{N} c_{nm} e^{imx},$$

where $\{c_{nm}\}$ is a computable double sequence of complex numbers.

Ultra-computability satisfies all of the axioms for a computability structure on a Banach space. The Linear Forms Axiom is obvious, since any linear combination of trigonometric polynomials of degree $\leqslant N$ is again a trigonometric polynomial of degree $\leqslant N$. The Norm Axiom is also obvious, since "ultra-computability" implies "ordinary L^2-computability", for which we know that the Norm Axiom holds.

The Limit Axiom is a little harder. We must show that if $\{f_{nk}\}$ is ultra-computable and converges effectively in L^2-norm to $\{f_n\}$ as $k \to \infty$, then $\{f_n\}$ is ultra-computable. Now by definition, if $\{f_{nk}\}$ is ultra-computable, then the functions f_{nk} are all

trigonometric polynomials of degree $\leqslant N$. This is because a double sequence $\{f_{nk}\}$ is computable if and only if the single sequence $\{h_j\}$ given by $h_{j(n,k)} = f_{nk}$ is computable —where j is a standard pairing function from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. It is now easy to see that the limit sequence $\{f_n\}$ is also a sequence of trigonometric polynomials of degree $\leqslant N$. For a function $g \in L^2[0, 2\pi]$ is a trigonometric polynomial of degree $\leqslant N$ if and only if the inner products

$$(g, e^{ipx}) = 0$$

for all p with $|p| > N$. Now by Schwarz' inequality (i.e. $|(g, h)| \leqslant \|g\| \|h\|$) we deduce: If the condition $(g, e^{ipx}) = 0$ holds for $g = f_{nk}$, then since $f_{nk} \to f_n$ in L^2-norm, the condition $(g, e^{ipx}) = 0$ also holds for the limits $g = f_n$. Hence the f_n are trigonometric polynomials of degree $\leqslant N$, as desired.

It is trivial to verify that the sequence of coefficients $\{c_{nm}\}$ of $\{f_n\}$ is computable.

Of course the space $L^2[0, 2\pi]$ is separable. It is easy to see that, in terms of ultra-computability, this space is not effectively separable. That is, there is no ultra-computable effective generating set $\{e_n\}$. Indeed any ultra-computable sequence $\{e_n\}$ must, by definition, consist only of trigonometric polynomials of degree $\leqslant N$. Then the closed linear span of $\{e_n\}$ will still consist only of trigonometric polynomials of degree $\leqslant N$, and cannot give all of $L^2[0, 2\pi]$.

Finally, the ultra-computable elements form a dense subset of $L^2[0, 2\pi]$, since every trigonometric polynomial with complex-rational coefficients is ultra-computable. □

Remarks. The preceding paragraph harks back to Chapter 0, Section 2, where we already saw a clear distinction between "pointwise computability" (i.e. computability of the individual elements of a sequence) and "sequential computability" (which is a global statement involving the sequence as a whole). The notion of ultra-computability used above gave a wide class of single computable elements: namely all trigonometric polynomials with computable coefficients. But the definition of sequential computability, requiring a *fixed* degree N (or less) for all of the elements of the sequence, was ultra-severe. We could not have an effective generating set which was ultra-computable, because the restriction to a fixed bound on the degrees made it impossible to span all of $L^2[0, 2\pi]$.

We turn now to $L^\infty[0, 1]$, the space of all bounded measurable functions on $[0, 1]$. This space is not separable at all. We shall show that it has no "natural" computability structure. To do this, we define condition (C) as follows:

(C): If $\{a_n\}$ and $\{b_n\}$ are computable sequences of reals with $a_n < b_n$, then the sequence of characteristic functions $\chi_{[a_n, b_n]}$ is computable.

As noted in Section 3, condition (C) holds in $L^p[0, 1]$ where $1 \leqslant p < \infty$ and p is computable. However it must fail for $L^\infty[0, 1]$, as the following example shows.

Example 2. There is no computability structure on $L^\infty[0, 1]$ which satisfies condition (C).

Proof. Let d: $\mathbb{N} \to \mathbb{N}$ be a $1-1$ recursive function enumerating a recursively enumerable nonrecursive set D. Let $a_n = 0$ for all n. Let

$$b_n = \begin{cases} \frac{1}{2} & \text{if } n \notin D, \\ \frac{1}{2} + \frac{1}{2^m} & \text{if } n = d(m),\ m \geqslant 1, \\ 1 & \text{if } n = d(0). \end{cases}$$

This description of b_n, as given, is not effective. However, $\{b_n\}$ is a computable sequence of reals. For we can approximate b_n—effectively—to within an error of 2^{-N} by computing $d(m)$ for $0 \leqslant m \leqslant N$. If one of these values $d(m) = n$, then we know b_n exactly: $b_n = (1/2) + (1/2^m)$ (or $b_n = 1$ if $m = 0$). If $d(m) \neq n$ for $0 \leqslant m \leqslant N$, then the value $1/2$ approximates b_n to within an error $< 2^{-N}$.

Suppose that there were a computability structure on $L^\infty[0, 1]$ which satisfied (C). Then $\{\chi_{[a_n, b_n]}\}$ would be computable. Similarly we would have that $\chi_{[1/2, 1]}$ is computable. Thus, by the Linear Forms Axiom, $\{\chi_{[a_n, b_n]} + \chi_{[1/2, 1]}\}$ would be computable.

The L^∞ norms of this sequence are:

$$\begin{array}{ll} 2 & \text{if } n \in D, \\ 1 & \text{if } n \notin D. \end{array}$$

By the Norm Axiom this sequence should be computable. Clearly it is not. □

7. *Ad Hoc Computability Structures*

In Sections 2, 3 and 4 we defined the "natural" computability structures on $C[a, b]$, L^p and l^p. These structures are called *intrinsic*. Moreover, in Section 5 we showed that an arbitrary computability structure (i.e. any structure satisfying the axioms) must coincide with the intrinsic structure, provided only that the two structures possess a computable dense sequence in common.

The above-mentioned results lead to the following questions.

1) Do non-intrinsic computability structures exist?
2) Do non-intrinsic computability structures—if they exist—serve any useful purpose?
3) What is the relation between the intrinsic and non-intrinsic computability structures?

This section is conerned with the answer to 1). In later chapters we shall answer 2) and 3). The answers are respectively: 1) Yes, non-intrinsic structures do exist;

2) they are useful for the creation of counterexamples (Chapter 4, Sections 5 and 6); 3) the non-intrinsic structures are isomorphic to the intrinsic one for Hilbert space, but not for Banach spaces in general (Chapter 4, Sections 6 and 7).

We now turn to the first question. We give two simple examples of ad hoc computability structures. We note in passing that both of these structures are effectively separable.

Example 3. (An ad hoc computability structure for $C[0, 1]$).

Let $C[0, 1]$ be the class of continuous *complex-valued* functions on $[0, 1]$. Let $\mathscr{S}_0$ be the intrinsic computability structure defined in Section 2. Then there exists an ad hoc computability structure $\mathscr{S}_1$ on $C[0, 1]$ such that $\mathscr{S}_0$ and $\mathscr{S}_1$ have only one sequence in common—the sequence $\{0, 0, 0, \ldots\}$.

Proof. Let c be a noncomputable complex number such that $|c| = 1$. (The existence of such a number follows from cardinality arguments, since there are uncountably many numbers on the unit circle.) The computability structure $\mathscr{S}_1$ is defined as follows. The sequence $\{f_n^c\}$ is computable in $\mathscr{S}_1$ if there exists a computable sequence $\{f_n\}$ of $\mathscr{S}_0$ such that

$$f_n^c = cf_n.$$

It is trivial to show that the sequences $\{f_n^c\}$ satisfy the axioms of a computability structure. We now show that $\mathscr{S}_0 \cap \mathscr{S}_1 = \{0, 0, 0, \ldots, \}$.

Suppose there exists a function $g^c \neq 0$ which is an element of a sequence in $\mathscr{S}_0$. Then $g^c(\xi) \neq 0$ for some computable real ξ. Now since $g^c(x) = cg(x)$ for some $g \in \mathscr{S}_0$, we have $g^c(\xi) = cg(\xi)$. Since both g and g^c are members of $\mathscr{S}_0$, $g(\xi)$ and $g^c(\xi)$ are both computable complex numbers. Hence c is computable, a contradiction. □

The previous example involved complex-valued functions. It is possible to obtain ad hoc structures for real Banach spaces. The following example shows how translation of the domain can lead to such structures.

Example 4. (An ad hoc computability structure for $C_0(R)$.)

Let $C_0(R)$ be the space of all continuous real-valued functions on R which approach 0 as $|x| \to \infty$. Let $\mathscr{S}_0$ be the intrinsic computability structure for $C_0(R)$ which is described in Section 3. Then there exists a computability structure $\mathscr{S}_1$ for $C_0(R)$ so that $\mathscr{S}_0 \neq \mathscr{S}_1$.

Proof. Let α be a noncomputable real. Define $\mathscr{S}_1$ as follows. The sequence $\{f_n^\alpha\}$ is computable in $\mathscr{S}_1$ if there exists a computable sequence $\{f_n\}$ of $\mathscr{S}_0$ such that

$$f_n^\alpha(x) = f_n(x + \alpha).$$

It is obvious that the sequences $\{f_n^\alpha\}$ satisfy the axioms of a computability structure. Furthermore $\mathscr{S}_1$ contains the sequence $\{g_n\}$, where

$$g_n(x) = e^{-(x+\alpha)^2} \qquad \text{for all } n,$$

which is not in $\mathscr{S}_0$. □

In chapters 4 and 5 we shall study the computability of eigenvalues and eigenvectors. We will find that, in general, the eigenvalues are computable, but the eigenvectors are not. The proof of the result for eigenvectors leans heavily on an ad hoc computability structure.