The sizes of consecutive repeat-free codes

Robin Hughes-Jones

Technical Report
RHUL–MA–2010–1
2 March 2010

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England

http://www.rhul.ac.uk/mathematics/techreports

Royal Holloway, University of London
Ph. D.

The sizes of consecutive repeat-free codes

Author:
Robin Hughes-Jones

Supervisor:
Prof. Simon Blackburn

August 31, 2009

Declaration of Authorship

I declare this is my own work, except as acknowledged.

Abstract

The notions of *strongly consecutive repeat free code* and *weakly consecutive repeat free code* were introduced by Pebody in his paper in the Journal of Combinatorial Theory Series A in 2006. This thesis aims to investigate the the maximum sizes of such codes, in particular in the case when the length is fixed and the alphabet size is large.

Pebody constructs a strongly consecutive repeat free code of maximal size, which he calls the *alternating code*. We show that the size of an alternating code is polynomial in the alphabet size, give methods for computing this polynomial and explicitly determine the most significant coefficients of this polynomial in terms of the sequence of 'up/down numbers' and related sequences.

Pebody defines a family of codes (which we call *Pebody codes*) that are weakly consecutive repeat free codes. Pebody conjectures that for all parameters there exists a member of this family that is a weakly consecutive repeat free code of maximal size. We show that the maximal size of a Pebody code agrees closely with the maximal size of a strongly consecutive repeat free code. We use techniques from combinatorics and functional analysis, together with computational results, to give estimates for the leading terms of the maximal size of a Pebody code of fixed length when the alphabet size is large.

Contents

Chapter 1

Introduction

For a positive integer b, let $[b] := \{1, 2, 3, \ldots, b\}$ be our *alphabet*. Let n be a positive integer, then a code C is any subset of $[b]^n$. An element $w \in C$ shall be known as a *word* (of *length* n) and for a positive integer $i \leq n$ the ith ordinate shall be denoted w_i. In other words

$$w = (w_1, w_2, w_3, \ldots, w_n).$$

We say that i is an index.

Definition 1.0.1. *For positive integers b and n, let w be a word in $[b]^n$. We say that w is* up/down *if, for any index $i < n$, we have $w_i < w_{i+1}$ when i is odd and $w_i > w_{i+1}$ when i is even. One could write*

$$w_1 < w_2 > w_3 < \ldots w_n.$$

1.1 Consecutive repeat-free codes and Pebody's conjecture

In a private communication with Luke Pebody, János Körner and Gábor Simonyi introduced the ideas of *weakly* and *strongly consecutive repeat free codes* and asked how large such codes could be. We provide the definitions for these codes (taken from [Peb06]) here but we do not deal with them directly in the thesis.

Definition 1.1.1. *Let $C \subset [b]^n$ be a code. Then C is* weakly consecutive repeat-free *if for any words $v, w \in C$ and any index i such that $v_i \neq w_i$ and $v_{i+1} \neq w_{i+1}$, we have that the sets $\{v_i, w_i\}$ and $\{v_{i+1}, w_{i+1}\}$ are not equal.*

Definition 1.1.2. *Let $C \subset [b]^n$ be a weakly consecutive repeat-free code. Then C is a* strongly consecutive repeat-free code *if for any word $w \in C$ and any index i we have that $w_i \neq w_{i+1}$.*

Luke Pebody proves that the alternating code, defined below, is a largest strongly consecutive repeat-free code in [Peb06].

Definition 1.1.3. *Let b and n be positive integers. The* alternating code $A_{n,b}$ *is defined as*

$$\{w \in [b]^n : w \text{ is up/down}\}.$$

Pebody then gives a construction of a weakly consecutive repeat-free code, which we call a Pebody code, and makes the following conjecture.

Definition 1.1.4. *Let y be a word in $[b]^{n-1}$. Define the code B_y to be the set of all words $w \in [b]^n$ such that for any index $i < n$, $w_i \leq w_{i+1}$ if i is odd, $w_i \geq w_{i+1}$ if i is even and if $w_i = w_{i+1}$ then $w_i = y_i = w_{i+1}$. We shall refer to such a code B_y as a* Pebody code *and we shall say that it has* length n.

Conjecture 1.1.5. [*Pebody 2005*] *For any positive integers n and b there exists $y \in [b]^{n-1}$ such that B_y is a largest weakly consecutive repeat-free code.*

This thesis investigates the sizes of alternating codes and Pebody codes, especially in the case where n is fixed and b is large. In particular we look at which choices of $y \in [b]^{n-1}$ give largest Pebody codes B_y and how large these weakly consecutive repeat-free codes can be. The aim is to find good approximations for the sizes of alternating and Pebody codes and produce efficient methods to compute these sizes accurately.

1.2 Structure of thesis

We commence our work in the area of strongly consecutive repeat-free codes by analysing the size of the alternating code $A_{n,b}$ when n is fixed. One of our first results is Theorem 3.1.5 where we prove that $|A_{n,b}|$ is a polynomial in b. This motivates the following definition.

Definition 1.2.1. *Let n be a fixed positive integer. For an alternating code $A_{n,b}$, let $\alpha_{n,n}$, $\alpha_{n,n-1}$ and $\alpha_{n,n-2}$ be such that*

$$|A_{n,b}| = \alpha_{n,n}b^n + \alpha_{n,n-1}b^{n-1} + \alpha_{n,n-2}b^{n-2} + O(b^{n-3}).$$

Chapter 3 concludes (see Corollary 3.2.12) by finding an explicit formula for $\alpha_{n,n}$ in terms of a previously studied sequence of numbers variously known as up/down numbers (see [BR90]), zig-zag numbers (see [BR09]) and Euler numbers (see [KB67]). Similar formulae for $\alpha_{n,n-1}$ and $\alpha_{n,n-2}$ are derived in the same corollary using sequences related to up/down numbers.

In Chapter 4 we focus on the Pebody code B_y and its size. Theorem 4.2.9 shows us that there exist polynomials $C_{n,k}(f)$ defined over the interval $[0, 1]$ such that

$$|B_y| = |A_{n,b}| + \sum_{k=1}^{n-1} C_{n,k}(f_k)b^{n-2} + O(b^{n-3})$$

where $f_k := \frac{y_k}{b}$. This allows us to give the following definition.

Definition 1.2.2. *Let n and b be positive integers with b much larger than n. Define $B_{n,b}$ to be a largest Pebody code. So*

$$|B_{n,b}| = \max_{y \in [b]^{n-1}} |B_y|.$$

Let us also define $\beta_{n,n-2}$ by

$$|B_{n,b}| = \alpha_{n,n} b^n + \alpha_{n,n-1} b^{n-1} + \beta_{n,n-2} b^{n-2} + O(b^{n-3}).$$

In order to determine $\beta_{n,n-2}$ we study the maximal values of the polynomials $C_{n,k}(f)$ over $[0, 1]$. In Chapter 5 we introduce the recursively defined polynomial $F_{n,i}(f)$ over $[0, 1]$ and show that we may write

$$C_{n,i}(f) = F_{-1,-i}(f) F_{n,i+1}(f).$$

By developing a Fourier series and with techniques from the theory of Hilbert spaces, we write down explicitly (see Theorem 5.2.14 and Theorem 5.2.19) the functions that the polynomials $F_{n,i}$ converge to in some useful sense. In Chapter 6 we use these functions to approximate the $F_{n,i}$ and so are able to approximate $C_{n,k}$ and derive upper and lower bounds for $\beta_{n,n-2}$ and hence Pebody codes in general. Finally, we show how these techniques can be used to efficiently compute good approximations for $\beta_{n,n-2}$ on a computer. A range of approximations are made available, each one offering a different balance between accuracy and computation time.

The thesis concludes with Chapter 7 which is mostly practical in nature. We present here tables of computation times and accuracies achieved by putting into practice the approximations of Chapter 6 with programmes written in *Mathematica*. The computational results validate our theoretical bounds on the sizes of Pebody codes and give a practical demonstration of the trade-offs between complexity and accuracy for our approximation methods.

Chapter 2

Preliminaries

This short chapter contains elementary results about binomial coefficients and sums of polynomials, which we will need throughout the thesis.

Definition 2.0.3. *We use the following standard notation for binomial expressions:*

$$\binom{n}{i} := \frac{n!}{i!(n-i)!}.$$

Definition 2.0.4. *Let k be a nonnegative integer and n be a positive integer. Define the function $S_k(n)$ by*

$$S_k(n) := \sum_{m=1}^{n} m^k.$$

The following lemma was proved by Pascal in [Pas54] but the proof we give here has been adapted from the one in [Bea96].

Lemma 2.0.5. *Let k and n be as in Definition* 2.0.4 *then*

$$S_k(n) = \frac{1}{k+1}\left((n+1)^{k+1} - 1 - \sum_{r=0}^{k-1}\binom{k+1}{r} S_r(n)\right)$$

Proof. By a "sum of differences" argument we have that

$$\begin{aligned}
(n+1)^{k+1}-1 &= \sum_{m=1}^{n}\left((m+1)^{k+1}-m^{k+1}\right)\\
&= \sum_{m=1}^{n}\sum_{r=0}^{k}\binom{k+1}{r}m^r\\
&= \sum_{r=0}^{k}\binom{k+1}{r}S_r(n)\\
&= \binom{k+1}{k}S_k(n)+\sum_{r=0}^{k-1}\binom{k+1}{r}S_r(n)\\
&= (k+1)S_k(n)+\sum_{r=0}^{k-1}\binom{k+1}{r}S_r(n).
\end{aligned}$$

Solving for $S_k(n)$ completes the proof. □

As Beardon observes, one proves by induction that the function $S_k(n)$ is a polynomial of degree $k+1$ and this is given in Lemma 2.0.6.

Lemma 2.0.6. *Again let k and n be as in Definition 2.0.4. The function $S_k(n)$ is a polynomial in n of degree $k+1$.*

Proof. We shall prove this result by induction on k so let us first observe that the result holds for the case where $k=0$, since $S_0(n)=n$.

Assume then that the result holds for all nonnegative integers r such that $r<k$. Recall from Lemma 2.0.5 that

$$S_k(n)=\frac{1}{k+1}\left((n+1)^{k+1}-1-\sum_{r=0}^{k-1}\binom{k+1}{r}S_r(n)\right).$$

By assumption, each of the functions $S_r(n)$ in the summand are polynomials of degree $r+1$ and so the sum has degree at most k. Therefore the term of highest degree is $\frac{n^{k+1}}{k+1}$, which is obtained by the binomial expansion of $(n+1)^{k+1}$. Hence $S_k(n)$ is a polynomial and has degree $k+1$. □

Lemma 2.0.7. *For fixed k and as $n\to\infty$, the three leading terms of the polynomial $S_k(n)$ are $\frac{1}{k+1}n^{k+1}$, $\frac{1}{2}n^k$ and $\frac{k}{12}n^{k-1}$ so*

$$S_k(n)=\frac{1}{k+1}n^{k+1}+\frac{1}{2}n^k+\frac{k}{12}n^{k-1}+O(n^{k-2}).$$

Proof. Recall from Lemma 2.0.5 that

$$S_k(n)=\frac{1}{k+1}\left((n+1)^{k+1}-1-\sum_{r=0}^{k-1}\binom{k+1}{r}S_r(n)\right).$$

By Lemma 2.0.6 we have $S_r(n) = 0 + O(n^{k-2})$ for $r \leq k-3$ and so

$$\begin{aligned} S_k(n) &= \frac{(n+1)^{k+1}}{k+1} - \binom{k+1}{k-1}\frac{S_{k-1}(n)}{k+1} - \binom{k+1}{k-2}\frac{S_{k-2}(n)}{k+1} + O(n^{k-2}) \\ &= \frac{(n+1)^{k+1}}{k+1} - \frac{kS_{k-1}(n)}{2} - \frac{k(k-1)S_{k-2}(n)}{6} + O(n^{k-2}). \end{aligned} \tag{2.1}$$

Again, by Lemma 2.0.5 we have

$$\begin{aligned} S_{k-2}(n) &= \frac{1}{k-1}\left((n+1)^{k-1} - 1 - \sum_{r=0}^{k-3}\binom{k-1}{r}S_r(n)\right) \\ &= \frac{(n+1)^{k-1}}{k-1} + O(n^{k-2}) \\ &= \frac{n^{k-1}}{k-1} + O(n^{k-2}) \end{aligned}$$

and

$$\begin{aligned} S_{k-1}(n) &= \frac{1}{k}\left((n+1)^k - 1 - \sum_{r=0}^{k-2}\binom{k}{r}S_r(n)\right) \\ &= \frac{1}{k}\left((n+1)^k - \binom{k}{k-2}S_{k-2}(n)\right) + O(n^{k-2}) \\ &= \frac{1}{k}\left(n^k + kn^{k-1} - \frac{k(k-1)}{2}S_{k-2}(n)\right) + O(n^{k-2}) \\ &= \frac{n^k}{k} + n^{k-1} - \frac{(k-1)}{2}S_{k-2}(n) + O(n^{k-2}) \\ &= \frac{n^k}{k} + n^{k-1} - \frac{n^{k-1}}{2} + O(n^{k-2}) \\ &= \frac{n^k}{k} + \frac{n^{k-1}}{2} + O(n^{k-2}). \end{aligned}$$

Using these formulae with (2.1) gives us

$$\begin{aligned} S_k(n) &= \frac{(n+1)^{k+1}}{k+1} - \left(\frac{n^k}{2} + \frac{kn^{k-1}}{4}\right) - \frac{kn^{k-1}}{6} + O(n^{k-2}) \\ &= \frac{n^{k+1}}{k+1} + n^k + \frac{kn^{k-1}}{2} - \left(\frac{n^k}{2} + \frac{kn^{k-1}}{4}\right) - \frac{kn^{k-1}}{6} + O(n^{k-2}) \\ &= \frac{n^{k+1}}{k+1} + \frac{n^k}{2} + \frac{kn^{k-1}}{12} + O(n^{k-2}). \end{aligned}$$

$\square$

Chapter 3

The alternating code

3.1 Enumeration of the alternating code

For fixed n, the size of the alternating code grows as the alphabet size increases, since $A_{n,b} \subset A_{n,b+1}$. In this section we prove this growth is polynomial and, by studying the up/down numbers, go on to determine the degree and the three leading terms of this polynomial.

3.1.1 Enumeration using a "sum of sums" method

Definition 3.1.1. *Let w be a word of length n and l be a positive integer such that $l \leq n$. For the word w, define its* head *of length l as the word $w_1 w_2 \dots w_l$ and its* tail *of length l as the word $w_{n-l+1} w_{n-l+2} w_{n-l+3} \dots w_n$.*

The relatively simple structure of the words in the alternating code allows us to count the number of distinct heads of a given length for these words. Investigating heads of increasing length will give us some idea of how to enumerate $A_{n,b}$.

Lemma 3.1.2. *Let b and c be indeterminates and let l and m be nonnegative integers. The sums $\sum_{c=1}^{d-1} c^l b^m$ and $\sum_{c=d+1}^{b} c^l b^m$ are equal to $\frac{b^m d^{l+1}}{l+1} + p(b,d)$ and $\frac{b^{m+l+1} - b^m d^{l+1}}{l+1} + q(b,d)$ respectively, where p and q are polynomials of combined degree at most $l+m$.*

Proof. Recall from Lemma 2.0.6 that $S_l(d)$ is polynomial and so Lemma 2.0.7 implies that, for some polynomial $p_l(d)$ of degree l,

$$S_l(d) = \frac{d^{l+1}}{l+1} + p_l(d).$$

So

$$\begin{aligned}\sum_{c=1}^{d-1} c^l b^m =& b^m \left(\sum_{c=1}^{d} c^l - d^l\right)\\ =& b^m \left(\frac{d^{l+1}}{l+1} + p_l(d) - d^l\right)\\ =& \frac{b^m d^{l+1}}{l+1} + p(b,d)\end{aligned}$$

where $p(b,d) = b^m \left(p_l(d) - d^l\right)$ and

$$\begin{aligned}\sum_{c=d+1}^{b} c^l b^m =& b^m \left(\sum_{c=1}^{b} c^l - \sum_{c=1}^{d} c^l\right)\\ =& b^m \left(\frac{b^{l+1}}{l+1} + p_l(b) - \frac{d^{l+1}}{l+1} - p_l(d)\right)\\ =& \frac{b^{m+l+1} - b^m d^{l+1}}{l+1} + q(b,d),\end{aligned}$$

where $q(b,d) = b^m (p_l(b) - p_l(d))$ are polynomials of combined degree at most $m + l$. $\square$

The following definition shall be used extensively throughout the thesis.

Definition 3.1.3. *Let n be a positive integer. Define $P_{n,n} := 1$ and for any integer $i < n$ recursively define*

$$P_{n,i}(b,c) := \sum_{x=c+1}^{b} P_{n,i+1}(b,x)$$

when i is odd and

$$P_{n,i}(b,c) := \sum_{x=1}^{c-1} P_{n,i+1}(b,x)$$

when i is even. We restrict the domain of $P_{n,i}(b,c)$ to be $\mathbb{N} \times \mathbb{N}$ and shall only be interested in the case where $b \geq c$.

Lemma 3.1.4. *For positive i the function $P_{n,i}(b,c)$ counts the number of distinct tails of length $n-i+1$ with first letter c of the up/down words of length n. In other words, $P_{n,i}(b,c)$ enumerates the set*

$$\{v \in [b]^{n-i+1} : \textit{there exists } w \in A_{n,b} \textit{ such that } w_i w_{i+1} \dots w_n = v, v_1 = c\}.$$

Proof. Throughout this proof a tail shall be a tail of a word in $A_{n,b}$. If $i = n$ then our set is just $\{c\}$. Since $P_{n,n}(b,c) = 1$ by definition, the result holds in this case. For a proof by induction let us suppose that $i < n$ and the result holds for $i + 1$.

Thus the number of tails of length $n - i$ with first letter x is $P_{n,i+1}(b, x)$. We want to evaluate the number of tails of length $n - i + 1$ with first letter c.

Suppose i is odd. Any tail of length $n - i + 1$ with the first two letters c then x must have $c < x$. Thus we are interested in summing the number of tails of length $n - i$ with first letter x for x from $c + 1$ to b. This is

$$\sum_{x=c+1}^{b} P_{n,i+1}(b, x) = P_{n,i}(b, c).$$

Similarly, for when i is even we need $c > x$ and we sum for x from 1 to $c - 1$ giving us

$$\sum_{x=1}^{c-1} P_{n,i+1}(b, x) = P_{n,i}(b, c),$$

completing the induction. □

Theorem 3.1.5. *The size of the alternating code $A_{n,b}$ is a polynomial of degree n in b.*

Proof. If $n = 1$ then $A_{n,b} = [b]$. As the result clearly holds in this case, let us suppose $n \geq 2$.

Let w be a word in $A_{n,b}$. The letter w_1 cannot be b as $w_1 < w_2$ and b is the largest letter in our alphabet. As w_1 may take any other value in $[b]$, there are $b - 1$ possibilities for w_1. Since $w \in A_{n,b}$ we may consider it to be a tail of length n (of a word in $A_{n,b}$) with first letter w_1. Then, by Lemma 3.1.4, for each value of w_1 there are $P_{n,1}(b, w_1)$ possibilities for the rest of the word. Summing for each possible value of w_1 counts all the words in $A_{n,b}$ and we get

$$\sum_{w_1=1}^{b-1} P_{n,1}(b, w_1).$$

Note that this is actually $P_{n,0}(b, b)$ and so

$$|A_{n,b}| = P_{n,0}(b, b).$$

It suffices to prove therefore that the $P_{n,i}(b, c)$ are polynomials of combined degree $n - i$ in b and c.

We now use a proof by induction to show that the $P_{n,i}$ are polynomials with combined degree at most $n - i$. For the inductive step let us assume $P_{n,i+1}(b, w_{i+1})$ is a polynomial of combined degree $n - (i + 1)$.

For ease of notation let us define

$$\mathbb{N}_m := \{(j, k) \in \mathbb{Z} \times \mathbb{Z} : j \geq 0, k \geq 0, j + k \leq m\}$$

to be the set of all pairs of nonnegative integers (j, k) whose sum is at most m. We can now write our polynomial $P_{n,i+1}(b, w_{i+1})$ as $\sum_{(j,k)\in\mathbb{N}_{n-(i+1)}} \alpha_{jk} w^j b^k$ where the α_{jk} are real coefficients. When i is even:

$$\begin{aligned} P_{n,i}(b, w_i) &= \sum_{w_{i+1}=1}^{w_i-1} P_{n,i+1}(b, w_{i+1}) \\ &= \sum_{w_{i+1}=1}^{w_i-1} \sum_{(j,k)\in\mathbb{N}_{n-(i+1)}} \alpha_{jk} w_{i+1}^j b^k \\ &= \sum_{(j,k)\in\mathbb{N}_{n-(i+1)}} \left(\alpha_{jk} \sum_{w_{i+1}=1}^{w_i-1} w_{i+1}^j b^k \right). \end{aligned}$$

Using Lemma 3.1.2 the above becomes:

$$P_{n,i}(b, w_i) = \sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} \left(\frac{b^k w_i^{j+1}}{j+1} + p_{jk}(b, w_i) \right)$$

where the p_{jk} are polynomials of combined degree at most $j + k$. Splitting off the terms of highest combined degree we have

$$\begin{aligned} P_{n,i}(b, w_i) = & \sum_{l=0}^{n-i-1} \left(\alpha_{l(n-i-l-1)} \frac{b^{(n-i-l-1)} w_i^{l+1}}{l+1} \right) \\ & + \sum_{(j,k)\in\mathbb{N}_{n-i-2}} \left(\alpha_{jk} \frac{b^k w_i^{j+1}}{j+1} \right) + \sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} p_{jk}(b, w_i). \end{aligned}$$

Each summand of $\sum_{(j,k)\in\mathbb{N}_{n-i-2}} \left(\alpha_{jk} \frac{b^k w_i^{j+1}}{j+1} \right)$ has combined degree $k + j + 1$ where $(j, k) \in \mathbb{N}_{n-i-2}$. So $j + k \leq n - i - 2$ and the sum has combined degree at most $n - i - 1$. Each summand of $\sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} p_{jk}(b, w_i)$ has combined degree $j + k$ with $(j, k) \in \mathbb{N}_{n-i-1}$ and so $j + k \leq n - i - 1$ and so this sum also has combined degree at most $n - i - 1$.

The remaining terms in $P_{n,i}(b, w_i)$ all have combined degree $n-i$. By assumption, the polynomial $P_{n,i+1}(b, c)$ has degree $n-i-1$, hence at least one of the coefficients of the terms of combined degree $n-i-1$, say $\alpha_{l(n-i-l-1)}$, is nonzero. The coefficient of $b^{n-i-l-1} w^{l+1}$, which is $\frac{\alpha_{l(n-i-l-1)}}{l+1}$, is then nonzero. Therefore $P_{n,i}(b, w_i)$ has combined degree $n - i$. This completes the inductive step for when i is even.

Let us assume then that i is odd. In this case:

$$\begin{aligned}
P_{n,i}(b,w_i) &= \sum_{w_{i+1}=w_i+1}^{b} P_{n,i+1}(b,w_{i+1}) \\
&= \sum_{w_{i+1}=w_i+1}^{b} \sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} w_{i+1}^{j} b^{k} \\
&= \sum_{(j,k)\in\mathbb{N}_{n-i-1}} \left(\alpha_{jk} \sum_{w_{i+1}=w_i+1}^{b} w_{i+1}^{j} b^{k} \right).
\end{aligned}$$

Again by Lemma 3.1.2 this becomes:

$$P_{n,i}(b,w_i) = \sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} \left(\frac{b^{k+j+1} - b^{k} w_i^{j+1}}{j+1} + q_{jk}(b,w_i) \right)$$

where the q_{jk} are polynomials of combined degree at most $j+k$ and as before we split off the terms of combined degree $n-i$.

$$\begin{aligned}
P_{n,i}(b,w_i) = &\sum_{l=0}^{n-i-l} \left(\alpha_{l(n-i-l-1)} \frac{b^{n-i} - b^{n-i-1-l} w_i^{l+1}}{l+1} \right) \\
&+ \sum_{(j,k)\in\mathbb{N}_{n-i-2}} \left(\alpha_{jk} \frac{b^{k+j+1} - b^{k} w_i^{j+1}}{j+1} \right) \\
&+ \sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} q_{jk}(b,w_i).
\end{aligned}$$

Similarly to the previous case, each summand of

$$\sum_{(j,k)\in\mathbb{N}_{n-i-2}} \left(\alpha_{jk} \frac{b^{k+j+1} - b^{k} w_i^{j+1}}{j+1} \right)$$

has combined degree $j+k+1$ where $j+k \leq n-i-2$ and so the sum has combined degree at most $n-i-1$. Each summand of $\sum_{(j,k)\in\mathbb{N}_{n-i-1}} \alpha_{jk} q_{jk}(b,w_i)$ has combined degree $j+k$ and hence the sum has combined degree at most $n-i-1$. Again, all the remaining terms of $P_{n,i}(b,w_i)$ have combined degree $n-i$. By assumption $P_{n,i+1}(b,c)$ has degree $n-i-1$ and so for some l we have that $\alpha_{l(n-i-l-1)}$ is nonzero. Therefore the coefficient of $b^{n-i-l-1}w^{l+1}$, which is $-\frac{\alpha_{l(n-i-l-1)}}{l+1}$, is nonzero and $P_{n,i}(b,w_i)$ has combined degree $n-i$ as required.

In order to complete our induction we see that, in the base case where $P_{n,n}=1$, the degree is zero. Hence $P_{n,0}(b,b)$, and therefore $|A_{n,b}|$, is a polynomial of degree n in b. $\square$

3.1.2 Computing $|A_{n,b}|$ for small values of n

It is often helpful, when trying to visualise or understand a series of mathematical objects, to calculate a handful of examples and inspect them side-by-side. In the case of enumerating our alternating code this was achieved through employing MatLab to calculate $|A_{n,b}|$ for successive n (for the actual MatLab code see Appendix A). The program's output is summarised in Table 3.1.

At the heart of Theorem 3.1.5 is the observation that for $n > 1$ the size of the alternating code $A_{n,b}$ is $P_{n,0}(b, b)$. The polynomials $P_{n,i}(b, c)$ are defined recursively and so a recursive function would be a sensible way to calculate them. A perfectly valid though somewhat naïve approach would be to define the polynomials $P_{n,n}$ for each value of n and apply the recursive step to each of these until the $P_{n,0}$ are obtained. Fortunately, the following lemma helps us cut down on the computation and storage required by our naïve method.

Lemma 3.1.6. *Let c be some letter in our alphabet $[b]$, n a positive integer and i an integer with $n \geq i$. The polynomials $P_{n,i}(b, c)$, introduced in Definition* 3.1.3*, satisfy the formula $P_{n,i}(b, c) = P_{n+2,i+2}(b, c)$.*

Proof. The recursive step in the definition of the polynomial $P_{n,i}(b, c)$ depends solely on the parity of i and so it follows that the $n - i$ recursions to calculate $P_{n,i}(b, c)$ from $P_{n,n}(b, c)$ are identical the $n - i$ recursions to calculate $P_{n+2,i+2}(b, c)$ from $P_{n+2,n+2}(b, c)$. As $P_{n,n}(b, c)$ and $P_{n+2,n+2}(b, c)$ are both defined to be 1, we can see that indeed $P_{n,i}(b, c) = P_{n+2,i+2}(b, c)$. □

As previously discussed, the polynomial $P_{n,i}(b, c)$ counts the number of tails of length $n - i + 1$ of up/down words of length n where each tail starts with the letter c. If i is odd then the tails will be up/down, otherwise they will be down/up. The number of such tails only depends on the tail length, the first letter c and whether the tails we are counting are up/down or down/up. It is not affected therefore if i is any larger or smaller as long as the parity of i is preserved and this is what Lemma 3.1.6 is telling us.

Let us suppose that n is even and that the polynomial $P_{n-2,0}(b, c)$ has been calculated. In order to obtain $P_{n,0}(b, c)$ we need only notice:

$$
\begin{aligned}
P_{n,0}(b, c) &= \sum_{w_1=1}^{c-1} P_{n,1}(b, w_1) \\
&= \sum_{w_1=1}^{c-1} \sum_{w_2=w_1}^{b} P_{n,2}(b, w_2) \\
&= \sum_{w_1=1}^{c-1} \sum_{w_2=w_1}^{b} P_{n-2,0}(b, w_2).
\end{aligned}
$$

Table 3.1: The coefficients of the polynomials $P_{n,0}(b,b)$ and their relevant codes $A_{n,b}$

Code	1	b	b^2	b^3	b^4	b^5	b^6	b^7	b^8	b^9	b^{10}
$A_{1,b}$	-1	1									
$A_{2,b}$	0	$\frac{-1}{2}$	$\frac{1}{2}$								
$A_{3,b}$	0	$\frac{1}{6}$	$\frac{-1}{2}$	$\frac{1}{3}$							
$A_{4,b}$	0	$\frac{-1}{12}$	$\frac{7}{24}$	$\frac{-5}{12}$	$\frac{5}{24}$						
$A_{5,b}$	0	$\frac{1}{30}$	$\frac{-1}{6}$	$\frac{1}{3}$	$\frac{-1}{3}$	$\frac{2}{15}$					
$A_{6,b}$	0	$\frac{-1}{60}$	$\frac{4}{45}$	$\frac{-11}{48}$	$\frac{47}{144}$	$\frac{-61}{240}$	$\frac{61}{720}$				
$A_{7,b}$	0	$\frac{1}{140}$	$\frac{-17}{360}$	$\frac{13}{90}$	$\frac{-19}{72}$	$\frac{53}{180}$	$\frac{-17}{90}$	$\frac{17}{315}$			
$A_{8,b}$	0	$\frac{-1}{280}$	$\frac{247}{10080}$	$\frac{-25}{288}$	$\frac{73}{384}$	$\frac{-49}{180}$	$\frac{241}{960}$	$\frac{-277}{2016}$	$\frac{277}{8064}$		
$A_{9,b}$	0	$\frac{1}{630}$	$\frac{-4}{315}$	$\frac{571}{11340}$	$\frac{-23}{180}$	$\frac{119}{540}$	$\frac{-47}{180}$	$\frac{389}{1890}$	$\frac{-31}{315}$	$\frac{62}{2835}$	
$A_{10,b}$	0	$\frac{-1}{1260}$	$\frac{41}{6300}$	$\frac{-5183}{181440}$	$\frac{14821}{181440}$	$\frac{-5653}{34560}$	$\frac{40453}{172800}$	$\frac{-28723}{120960}$	$\frac{19811}{120960}$	$\frac{-50521}{725760}$	$\frac{50521}{3628800}$

The MatLab code in Appendix A uses this formula to compute $P_{n,0}(b,b)$ efficiently. The results of which are displayed in Table 3.1, where the nth row gives the coefficients of $P_{n,0}(b,b)$ as a polynomial in b next to the alternating code of length n. We mentioned earlier, in the proof of Theorem 3.1.5, that for a word w in $A_{n,b}$, the first letter w_1, could not take the value b as $w_1 < w_2$ and $w_2 \leq b$. This is of course not true if n takes the value 1, as there is no second letter to impinge upon w_1 taking the value b and it should be quite clear that $A_{1,b}$ is exactly $[b]$ and therefore $|A_{1,b}|$ is b. This is the only circumstance where $|A_{n,b}|$ is not $P_{n,0}(b,b)$.

The leading coefficient in the polynomials depicted in Table 3.1 are the most significant when b is large. The denominators of these (fractional) coefficients suggest that they could have been $n!$ before cancellation and they are certainly consistent with that idea. This observation motivated us to investigate the integer sequence obtained by multiplying the nth leading coefficient by $n!$ and then search for it in "The On-Line Encyclopedia of Integer Sequences" [Slo]. Fortunately, the sequence was there and it was under the definition of, amongst others, the number of up/down permutations on n letters. This finding is largely responsible for motivating the rest of this chapter.

3.2 Using the up/down numbers to calculate the three leading coefficients of the alternating code

Now that we know the size of the alternating code is a polynomial we shall prove its connection with the number of up/down permutations. We shall then investigate the consequences of this connection with an aim to actually finding the first few coefficients of highest degree in the polynomials $|A_{n,b}|$.

3.2.1 Up/down permutations and up/down numbers

Definition 3.2.1. *Let n be a positive integer. We define the set of up/down permutations, U_n, as*

$$\{w \in A_{n,n} : \textit{all } w_i \textit{ are distinct}\}$$

The *up/down numbers*, u_n, are precisely the number of up/down permutations, so $u_n := |U_n|$. The use of the word *permutation* here comes from the requirement that each word $w \in U_n$ has n distinct letters, i.e. that w is a permutation of $[n]$. We are certainly interested in such up/down permutations, but we shall also need to consider words that are up/down *"near permutations"* — where all but a few letters of w are distinct. We shall therefore require a way of counting the number of distinct letters that make up a word.

Definition 3.2.2. *Let w be a word in $[b]^n$, then define the set S_w as:*

$$\{w_1, w_2, w_3, \ldots, w_n\}.$$

In order to qualify the term "near permutations", we must also introduce the idea of frequencies of letters in words.

Definition 3.2.3. *Let w be a word of length n and suppose that $s \in S_w$, so s is a letter of w. We define the w-*frequency *of s, $Freq_w(s)$, as*

$$|\{i \in [n] : w_i = s\}|$$

We are now finally in a position to give a tangible idea of what a "near permutation" is.

Definition 3.2.4. *Let n be a positive integer. Let the set of up/down words where $n-2$ letters occur once and one letter occurs twice be $U_n^{[2]}$, so*

$$U_n^{[2]} := \{w \in A_{n,n-1} : |S_w| = n-1\}.$$

The set of up/down words where $n-4$ letters occur once and two letters occur twice each is denoted $U_n^{[2,2]}$, so

$$U_n^{[2,2]} := \{w \in A_{n,n-2} : |S_w| = n-2; \forall w_i \, Freq_w(w_i) \leq 2\}.$$

Finally, the set of up/down words where $n-3$ letters occur once and one letter occurs thrice is denoted $U_n^{[3]}$, so

$$U_n^{[3]} := \{w \in A_{n,n-2} : |S_w| = n-2; \exists i \, Freq_w(w_i) = 3\}.$$

3.2.2 Collapsing the alternating code onto an up/down permutation

The mapping we are about to define is the key to establishing the connection between the leading coefficients observed earlier in this chapter in Table 3.1 and the up/down numbers. Let us loosely define the *shape* of a word to be what we would get if we were to "plot" the word (with w_i against i) and then remove the axes so that we ignore any sense of scale. In effect, we can only see that points are "higher", "lower" or the same height as others and not by how much. If we then take the distance between vertically adjacent points to be 1 and then also take the height of the lowest point to be 1, we have encapsulated the essence of Definition 3.2.5.

The elegance of the mapping lies in that up/down words in the alternating code that share the same *shape* can be thought of as coming from the element with that same shape in one of the up/down "near permutation" sets (e.g. $U_n^{[2]}$). Furthermore, the frequency of these up/down words in the alternating code is the first thing one is taught when one is learning combinatorics — ${}^nC_r$.

Definition 3.2.5. *Let* $\Phi : [b]^n \to [n]^n$ *be defined as follows. Let* $m := |S_w|$ *so that we can order the elements of* S_w*, smallest to largest, and label them* $s_1, s_2, s_3, \ldots, s_m$*, in other words* $s_i < s_{i+1}$*. Now define* $\Phi(w) := v$ *where* $v_i = j \iff w_i = s_j$*.*

With Φ defined in this way we have, position-wise, the smallest letter of w being mapped to a 1, the second smallest to a 2 etc. and in the case of there being multiple joint ith smallest letters, they are all mapped to i. For example, the word 35263 would be mapped to 23142.

Lemma 3.2.6. *The following disjoint union gives us* $A_{n,b}$*:*

$$\Phi^{-1}(U_n) \cup \Phi^{-1}(U_n^{[2]}) \cup \Phi^{-1}(U_n^{[3]}) \cup \Phi^{-1}(U_n^{[2,2]}) \cup \{w \in A_{n,b} : |S_w| < n-2\}$$

Proof. It should be noted that Φ preserves up/down-ness, so the images of the words in $A_{n,b}$ are up/down and the pre-image of any up/down word in $[n]^n$ is also up/down.

Suppose w is a word in the above union. If w is a member of the set on the far right then $w \in A_{n,b}$ so suppose this is not the case. The image $\Phi(w)$ is then a member of one of U_n,$U_n^{[2]}$,$U_n^{[3]}$ or $U_n^{[2,2]}$ and therefore is up/down. So w is up/down and, because $w \in [b]^n$ we have $w \in A_{n,b}$.

Now let $w \in A_{n,b}$. If $|S_w| = n$, then clearly $w \in \Phi^{-1}(U_n)$. Similarly if $|S_w| = n-1$ then $w \in \Phi^{-1}(U_n^{[2]})$. When $|S_w| = n-2$ there are two possibilities: either a letter in w occurs in w three times or two different letters in w occur twice each. Either way, w lies in $\Phi^{-1}(U_n^{[3]})$ or $\Phi^{-1}(U_n^{[2,2]})$ respectively. □

Lemma 3.2.7. *For* $w \in A_{n,b}$*, if we are given* S_w *and* $\Phi(w)$ *then we can determine* w*.*

Proof. Let $u = \Phi(w)$ and in order of size, smallest first, let $s_1, s_2, \ldots, s_{|S_w|}$ be the elements of $|S_w|$. The image u is defined in terms of w by $u_i = j \iff w_i = s_j$ for $i = 1, 2, \ldots, n$, so when u and S_w are known we can find w. □

Lemma 3.2.8. *For any up/down word* $w \in A_{n,b}$ *the size of the pre-image* $\Phi^{-1}(\Phi(w))$ *is* $\binom{b}{|S_w|}$*.*

Proof. Take any word $w \in A_{n,b}$ and let $u = \Phi(w)$. By Lemma 3.2.7, we can determine w given only u and S_w. Suppose we are, instead of necessarily S_w, given some set $S \subset [b]$ such that $|S| = |S_w|$. The word we would determine, v say, is still an up/down word and thus in $A_{n,b}$; moreover for each such set S we necessarily generate a new word v, in $A_{n,b}$, such that $\Phi(v) = u$. Finally, because each set S lets us generate a unique v, we can see that $\Phi^{-1}(u)$ contains as many words as there are sets $S \subset [b]$. There are of course $\binom{b}{|S_w|}$ such sets available to us and so this amount also enumerates the pre-image. □

Lemma 3.2.9. *The number of words w in $A_{n,b}$ with $|S_w| \geq n-2$ is*

$$\binom{b}{n}|U_n| + \binom{b}{n-1}|U_n^{[2]}| + \binom{b}{n-2}\left(|U_n^{[2,2]}| + |U_n^{[3]}|\right).$$

Proof. Let us consider a word w in $A_{n,b}$ with $|S_w| \geq n-2$. From Lemma 3.2.6 we can see that the image $u = \Phi(w)$ will belong to one of U_n, $U_n^{[2]}$, $U_n^{[2,2]}$ or $U_n^{[3]}$ and as $|S_w| = |S_u|$, the word u has pre-image size $\binom{n}{|S_u|}$. Of course, each element of the sets U_n, $U_n^{[2]}$, $U_n^{[2,2]}$ and $U_n^{[3]}$ has a distinct disjoint pre-image of that size and so adding the relevant products yields the desired result. □

Let X be the set $\{w \in A_{n,b} : |S_w| < n-2\}$ so that we may enumerate the rest of $A_{n,b}$. If $x \in X$ then x has at most $n-3$ distinct letters. Let $i_1 > 1$ be the smallest integer such that there exists an integer $j_1 < i_1$ with $x_{j_1} = x_{i_1}$. Define x' to be the word of length $n-1$ obtained by removing from x the letter at position i_1. The pair (i_1, j_1) then encodes the first occurrence of a letter appearing more than once in x in such a way that we may use it to recover x from x'.

In the same way, determine the pairs (i_2, j_2) and (i_3, j_3) from x' and x'' respectively, so as to leave us with x''' — a word of length $n-3$.

For example, consider the word $y = 132312$ then $S_y = \{1, 2, 3\}$. Here $n = 6$ and so, since y is up/down, we have $y \in X$. The first letter to appear a second time in y is 3 and so $j_1 = 2$ and $i_1 = 4$ and $y' = 13212$. From the pair $(4, 2)$ and the word 13212 we are able to reconstruct y by inserting a copy of the letter at position 2 into position 4.

In the word y' the first letter to appear a second time is 1 and so $(i_2, j_2) = (4, 1)$ and $y'' = 1322$. Finally, we have that $(i_3, j_3) = (4, 3)$ and $y''' = 132$.

Each of our pairs of integers belongs to $[n] \times [n]$, so this process defines a mapping, say $\Phi : X \to [b]^{n-3} \times [n]^6$. Since we may use the pairs to eventually recover x from x''', Φ must be injective. The range of Φ is easily enumerated as $n^6 b^{n-3}$; this together with the injectivity of Φ shows that $|X| \leq n^6 b^{n-3}$. Hence $|X| = O(b^{n-3})$. This now proves the following.

Theorem 3.2.10. *For fixed n, the size of the alternating code is*

$$\binom{b}{n}|U_n| + \binom{b}{n-1}|U_n^{[2]}| + \binom{b}{n-2}\left(|U_n^{[2,2]}| + |U_n^{[3]}|\right) + O(b^{n-3}).$$

Lemma 3.2.11. *For fixed n, the expression $n!\binom{b}{n}$ can be written as*

$$b^n - \frac{n(n-1)}{2}b^{n-1} + \frac{n(n-1)(n-2)(3n-1)}{24}b^{n-2} + O(b^{n-3}).$$

Proof. Express $n!\binom{b}{n}$ as $b(b-1)(b-2)\dots(b-(n-1))$. The b^n term is obtained by taking a b from each bracket. If then from the ith bracket we take i and a b from all the others we get $\sum_{i=0}^{n-1} -ib^{n-1}$ which is $-\frac{n(n-1)}{2}b^{n-1}$. Finally, we wish to count the ways to take $-i$ and $-j$ from the ith and jth brackets with $i < j$, in order to find the b^{n-2} coefficient. This is $\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1} ij$, but may be more easily evaluated by summing ij with i and j independently from 0 to $n-1$ and then discounting the products where $i \geq j$.

We have $\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} ij = \left(\sum_{k=0}^{n-1} k\right)^2 = \left(\frac{n(n-1)}{2}\right)^2$. Firstly, we shall discount the cases where $i = j$, a contribution of $\sum_{i=0}^{n-1} i^2$, evaluating to $\frac{1}{6}n(n-1)(2n-1)$. Secondly, we notice that for each contribution of ij where $i < j$ to the sum, there is one of ij with $i > j$; we need only halve the remainder. So, the b^{n-2} coefficient is $\frac{1}{2}\left(\left(\frac{n(n-1)}{2}\right)^2 - \frac{1}{6}n(n-1)(2n-1)\right)$, which simplifies to the desired $\frac{n(n-1)(n-2)(3n-1)}{24}$.

□

Corollary 3.2.12. *For fixed n, the size of the alternating code is as follows:*

$$\begin{aligned}|A_{n,b}| =& b^n\frac{|U_n|}{n!}\\ &+ b^{n-1}\left(\frac{|U_n^{[2]}|}{(n-1)!} - \frac{|U_n|}{2(n-2)!}\right)\\ &+ b^{n-2}\left(\frac{|U_n^{[2,2]}| + |U_n^{[3]}|}{(n-2)!} - \frac{|U_n^{[2]}|}{2(n-3)!}\right)\\ &+ b^{n-2}\left(\frac{|U_n|(3n-1)}{24(n-3)!}\right)\\ &+ O(b^{n-3}).\end{aligned}$$

Proof. Upon expanding $\binom{b}{n}$, $\binom{b}{n-1}$ and $\binom{b}{n-2}$, we obtain

$$\begin{aligned}\binom{b}{n} =& \frac{1}{n!}\left(b^n - \frac{n(n-1)}{2}b^{n-1} + \frac{n(n-1)(n-2)(3n-1)}{24}b^{n-2}\right)\\ &+ O(b^{n-3}),\\ \binom{b}{n-1} =& \frac{1}{(n-1)!}\left(b^{n-1} - \frac{(n-1)(n-2)}{2}b^{n-2}\right) + O(b^{n-3}) \text{ and}\\ \binom{b}{n-2} =& \frac{1}{(n-2)!}b^{n-2} + O(b^{n-3})\end{aligned}$$

respectively. Bringing this all together, with the result of Theorem 3.2.10 gives the required result. □

3.2.3 Computing up/down numbers and related sequences

Recall Definition 1.2.1 where we said that for the alternating code $A_{n,b}$:

$$|A_{n,b}| = \alpha_{n,n}b^n + \alpha_{n,n-1}b^{n-1} + \alpha_{n,n-2}b^{n-2} + O(b^{n-3}).$$

Table 3.2: Calculations of $|U_n|$, $|U_n^{[2]}|$, $|U_n^{[2,2]}| + |U_n^{[3]}|$ and how their magnitudes compare to each other

n	$\lvert U_n\rvert$	$\lvert U_n^{[2]}\rvert$	$\lvert U_n^{[2,2]}\rvert + \lvert U_n^{[3]}\rvert$	$\frac{\lvert U_n^{[2]}\rvert}{\lvert U_n\rvert}$	$\frac{\lvert U_n^{[2,2]}\rvert+\lvert U_n^{[3]}\rvert}{\lvert U_n\rvert} - \frac{3n^2-17n+25}{24}$
3	2	1	0	1/2	$\frac{-1}{24}$
4	5	5	1	1	$\frac{-2}{240}$
5	16	24	10	3/2	0
6	61	122	79	2	3.4153e(-3)
7	272	680	602	5/2	4.9020e(-3)
8	1385	4155	4682	3	5.5054e(-3)
9	7936	27776	38072	7/2	5.7124e(-3)
10	50521	202084	326570	4	5.7113e(-3)
20	3.7037e(14)	3.3333e(15)	1.3659e(16)	9	4.0582e(-3)
40	1.4851e(40)	2.8217e(41)	2.5650e(42)	19	2.3066e(-3)
60	1.8109e(70)	5.2516e(71)	7.3983e(72)	29	1.5973e(-3)
80	1.8623e(103)	7.2629e(104)	1.3862e(106)	39	1.2200e(-3)
100	2.9035e(138)	1.4227e(140)	3.4268e(141)	49	9.8655e(-4)
120	2.4884e(175)	1.4682e(177)	4.2702e(178)	59	8.2794e(-4)
140	5.9874e(213)	4.1313e(215)	1.4082e(217)	69	7.1321e(-4)
160	2.5072e(253)	1.9807e(255)	7.7415e(256)	79	6.2639e(-4)
180	1.2773e(294)	1.1368e(296)	5.0117e(297)	89	5.5839e(-4)
200	5.9955e(335)	5.9355e(337)	2.9134e(339)	99	5.0371e(-4)

For each alternating code $A_{n,b}$, Corollary 3.2.12 gives the coefficients $\alpha_{n,n}$, $\alpha_{n,n-1}$ and $\alpha_{n,n-2}$ in terms of the up/down numbers, $|U_n|$, and the related sequences $|U_n^{[2]}|$, $|U_n^{[2,2]}|$ and $|U_n^{[3]}|$. We can easily solve these simultaneous equations to give us the sequences $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}|+|U_n^{[3]}|$. The simultaneous equations do not however give us enough information to solve individually for the sequences $|U_n^{[2,2]}|$ and $|U_n^{[3]}|$.

We generated these coefficients computationally and substituted the results into the solutions for the simultaneous equations to compute the sequences $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}| + |U_n^{[3]}|$ for n up to 200. These are given in Table 3.2 along with two other columns, present to help us better understand how the sequences relate to each other. The code used for the computations was written in *Mathematica* and is given in Appendix D. A more detailed discussion of the code and the optimisations it exploits is given in 7.4.

The $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}|+|U_n^{[3]}|$ all grow faster than exponential, the dominant value being the latter. Probably the most striking column of Table 3.2 is however the column for $\frac{|U_n^{[2]}|}{|U_n|}$. It is telling us that $|U_n^{[2]}| = \frac{n-2}{2}|U_n|$. Although not depicted, for obvious reasons, the $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}| + |U_n^{[3]}|$ were all calculated to infinite precision (symbolically) and so the fifth column does not represent any process that, say, merely tends to $\frac{n-2}{2}$. This observed result is proved later in Corollary 3.2.20.

On inspection of the sequence $\frac{|U_n^{[2,2]}|+|U_n^{[3]}|}{|U_n|}$ it was clear that it resembles a parabola. Using the 200 or so terms calculated for it a least square fit was performed on them to the terms n^2, n and 1 and to several decimal places the values $\frac{3}{24}$, $\frac{-17}{24}$ and $\frac{25}{24}$ were returned. Upon taking this quadratic away from the sequence, we are left with the last column of Table 3.2. This column tells us that, although the aforementioned parabola is dominant in the quotient $\frac{|U_n^{[2,2]}|+|U_n^{[3]}|}{|U_n|}$, there is an additional factor to be considered. However, upon inspection of this factor, it appears to gradually decay.

Conjecture 3.2.13. *As $n \to \infty$*

$$\left| \frac{|U_n^{[2,2]}| + |U_n^{[3]}|}{|U_n|} - \frac{3n^2 - 17n + 25}{24} \right| \to 0.$$

3.2.4 Proving an observed result

Recall that in Table 3.2 we observed rather strong evidence for the quotient $\frac{|U_n^{[2]}|}{|U_n|}$ taking the value $\frac{n-2}{2}$. In this section we develop some terminology and tools to prove this observation.

Definition 3.2.14. *Let w be some word and i and j some indices. Say that w_i and w_j are* neighbours *if $|i - j| = 1$.*

Definition 3.2.15. *Let x and n be positive integers with $1 < x \leq n$ and w a word in U_n. Suppose that for some integer i we have $w_i = x$. Define the pair (x, w) to be an x-*adjacency *if $w_{i-1} = x - 1$ or $w_{i+1} = x - 1$, in other words x and $x - 1$ are neighbours in the word w. We shall also refer to an x-adjacency as an* adjacency.

Definition 3.2.16. *Define Adj_n to be the set of all adjacencies associated with U_n and $Adj_n[x]$ to be the set of all words $w \in U_n$ with an x-adjacency. So*

$$\begin{aligned} Adj_n &:= \{(x, w) \in \{2, 3, \ldots, n\} \times U_n : (x, w) \text{ is an } x\text{-adjacency}\} \text{ and} \\ Adj_n[x] &:= \{w \in U_n : (x, w) \text{ is an } x\text{-adjacency}\}. \end{aligned}$$

Recall Definition 3.2.2 where, for some word w of length n, we defined

$$S_w := \{w_1, w_2, w_3, \ldots, w_n\}.$$

In Definition 3.2.3 we defined, for some letter s and some word w, the integer $Freq_w(s)$ to be the frequency that the letter s occurs in the word w. Finally, recall Definition 3.2.4:

$$U_n^{[2]} := \{w \in A_{n,n-1} : |S_w| = n - 1\}.$$

Definition 3.2.17. *Let x and n be positive integers such that $x \leq n - 1$. Define then $U_n^{[2]}[x]$ to be the set of words w in $U_n^{[2]}$ such that $Freq_w(x) = 2$. So*

$$U_n^{[2]}[x] := \left\{w \in U_n^{[2]} : Freq_w(x) = 2\right\}.$$

Lemma 3.2.18. *Let x and n be positive integers such that $1 < x \leq n$. Then the set $U_n \backslash Adj_n[x]$ has twice as many elements as $U_n^{[2]}[x-1]$. So*

$$|U_n^{[2]}[x-1] \times \{0, 1\}| = |U_n| - |Adj_n[x]|.$$

Equivalently

$$|U_n^{[2]}[x-1]| = \frac{|U_n| - |Adj_n[x]|}{2}.$$

Proof. We shall prove this result by defining a bijection between $U_n \backslash Adj_n[x]$ and $U_n^{[2]}[x-1] \times \{0, 1\}$. Let Φ be a mapping so that

$$\Phi : U_n \backslash Adj_n[x] \to U_n^{[2]}[x-1] \times \{0, 1\}.$$

Let w be any word in $U_n \backslash Adj_n[x]$ and define $\Phi(w) := (v, a)$ where, for any index i

$$v_i = \begin{cases} w_i & \text{if } w_i < x, \\ w_i - 1 & \text{if } w_i \geq x \end{cases}$$

and $a = 0$ if $x - 1$ comes before x in w, 1 otherwise. Firstly, we must show that Φ is well defined and to do this we need to show that our v is up/down, $v \in [n-1]^n$, $|S_v| = n - 1$ and $Freq_v(x) = 2$.

If for any index i less than n we can show that $w_{i+1} - w_i$ has the same sign as $v_{i+1} - v_i$ then we have proved Φ preserves up/downness. Suppose that i is an odd index, then $w_{i+1} - w_i$ is positive. The only way $v_{i+1} - v_i$ can be less than $w_{i+1} - w_i$ is if $w_i < x$ and $w_{i+1} \geq x$. This only causes $v_{i+1} - v_i$ not to be positive if $w_i = x - 1$ and $w_{i+1} = x$, but this case is an x-adjacency and hence not in our domain. The even case is proved in an identical way therefore v is up/down.

Any letter w_i is between 1 and n inclusively. Since $1 < x \leq n$, w_i will not get mapped to anything less than 1 and the most v_i can be is $n - 1$. Therefore v is a member of $[n-1]^n$.

The set of letters of w is S_w and this is always $[n]$. Since the letters at least as large as x will be mapped to themselves less one, the only case where two letters are mapped to the same letter will be $x - 1 \mapsto x - 1$ and $x \mapsto x - 1$. Therefore $|S_v| = n - 1$ and $Freq_v(x-1) = 2$. Hence Φ is well defined.

Secondly, we show that Φ is injective. So suppose then that we have $\Phi(w) = (v, a) = \Phi(w')$. The letters in v less than $x - 1$ were unchanged by the application of Φ on w and w' so w and w' must agree in those positions. Similarly, the letters larger than $x - 1$ in v were all greater by one in w and w', so again w and w' agree in those positions. The two positions in v containing the letter $x - 1$ came from x and $x - 1$ in the words w and w' and the value of a fixes this order to be the same for both w and w'. So $w = w'$ and Φ is injective.

Finally, we prove that Φ is surjective. Let (v, a) be some element in the codomain of Φ. We can generate a w such that $w_i = v_i$ where $v_i < x - 1$, $w_i = v_i + 1$ where $v_i > x - 1$. We then use a to determine which of the positions in v containing the letter $x - 1$ to put $x - 1$ and which to put x: the first occurrence is $x - 1$ if $a = 0$ and the second otherwise.

Again, let i be an positive odd integer less than n. Then $v_{i+1} - v_i$ is positive and we need to show $w_{i+1} - w_i$ is also positive. However, the difference between w_{i+1} and w_i can only be more than that of v_{i+1} and v_i, so $w_{i+1} - w_i \geq v_{i+1} - v_i > 0$. Since there is an identical argument for an even index i less than n, w is up/down.

If $x - 1 < n$ then the letters 1 to $x - 2$ in v remain as they are in w and the letters x to $n - 1$ are increased by one. One of the occurrences of $x - 1$ remains and the other is mapped to x. Therefore w is a permutation of $[n]$ and hence a member of U_n. In the case where $x - 1 = n$, each of the letters 1 to $n - 2$ in v remain as they are in w and one of the occurrences of $n - 1$ is mapped to n and again $w \in U_n$.

Since v is up/down, we do not have the two occurrences of $x - 1$ next to each other. Therefore (x, w) is not an x-adjacency. This w is then a member of $U_n \backslash Adj_n[x]$ and Φ is surjective.

Hence Φ is bijective and the result follows. $\square$

Lemma 3.2.19. *Let n be an integer larger than 1. Then*

$$|Adj_n| = |U_n|.$$

Proof. Let Ψ be a mapping from Adj_n to U_n. So

$$\Psi : Adj_n \to U_n.$$

Let (x, w) be any adjacency in Adj_n and define $\Psi((x, w)) := v$ where, for any index i

$$v_i = \begin{cases} w_i & \text{if } w_i < x, \\ n & \text{if } w_i = x, \\ w_i - 1 & \text{if } w_i > x. \end{cases}$$

The word w is a member of U_n, so for $x = n$ we have that $v = w$ and so then v is also in U_n. Let us assume then that $x < n$. All the letters of w are distinct and Ψ cyclically permutes the letters $x, x+1, x+2, \ldots, n$ so the letters of v are also distinct. In order to show that $v \in U_n$ we therefore only need show it to be up/down.

If we can show that, for any even index i, v_i is larger than its neighbours, it follows that, for any odd index j, v_j is less than its neighbours. Also, since (x, w) is an x-adjacency, x is next to an $x - 1$ so if $w_k = x$ then k must be even because w is up/down.

So, let i be an even index, then w_i is greater than w_{i-1} and w_{i+1}, because w is up/down. If $w_i < x$ then $v_{i-1} = w_{i-1}$, $v_i = w_i$ and $v_{i+1} = w_{i+1}$ so v_i is larger than its neighbours. If $w_i = x$ then $v_i = n$ and so larger than its neighbours. Suppose then that $w_i > x$ and that the letter y is one of its neighbours. Then $v_i = w_i - 1$. Since x only occurs in w at a position with an even index, we have that $y \neq x$. If $y < x$ then y remains unchanged in v, but the smallest v_i can be is x, so $v_i > y$. If $y > x$ then y is replaced by $y - 1$ and since $y < w_i$ we have that $y - 1 < w_i - 1 = v_i$. Therefore v_i is larger than its neighbours and v is up/down. Hence Ψ is well defined.

To prove injectivity, suppose that for some adjacencies (x, w) and (x', w') we have that $\Psi((x, w)) = v = \Psi((x', w'))$. Suppose that $v_i = n$. Since all other letters in the word v are less than v_i, we have that v_i is larger than its neighbours (or neighbour if it is the last letter, in which case $i = n$). Therefore i is even. Let y be the larger of these neighbours v_{i-1} and v_{i+1} (or just v_{i-1} if $i = n$).

When Ψ maps some z-adjacency to an up/down permutation, the only letter that ends up being n is z and the neighbours to z, which are smaller than z (the larger being $z - 1$), remain unchanged. It follows that both adjacencies must be $(y + 1)$-adjacencies and so $x = y + 1 = x'$. Indeed $w_i = y + 1 = w'_i$.

If, for some index j, $v_j < y + 1$ then $w_j = v_j = w'_j$, since Ψ left all letters less than $y + 1$ unchanged. Lastly, all letters v_j that are at least $y + 1$ and at most $n - 1$ must have been $v_j + 1$ before Ψ was applied hence $w_j = v_j + 1 = w'_j$. All letters have now been accounted for and so $(x, w) = (x', w')$ and Ψ is injective.

For surjectivity, take any word v in U_n. Let i be the index such that $v_i = n$. Let

y be the larger (or only) neighbour of v_i. Construct the word w so that

$$w_i = \begin{cases} v_i & \text{if } v_i < y+1, \\ y+1 & \text{if } v_i = n, \\ v_i + 1 & \text{if } y+1 \leq v_i < n. \end{cases}$$

Since w is constructed so that when Ψ is applied to it it yields v, it suffices to show that w is a member of U_n.

Again, let i be an even index. If $w_i < y+1$ then $w_i = v_i$. Since i is even and v is up/down, any neighbour of v_i will be less than v_i and so are unchanged in w. Thus w_i is larger than its neighbours. If $w_i = y+1$ then $v_i = n$ and the largest neighbour of n in v is y. Since all neighbours of n in v are at most y, they remain unchanged in w and so w_i is larger than its neighbours. If $w_i > y+1$ then $v_i = w_i - 1$ and any neighbour z of v_i in v is such that $v_i - z > 0$. Since $w_i = v_i + 1$ and a letter is increased by at most one when generating w from v, it follows that w_i is larger than the neighbour corresponding to z in v. Since z is a general neighbour, w_i is larger than all of its neighbours in w and hence w is up/down.

Our construct for w cyclically permutes the letters $y+1, y+2, y+3, \ldots, n$. Since all the letters of v are distinct, so are the ones of w. Therefore $w \in U_n$. This proves surjectivity.

Therefore Ψ is bijective and the result follows. $\square$

Corollary 3.2.20. *For $n > 2$ we have that*

$$|U_n^{[2]}| = |U_n| \frac{n-2}{2}.$$

Proof. Let x be an integer such that $2 \leq x \leq n$. If $w \in Adj_n[x]$ then by definition $(x, w) \in Adj_n$. Also, if $(y, v) \in Adj_n$ then by definition $v \in Adj_n[y]$. Therefore

$$|Adj_n| = \sum_{x=2}^{n} |Adj_n[x]|.$$

Indeed, we can also write

$$Adj_n = \bigcup_{x \in \{2,3,4,\ldots,n\}} \{x\} \times Adj_n[x].$$

Recall also that in Lemma 3.2.19 we showed that

$$|Adj_n| = |U_n|.$$

If $w \in U_n^{[2]}$ then for some letter $x \in [n-1]$ we have that $Freq_w(x) = 2$ and so

$w \in U_n^{[2]}[x]$. By definition $U_n^{[2]}[x] \subset U_n^{[2]}$ and so

$$U_n^{[2]} = \bigcup_{x \in [n-1]} U_n^{[2]}[x].$$

Therefore, since this is a disjoint union,

$$\left|U_n^{[2]}\right| = \sum_{x=1}^{n-1} \left|U_n^{[2]}[x]\right|.$$

Applying Lemma 3.2.18 to this gives us

$$\begin{aligned}
\left|U_n^{[2]}\right| &= \sum_{x=1}^{n-1} \frac{|U_n| - |Adj_n[x+1]|}{2} \\
&= \frac{(n-1)\,|U_n| - \sum_{x=1}^{n-1} |Adj_n[x+1]|}{2} \\
&= \frac{(n-1)\,|U_n| - \sum_{x=2}^{n} |Adj_n[x]|}{2} \\
&= \frac{(n-1)\,|U_n| - |Adj_n|}{2} \\
&= \frac{(n-1)\,|U_n| - |U_n|}{2} \\
&= |U_n| \, \frac{n-2}{2}.
\end{aligned}$$

□

If Conjecture 3.2.13 is correct then, together with Corollary 3.2.20, we can give a more explicit and greatly simplified version of Corollary 3.2.12. For the remainder of this chapter let us assume that Conjecture 3.2.13 is true. We may therefore find a nonnegative function $\epsilon(n)$ such that $\epsilon(n) \to 0$ as $n \to \infty$ and

$$|U_n^{[2,2]}| + |U_n^{[3]}| = |U_n| \left(\frac{3n^2 - 17n + 25}{24} + \epsilon(n) \right).$$

Recall that in Corollary 3.2.12 we showed that

$$\begin{aligned}
|A_{n,b}| =& b^n \frac{|U_n|}{n!} \\
&+ b^{n-1} \left(\frac{|U_n^{[2]}|}{(n-1)!} - \frac{|U_n|}{2(n-2)!} \right) \\
&+ b^{n-2} \left(\frac{|U_n^{[2,2]}| + |U_n^{[3]}|}{(n-2)!} - \frac{|U_n^{[2]}|}{2(n-3)!} \right) \\
&+ b^{n-2} \left(\frac{|U_n|(3n-1)}{24(n-3)!} \right) \\
&+ O(b^{n-3})
\end{aligned}$$

and in Corollary 3.2.20 we showed that

$$|U_n^{[2]}| = |U_n|\frac{n-2}{2}.$$

Therefore

$$\begin{aligned}
|A_{n,b}| &= b^n \frac{|U_n|}{n!} \\
&\quad + b^{n-1}\left(\frac{|U_n|\frac{n-2}{2}}{(n-1)!} - \frac{|U_n|}{2(n-2)!}\right) \\
&\quad + b^{n-2}\left(\frac{|U_n|\left(\frac{3n^2-17n+25}{24} + \epsilon(n)\right)}{(n-2)!} - \frac{|U_n|\frac{n-2}{2}}{2(n-3)!}\right) \\
&\quad + b^{n-2}\left(\frac{|U_n|(3n-1)}{24(n-3)!}\right) \\
&\quad + O(b^{n-3}) \\
&= b^n \frac{|U_n|}{n!} \\
&\quad + b^{n-1}|U_n|\left(\frac{n-2}{2(n-1)!} - \frac{n-1}{2(n-1)!}\right) \\
&\quad + b^{n-2}|U_n|\left(\frac{3n^2-17n+25}{24(n-2)!} - \frac{6(n-2)^2}{24(n-2)!} + \frac{\epsilon(n)}{(n-2)!}\right) \\
&\quad + b^{n-2}\left(\frac{|U_n|(3n-1)(n-2)}{24(n-2)!}\right) \\
&\quad + O(b^{n-3}) \\
&= b^n \frac{|U_n|}{n!} - b^{n-1}\frac{|U_n|}{2(n-1)!} + b^{n-2}\frac{|U_n|\,(1+8\epsilon(n))}{8(n-2)!} + O(b^{n-3}).
\end{aligned}$$

Chapter 4

Pebody's weakly consecutive repeat-free codes

4.1 Large weakly consecutive repeat-free codes

4.1.1 Pebody's conjecture

Recall Definition 1.1.4 where, for a word $y \in [b]^{n-1}$, we defined the Pebody code $B_y \subset [b]^n$ to be the set of all words $w \in [b]^n$ such that for any index $i < n$, $w_i \leq w_{i+1}$ if i is odd, $w_i \geq w_{i+1}$ if i is even and if $w_i = w_{i+1}$ then $w_i = y_i = w_{i+1}$.

As we discussed in Chapter 1, Pebody conjectured in [Peb06] that for each n, there is some $y \in [b]^{n-1}$ such that B_y is a largest weakly consecutive repeat-free code. He then gives without proof the values of y that generate largest Pebody codes for $n \leq 5$.

Even for these first few values of n however, patterns in the choice of such y emerge. This chapter shows us why these patterns exist and describes properties of large and largest Pebody codes. For example, the first letter of y can always be exchanged for the letter b without decreasing the size of B_y.

Lemma 4.1.1. *Let B_y be any Pebody code of length n and let the word y' be $by_2y_3 \dots y_{n-1}$ then $|B_y| \leq |B_{y'}|$.*

Proof. Define the map $\Psi : B_y \to B_{y'}$ so that for any word w in B_y we have $\Psi(w) = w$ if $w_1 \neq w_2$ and $\Psi(w) = bbw_3w_4 \dots w_n$ otherwise. Observe that Ψ is well defined since $\Psi(B_y) \subseteq B_{y'}$. We shall show that Ψ is injective to complete the proof.

Let us suppose then that for some $v, w \in B_y$ that $\Psi(v) = \Psi(w)$. If we can then prove that $v = w$, we will have shown that Ψ is injective. For ease of notation, say $u = \Psi(v)$. If $u_1 \neq b$ then $v = u$ and $w = u$ hence $v = w$ as required.

Let us suppose then that $u_1 = b$. The code $B_{y'}$ is Pebody and so by definition we have that $u_1 \leq u_2$ hence u_2 is also b. If it were the case that $v_1 \neq v_2$ then, by definition of Ψ, we would have that $u_1 \neq u_2$ and so we must have that $v_1 = v_2$ and by the same reasoning also that $w_1 = w_2$. As B_y is a Pebody code, and the first two letters of v are the same, we must have that $v_1 = y_1 = v_2$ and also that $w_1 = y_1 = w_2$. The words v and w therefore agree in the first two letters and, as Ψ does not change any but the first two letters, v and w agree in the rest of the letters also. Hence $v = w$, Ψ is injective and so $|B_y| \leq |B_{y'}|$. □

4.1.2 A result on Pebody codes

This section proves a small result on Pebody codes, as an aside.

Definition 4.1.2. *Let w be a word of length n and define the* reverse *of w to be the word $w_n w_{n-1} w_{n-2} \dots w_1$ and let us denote it by w^R.*

Corollary 4.1.3. *Let B_y be any Pebody code of length n then $|B_y| \leq |B_{y'}|$ where y' is $y_1 y_2 y_3 \dots y_{n-2} b$ if n is odd and $y_1 y_2 y_3 \dots y_{n-2} 1$ if n is even.*

Proof. For the case where n is odd there is a natural (self-inverse) bijection between the codes B_y and B_{y^R} defined by reversing the words. By application of Lemma 4.1.1 to B_{y^R} and reflecting the changes in B_y we find that $|B_y| \leq |B_{y'}|$.

Let us suppose then that n is even. Reversing the words of B_y leaves us with a set that is not a Pebody code, but if in addition to reversing the words we then relabel the letters $1, 2, 3, \dots, b$ to $b, b-1, b-2, \dots, 1$ then our set is a Pebody code, B_x say, and we can once again apply Lemma 4.1.1.

Consider for example the word $w = 2311 \in B_{321}$. The reverse of this word, $w^R = 1132$ has $w_2^R < w_3^R$ and so is not a member of any Pebody code. If we now relabel $1, 2, 3$ to $3, 2, 1$ then we get 3312 which is in B_{321}.

To clarify this process we shall describe the word x in terms of y. Recall that the word y is of length $n-1$. Take then any index i of x, this will relate to an index $n-i$ of y because of the reversing step. Let c be a letter in $[b]$, the relabelling step may then be described as $c \mapsto b+1-c$. We can now write $x_i = b+1-y_{n-i}$. Let x' be $bx_2x_3 \dots x_{n-1}$ as the application of Lemma 4.1.1 changes x_1 to be b. This forces y_{n-1} to become 1.

Both the reversing and relabelling steps of this process are self-inverse (even commutative) and so reapplying them to the Pebody code $B_{x'}$ yields $B_{y'}$. This completes the proof since $|B_y| = |B_x|$, $|B_x| \leq |B_{x'}|$ and $|B_{x'}| = |B_{y'}|$. □

4.2 Maximising Pebody codes

Immediately below, we see that the alternating code is a subset of any Pebody code B_y. For fixed n, we then show that the number of words in $B_y \backslash A_{n,b}$ is $O(b^{n-2})$. This is far fewer than the number of words in the alternating code which, as Corollary 3.2.12 states, is $\frac{|U_n|}{n!} b^n + O(b^{n-1})$. Therefore the sizes of a Pebody code and the alternating code agree in the coefficients of the b^n and b^{n-1} terms. We go on to give a way of calculating the coefficient of the b^{n-2} term in $|B_y \backslash A_{n,b}|$ by extending the definition of the polynomial $P_{n,i}$, thus allowing us to estimate the size of Pebody codes. In particular, we note a special implication for maximal Pebody codes.

4.2.1 On Pebody codes and strongly consecutive repeat-free codes

Lemma 4.2.1. *Let n be a positive integer and y be a word of length $n-1$. The alternating code $A_{n,b}$ is a subset of the Pebody code B_y.*

Proof. The result follows since any word in $A_{n,b}$ is up/down and this is a stronger than the condition for membership of B_y. $\square$

Definition 4.2.2. *Let n be a positive integer, $w \in [b]^n$ and i an index of w such that $i < n$. If $w_i = w_{i+1}$ then we say that w has a* repeat *at position i.*

Any word w in a Pebody code B_y that is not also up/down must then have a repeat, at position i say, such that $w_i = y_i = w_{i+1}$. Since we have already seen results on the sizes of alternating codes, we shall explore the words in B_y which are not in $A_{n,b}$. These are the words which have at least one repeat and hence are not up/down.

Lemma 4.2.3. *Let B_y be a Pebody code. The number of words with at least two repeats is $O(b^{n-3})$ and the number of words with exactly one repeat is $O(b^{n-2})$.*

Proof. Suppose w is a word in B_y with repeats at positions i and j where $i < j$. From Definition 4.2.2 we can see that $w_i = y_i = w_{i+1}$ and $w_j = y_j = w_{j+1}$.

If $i + 1 = j$ then this forces w_i, w_{i+1} and w_{i+2} to all take the value y_i. There are at most b^{n-3} choices for the other letters in w and i may take any value from 1 to $n-2$. Therefore there are at most $(n-2)b^{n-3}$ possibilities for w in this case.

Suppose then that $i + 1 \neq j$. Since $i < j$ the repeats do not overlap. This determines four of the letters of w and so there are b^{n-4} choices for the remaining ones. The positions of our repeats may take the values from 1 to $n-1$. The value $\binom{n-1}{2}$ will give us the number of suitable choices for i and j once we discount from it the adjacent cases since $i + 1 \neq j$. For the pair (i, j) then, we must discount all cases $(1, 2), (2, 3), \ldots, (n-2, n-1)$. We now have a total of $\left(\binom{n-1}{2} - (n-2)\right) b^{n-4}$ choices for words in B_y with at least two nonadjacent repeats. In either case there are at most $O(b^{n-3})$ words with at least two repeats.

Suppose now that w is word in B_y with exactly one repeat. If the position of that repeat is i then $w_i = y_i = w_{i+1}$, leaving at most b^{n-2} choices for the other letters in w. There are $n-1$ possibilities for i and so there are at most $(n-1)b^{n-2}$ possibilities for w, which is $O(b^{n-2})$. $\square$

The purpose of Lemma 4.2.3 is to draw our attention more to words with exactly one repeat. If we can show that there are more than $O(b^{n-3})$ of these words then this would indicate that they make up the most significant portion of the non-up/down words in a Pebody code B_y for large enough alphabet size b. Perhaps then, this can help us say more about choices for the letters of y that generate large Pebody codes B_y.

Definition 4.2.4. *Let b, n and i be integers with $b > 0$, $n \geq i$. Recall the definition of $P_{n,i}(b, w_i)$ for $n > 0$ given in Definition* 3.1.3. *Recall also that in Lemma* 3.1.6 *we proved that $P_{n,i} = P_{n+2,i+2}$ for $n > 0$.*

Let us extend the definition of $P_{n,i}$ for $n > 0$ to the case where n can be any integer. As $P_{n,i}$ is already defined for $n > 0$, suppose that $n \leq 0$. Let $r = \left\lfloor \frac{n-1}{2} \right\rfloor$. Define $P_{n,i} = P_{n-2r,i-2r}$. Since $n - 2r$ is positive, $P_{n,i}$ is well defined.

Lemma 4.2.5. *Let n and i be any integers with $n \geq i$ then the equality $P_{n,i} = P_{n+2,i+2}$ proved for positive n in Lemma* 3.1.6 *holds now for any integer n.*

Proof. Note that the result already holds for positive n. Suppose then that $n \leq 0$. Again, let $r = \left\lfloor \frac{n-1}{2} \right\rfloor$. By definition $P_{n,i} = P_{n-2r,i-2r}$.

If $n < -1$ then $n + 2 \leq 0$ so $P_{n+2,i+2}$ is defined as $P_{(n+2)-2(r+1),(i+2)-2(r+1)} = P_{n-2r,i-2r}$, hence $P_{n,i} = P_{n+2,i+2}$.

If $n = -1$ or $n = 0$ then $r = -1$ so $P_{n,i} = P_{n-2r,i-2r} = P_{n+2,i+2}$. □

Lemma 4.2.6. *Let n and i be integers such that $n \geq i$. The combined degree of the polynomial $P_{n,i}(b, c)$ is $n - i$.*

Proof. In Theorem 3.1.5 we showed that for positive n, the combined degree of the polynomial $P_{n,i}(b, c)$ is $n - i$. Suppose then that n is not positive. Certainly $-n + 2$ is positive and so, since $P_{n,i} = P_{-n+2,i-2n+2}$, the combined degree is $-n + 2 - (i - 2n + 2) = n - i$. □

Let us start to investigate how to enumerate the words with exactly one repeat in Pebody codes. So let B_y be a Pebody code of length n. Suppose w is a word in B_y and has only one repeat. Suppose also that this repeat is at position i. Hence $w_i = y_i = w_{i+1}$.

It is important to notice at this point that unless i is 1 or $n-1$, the extreme values that i may take, it is not possible for w_i to take the value b or 1. To demonstrate this, let u and v be the words $w_1 w_2 w_3 \ldots w_i$ and $w_{i+1} w_{i+2} w_{i+3} \ldots w_n$ respectively. Since w has only one repeat, the word u is up/down. If i is even then v is up/down and if i is odd then v is down/up. The word u^R, the reverse of u, is up/down or down/up if i is odd or even respectively. This means that one of v and u^R is up/down and the other is down/up.

The lengths of v and u^R are both more than 1 as i is not 1 or $n-1$ by assumption. Both v and u^R start with the same letter (they both have first letter equal to y_i).

Words that are up/down and words that are down/up may not start with b and 1 respectively. Since one of v and u^R is up/down and the other down/up we have that if y_i were to take the value 1 or b then there could be no such word w.

We already have the mechanism in place to succinctly describe the frequency of words such as u and v as introduced in this latest observation. This can be seen in the following two results.

Lemma 4.2.7. *Let i be a positive integer and c a letter in $[b]$. The number of up/down words of length i with final letter c is $P_{-1,-i}(b, c)$*

Proof. Suppose that i is odd and recall the bijective self-inverse in the proof of Corollary 4.1.3 defined by reversing words. The word $w \in A_{i,b}$ is an up/down word ending in the letter c if and only if w^R is an up/down word with first letter c. There are $P_{i,1}(b, c)$ words in $A_{i,b}$ with first letter c and so it follows that there are $P_{i,1}(b, c)$ up/down words ending with the letter c. Since $i + 1$ is even, application of Lemma 4.2.5 yields $P_{i,1}(b, c) = P_{-1,-i}(b, c)$.

Suppose then that i is even and let S be the set of words $w \in A_{i,b}$ with last letter c. Define S^R to be the set of reverses of the words in S so that the operation of reversing words maps S to S^R. Since this operation is bijective $|S| = |S^R|$.

Consider now the set of up/down words of length $i + 1$. Let T be the set of tails of length i with first letter c of these words. Then $|T| = P_{i+1,2}(b, c)$. Suppose t is an element of T, then t is an down/up word with first letter c. Since i is even t^R is up/down. Therefore $t^R \in S$, so $t \in S^R$ and $T \subseteq S^R$.

So that we may show that $T = S^R$, suppose t is an element of S^R. Since i is even and positive, we have that the length of t is at least two. As t is down/up, we also have that $t_1 > t_2$. So $t_1 > 1$. Define w to be the word $1t_1t_2 \dots t_i$. The word w has length $i + 1$ and is up/down. The tail of length i of w is t and so $S^R \subseteq T$. Therefore $|T| = |S^R|$ and so $|S| = P_{i+1,2}(b, c)$. Since $i + 2$ is even, we apply Lemma 4.2.5 to get $P_{i+1,2}(b, c) = P_{-1,-i}(b, c)$ as required. $\square$

Theorem 4.2.8. *Let B_y be a Pebody code of length n. The number of words in B_y with a repeat at position i and no other repeats is $P_{-1,-i}(b, y_i)P_{n,i+1}(b, y_i)$.*

Proof. Let U be the set of up/down words of length i that end with the letter y_i, so

$$U := \{u \in A_{i,b} : u_i = y_i\}.$$

Let V be the set of tails of length $n - i$ with first letter y_i of the up/down words of length n, so

$$V := \left\{v \in [b]^{n-i} : v_1 = y_i \text{ and } v \text{ is a tail of a word in } A_{n,b}\right\}.$$

Let W be the set of words in B_y with a single repeat where the position of that repeat is i, so

$$W := \{w \in B_y : w_i = w_{i+1} \text{ and } w \text{ only has one repeat}\}.$$

We shall now find a bijection between $U \times V$ and W. Define

$$\Phi : U \times V \to W$$

such that Φ maps the pair (u, v) to the concatenated word

$$w = u_1 u_2 u_3 \ldots u_i v_1 v_2 v_3 \ldots v_{n-i}.$$

Let j be any integer from 1 to $n-1$. If j is odd we need $w_j \leq w_{j+1}$, $w_j \geq w_{j+1}$ if j is even and $w_j = y_j = w_{j+1}$ in the cases where $w_j = w_{j+1}$ to prove that w is a word in B_y. Suppose then that j is odd. If $j < i$ then, since $u_j < u_{j+1}$, $w_j = u_j$ and $w_{j+1} = u_{j+1}$, we have that $w_j < w_{j+1}$. Suppose then that $j > i$. Since v is the tail of an up/down word of length n, x say, $x_j < x_{j+1}$. We have $w_j = v_{j-i} = x_j$ and $w_{j+1} = v_{j-i+1} = x_{j+1}$ so $w_j < w_{j+1}$. As $w_i = u_i = y_i$ and $w_{i+1} = v_1 = y_i$ we have $w \in B_y$ and also that w has only one repeat and that this repeat has position i. Therefore $w \in W$ and Φ is well defined.

Now let w be a word in W. Then w may be split into two words $u = w_1 w_2 w_3 \ldots w_i$ and $v = w_{i+1} w_{i+2} w_{i+3} \ldots w_n$. Since w has only one repeat and that the position of this repeat is i, the word u will be up/down. The word $w \in B_y$ has a repeat at position i and so $w_i = y_i$. Hence $u \in U$. Define the word x so that for every integer j from 1 to $i-1$ and from $i+1$ to n, $x_j = w_j$. If i is odd set $x_i = 1$ and if i is even set $x_i = b$. The word x is up/down and its tail of length $n-i$ is v therefore $v \in V$. Therefore Φ is surjective.

Obviously Φ is injective, so it is a bijection and so $|U \times V| = |W|$. Lemma 3.1.4 tells us that $|V| = P_{n,i+1}(b, y_i)$ and Lemma 4.2.7 tell us that $|U| = P_{-1,-i}(b, y_i)$. Since $|U \times V| = |U||V|$, the result follows. □

4.2.2 Independently choosing the letters of y to maximise the Pebody code B_y

In the remainder of this chapter we shall describe a method for estimating the size of a largest Pebody code for fixed n and sufficiently large b. In order to do this we introduce a new variable f_k, so that $y_k = bf_k$ for k from 1 to $n-1$. Note that since $1 \leq y_k \leq b$ we have that $f_k \in [0, 1]$.

Theorem 4.2.9. *Let n be fixed. For $1 \leq k \leq n-1$ there exist polynomials $C_{n,k}(f_k)$ such that*

$$|B_y| = |A_{n,b}| + \sum_{k=1}^{n-1} C_{n,k}(f_k) b^{n-2} + O(b^{n-3})$$

where $y_k = bf_k$.

Proof. There are only $O(b^{n-3})$ words in B_y with more than one repeat. Words in B_y that have no repeats are up/down and hence are contained in $A_{n,b}$. Therefore we need only account for words that have a single repeat. If the position of a repeat is k then k may take the values $1, 2, 3, \ldots, n-1$ and by the definition of a Pebody code, the letter being repeated is necessarily y_k. By Theorem 4.2.8 there are therefore

$$\sum_{k=1}^{n-1} P_{-1,-k}(b, y_k) P_{n,k+1}(b, y_k) \tag{4.1}$$

words in B_y with a single repeat. Substituting bf_k for y_k in (4.1) we define $C_{n,k}(f_k)$ as the coefficient of b^{n-2}. This coefficient is a polynomial in f_k.

Recall that in Lemma 4.2.6 we proved that the combined degree of $P_{n,i}(b, c)$ is $n - i$. The combined degrees of $P_{-1,-k}(b, y_k)$ and $P_{n,k+1}(b, y_k)$ are therefore $k - 1$ and $n - k - 1$ respectively so their product has combined degree $n - 2$. This means that when we make the substitution $y_k = bf_k$ we are left with a polynomial in b of degree $n - 2$ whose coefficients are polynomials in f_k. The number of words with a repeat at position k and no other repeats is therefore

$$P_{-1,-k}(b, bf_k) P_{n,k+1}(b, bf_k) = C_{n,k}(f_k) b^{n-2} + O(b^{n-3}).$$

Summing over the possible values of k completes the proof. □

Corollary 4.2.10. *Fix n to be a positive integer. Let $\bar{y}$ and $\hat{y}$ be words in $[b]^{n-1}$ such that $\bar{y}_k = b\bar{f}_k$, where $\bar{f}_k$ maximises $C_{n,k}(f)$ over $[0, 1]$, and $\hat{y}$ maximises the Pebody code B_y. Then*

$$|B_{\hat{y}}| = |B_{\bar{y}}| + O(b^{n-3}).$$

Proof. For any Pebody code B_y let $\Psi(y, b)$ be the number of words with more than one repeat. Clearly $\Psi(y, b)$ is a positive function and by Lemma 4.2.3 we have that $\Psi(y, b) = O(b^{n-3})$. Using similar reasoning to the proof of Theorem 4.2.9, we can see that

$$|B_y| = |A_{n,b}| + \sum_{k=1}^{n-1} P_{-1,-k}(b, y_k) P_{n,k+1}(b, y_k) + \Psi(y, b). \tag{4.2}$$

Let us define the polynomial $C_{n,k,l}(f)$ to be the coefficient of b^l in the summand $P_{-1,-k}(b, bf) P_{n,k+1}(b, bf)$. This definition is just an extension of the ideas employed in Theorem 4.2.9 and $C_{n,k,n-2}(f)$ is simply $C_{n,k}(f)$. We now rewrite (4.2) as

$$|B_y| = |A_{n,b}| + \sum_{k=1}^{n-1} \sum_{l=0}^{n-2} C_{n,k,l}(\tfrac{y_k}{b}) b^l + \Psi(y, b).$$

Recall that we chose the $\bar{f}_k$ to maximise $C_{n,k}(f)$, so $C_{n,k}(\frac{\hat{y}_k}{b}) \leq C_{n,k}(\bar{f}_k)$. Hence

$$\begin{aligned}
|B_{\hat{y}}| - |B_{\bar{y}}| &= \sum_{k=1}^{n-1}\sum_{l=0}^{n-2} \left(C_{n,k,l}(\tfrac{\hat{y}_k}{b}) - C_{n,k,l}(\bar{f}_k)\right) b^l + \Psi(\hat{y}, b) - \Psi(\bar{y}, b) \\
&= \sum_{k=1}^{n-1} \left(C_{n,k}(\tfrac{\hat{y}_k}{b}) - C_{n,k}(\bar{f}_k)\right) \\
&\quad + \sum_{k=1}^{n-1}\sum_{l=0}^{n-3} \left(C_{n,k,l}(\tfrac{\hat{y}_k}{b}) - C_{n,k,l}(\bar{f}_k)\right) b^l + \Psi(\hat{y}, b) - \Psi(\bar{y}, b) \\
&\leq \sum_{k=1}^{n-1}\sum_{l=0}^{n-3} \left(C_{n,k,l}(\tfrac{\hat{y}_k}{b}) - C_{n,k,l}(\bar{f}_k)\right) b^l + \Psi(\hat{y}, b) - \Psi(\bar{y}, b) \\
&\leq \sum_{k=1}^{n-1}\sum_{l=0}^{n-3} \left(C_{n,k,l}(\tfrac{\hat{y}_k}{b}) - C_{n,k,l}(\bar{f}_k)\right) b^l + \Psi(\hat{y}, b).
\end{aligned}$$

Each of the polynomials $C_{n,k,l}(f)$ are bounded above and below over the interval $[0, 1]$ and so we can find a positive real number $\delta_{n,k,l}$ at least as big as the difference between these bounds. I.e. for any $f, g \in [0, 1]$ we have

$$\delta_{n,k,l} \geq \left|C_{n,k,l}(f) - C_{n,k,l}(g)\right|.$$

Define $\Delta_{n,l}$ to be $\sum_{k=0}^{n-1} \delta_{n,k,l}$. The difference between the number of words in $B_{\bar{y}}$ and $B_{\hat{y}}$ is therefore

$$\begin{aligned}
|B_{\hat{y}}| - |B_{\bar{y}}| &\leq \left|\sum_{k=1}^{n-1}\sum_{l=0}^{n-3} C_{n,k,l}(\tfrac{\hat{y}_k}{b}) - C_{n,k,l}(\bar{f}_k) b^l + \Psi(\hat{y}, b)\right| \\
&\leq \sum_{k=1}^{n-1}\sum_{l=0}^{n-3} \left|C_{n,k,l}(\tfrac{\hat{y}_k}{b}) - C_{n,k,l}(\bar{f}_k)\right| b^l + \Psi(\hat{y}, b) \\
&\leq \sum_{l=0}^{n-3}\sum_{k=1}^{n-1} \delta_{n,k,l} b^l + \Psi(\hat{y}, b) \\
&= \sum_{l=0}^{n-3} \Delta_{n,l} b^l + \Psi(\hat{y}, b).
\end{aligned}$$

Therefore $|B_{\hat{y}}| - |B_{\bar{y}}| = O(b^{n-3})$ and the result follows. □

In Theorem 4.2.9 we showed that the difference between the sizes of the alternating code and the Pebody code is $\sum_{k=1}^{n-1} C_{n,k}(f_k) b^{n-2} + O(b^{n-3})$. So far we have demonstrated, in the proof of Theorem 4.2.9, that the degree of the polynomials $P_{-1,-k}(b, bf) P_{n,k+1}(b, bf)$ is $n - 2$ and we defined $C_{n,k}(f)$ to be its leading coefficient. For a better understanding of the nature of the functions $C_{n,k}(f)$, we study them in detail in the following chapters with the aims of calculating and eventually approximating them.

Chapter 5

Counting Words in the Alternating Code starting with a given Letter

Up to this point we have seen the importance of the polynomial $P_{n,i}(b,c)$ in understanding the size of Pebody codes. In particular, the coefficient of the highest power of b in $P_{n,i}(b,bf)$ is a polynomial, which we shall call $F_{n,i}(f)$, that plays a crucial rôle. We now represent the function $C_{n,i}(f)$, discussed at the end of the previous chapter, in terms of this new polynomial so that

$$C_{n,i}(f) = F_{-1,-i}(f)F_{n,i+1}(f).$$

In the next section we uncover properties of this polynomial that help simplify the way we treat it. For example, we discover that we need only consider the polynomial $F_{n,1}(f)$ in order to understand the polynomial $F_{n,i}(f)$. We then observe that the first few polynomials in the sequence $\{F_{n,1}\}_{n=1}^{\infty}$, which we generate computationally, suggest convergence as $n \to \infty$. The chapter concludes by proving this result.

5.1 A recursive counting method

5.1.1 Approximating the "sum of sums" method by integration

Recall the definition of the polynomials $P_{n,i}(b,c)$ given in Definition 4.2.4. Since we are mostly interested in the terms of highest combined degree, let us consider the substitution $c = fb$. This allows the terms of highest combined degree in $P_{n,i}(b,c)$ to be succinctly represented by the coefficient of b^{n-i} in $P_{n,i}(b,bf)$ as described in the following definition.

Definition 5.1.1. *Define the polynomial $F_{n,i}(f)$ to be the coefficient of the highest power of b in $P_{n,i}(b, bf)$.*

Definition 5.1.2. *Let b, n and i be integers with $b > 0$ and $n > i$. Define $I_{n,n} := 1$ and, for $c \in [0, b]$, recursively define*

$$I_{n,i}(b, c) := \int_c^b I_{n,i+1}(b, x)\, dx$$

when i is odd and

$$I_{n,i}(b, c) := \int_0^c I_{n,i+1}(b, x)\, dx$$

when i is even.

Note that $I_{n,i}(b, c)$ defined as above is a polynomial in b and c.

Theorem 5.1.3. *The polynomial $I_{n,i}(b, bf)$ agrees with $P_{n,i}(b, bf)$ in the highest power of b, so $I_{n,i}(b, bf) = P_{n,i}(b, bf) + O(b^{n-i-1})$ with equality when $n = i$.*

Proof. By definition $I_{n,n} = P_{n,n}$. Suppose that $I_{n,i+1}(b, w_{i+1})$ and $P_{n,i+1}(b, w_{i+1})$ agree in all the terms of highest combined degree. The polynomial $P_{n,i+1}(b, w_{i+1})$ has combined degree $n - (i + 1)$. Consider a term $\alpha b^r w_{i+1}^s$ of $P_{n,i+1}(b, w_{i+1})$; using the recursive definition of $P_{n,i}(b, w_i)$, the contribution of $\alpha b^r w_{i+1}^s$ to this polynomial when i is even is

$$\sum_{w_{i+1}=1}^{w_i - 1} \alpha b^r w_{i+1}^s = \alpha b^r \sum_{w_{i+1}=1}^{w_i - 1} w_{i+1}^s.$$

Applying Lemma 2.0.7 to this sum gives us

$$\alpha\, b^r \left(\frac{w_{i+1}^{s+1}}{s+1} + p_s(w_{i+1}^s) \right) \Bigg|_{w_{i+1} = w_i - 1}$$

where p_s is some polynomial of degree at most s. After evaluating this expression at $w_{i+1} = w_i - 1$ and expanding the $\frac{(w_i-1)^{s+1}}{s+1}$ term, we can see the contribution is

$$\alpha b^r \left(\frac{w_i^{s+1}}{s+1} + \text{lower order terms} \right).$$

The lower order terms are polynomial in w_i and have degree at most s.

When i is odd then the limits of the sum are $w_i + 1$ and b, so in this case the contribution is

$$\sum_{w_{i+1}=w_i+1}^{b} \alpha b^r w_{i+1}^s = \sum_{w_{i+1}=1}^{b} \alpha b^r w_{i+1}^s - \sum_{w_{i+1}=1}^{w_i} \alpha b^r w_{i+1}^s$$
$$= \alpha b^r \left[\frac{w_{i+1}^{s+1}}{s+1} + p_s(w_{i+1}) \right]_{w_i}^{b}$$
$$= \alpha b^r \left(\frac{b^{s+1} - w_i^{s+1}}{s+1} + \text{lower order terms} \right).$$

Here the lower order terms are polynomial in w_i and b but still have combined degree at most s.

So, a term of combined degree $r+s$ in $P_{n,i+1}(b, w_{i+1})$ makes a contribution of combined degree $r+s+1$ to $P_{n,i}(b, w_i)$. Hence only terms of highest combined degree in $P_{n,i+1}(b, w_{i+1})$ contribute to terms of highest combined degree in $P_{n,i}(b, w_i)$.

Similarly for $I_{n,i+1}(b, w_{i+1})$, only terms of highest degree contribute to the terms of highest degree of $I_{n,i}(b, w_i)$. In fact, $I_{n,i}(b, w_i)$ is homogeneous. The contribution to $I_{n,i}(b, w_i)$ of the term $\alpha b^r w_{i+1}^s$ is $\alpha b^r \frac{w_i^{s+1}}{s+1}$ and $\alpha b^r \frac{b^{s+1} - w_i^{s+1}}{s+1}$ when i is even and odd respectively, which agrees in the terms of highest combined degree with the contribution this term would have made to $P_{n,i}(b, w_i)$.

By assumption, the polynomials $P_{n,i+1}(b, w_{i+1})$ and $I_{n,i+1}(b, w_{i+1})$ agree in the terms of highest degree, so the polynomials $I_{n,i}(b, w_i)$ and $P_{n,i}(b, w_i)$ also agree in the terms of highest degree. Therefore the polynomial $I_{n,i}(b, bf)$ agrees with $P_{n,i}(b, bf)$ in the highest power of b. □

Corollary 5.1.4. *We have* $F_{n,i}(f) = I_{n,i}(1, f)$.

Proof. We saw in the proof of Theorem 5.1.3 that the contribution to $I_{n,i}(b, w_i)$ from the term $\alpha b^r w_{i+1}^s$ was homogenous and of degree $r+s+1$. Therefore if $I_{n,i+1}(b, w_i)$ is homogenous then $I_{n,i}(b, w_i)$ is homogenous. As $I_{n,n} = 1$ is homogenous, by induction, $I_{n,i}(b, w_i)$ is homogenous. This means that $I_{n,i}(b, bf) = b^{n-i} F_{n,i}(f)$ and so $I_{n,i}(1, f) = F_{n,i}(f)$. □

The consequence of Corollary 5.1.4 is that, by definition of $I_{n,i}(f)$, we can say

$$F_{n,i}(b, w_i) := \int_{w_i}^{b} F_{n,i+1}(b, w_{i+1})\, dw_{i+1}$$

when i is odd and

$$F_{n,i}(b, w_i) := \int_{0}^{w_i} F_{n,i+1}(b, w_{i+1})\, dw_{i+1}$$

when i is even.

Now that we actually have a recursive definition of the polynomials $F_{n,i}$ we can start to describe them more fully.

Theorem 5.1.5. *The coefficients of $F_{n,i}(f)$ may themselves be defined recursively.*

Proof. The polynomial $F_{n,i+1}(f)$ has degree $n-(i+1)$, so let us represent it by

$$\sum_{j=0}^{n-(i+1)} \alpha_j f^j.$$

Then, when i is even,

$$\begin{aligned}
F_{n,i}(f) &= \int_0^f \sum_{j=0}^{n-(i+1)} \alpha_j f^j \, df \\
&= \sum_{j=0}^{n-(i+1)} \alpha_j \int_0^f f^j \, df \\
&= \sum_{j=0}^{n-(i+1)} \frac{\alpha_j}{j+1} f^{j+1} \\
&= \sum_{j=1}^{n-i} \frac{\alpha_{j-1}}{j} f^j,
\end{aligned}$$

and when i is odd

$$\begin{aligned}
F_{n,i}(f) &= \int_f^1 \sum_{j=0}^{n-(i+1)} \alpha_j f^j \, df \\
&= \sum_{j=0}^{n-(i+1)} \alpha_j \int_f^1 f^j \, df \\
&= \sum_{j=0}^{n-(i+1)} \frac{\alpha_j}{j+1} (1 - f^{j+1}) \\
&= \sum_{j=0}^{n-(i+1)} \frac{\alpha_j}{j+1} + \sum_{j=1}^{n-i} \frac{-\alpha_{j-1}}{j} f^j.
\end{aligned}$$

□

Just as we proved in Lemma 4.2.5 that $P_{n,i} = P_{n+2,i+2}$, it follows by definition that the same holds for the polynomials $F_{n,i}$.

Lemma 5.1.6. *Let n and i be integers with $n \geq i$, then $F_{n,i} = F_{n+2,i+2}$.*

A simple corollary of this lemma is that for even n we have $F_{n,i} = F_{0,i-n}$ and when n is odd $F_{n,i} = F_{1,i-n+1}$. Any polynomial $F_{n,i}$ is therefore contained in one of the following two families of polynomials

$$\{F_{1,1}, F_{1,0}, F_{1,-1}, \dots\} \text{ or } \{F_{0,0}, F_{0,-1}, F_{0,-2}, \dots\}.$$

So we may restrict our attention to just these polynomials.

5.1.2 Generalising the recursive counting method

Recall the two families of polynomials mentioned at the end of the previous section. Let n be a positive integer and consider the polynomials $F_{1,1-n}(f)$ and $F_{0,-n}(f')$. These polynomials are the coefficients of the highest power of b in the polynomials $P_{1,1-n}(b, bf)$ and $P_{0,-n}(b, bf')$ respectively. We therefore may write:

$$\begin{aligned} P_{1,1-n}(b, bf) &= F_{1,1-n}(f)b^n + O(b^{n-1}), \\ P_{0,-n}(b, bf') &= F_{0,-n}(f')b^n + O(b^{n-1}). \end{aligned}$$

Let c and c' be integers between 1 and b and set $f = c/b$ and $f' = c'/b$. Suppose for the moment that n is even, then $P_{1,1-n}(b, c)$ counts the number of up/down words of length n with first letter c and $P_{0,-n}(b, c')$ counts the number of down/up words of length n with first letter c'. Take this set of up/down words of length n with first letter c and relabel the letters $1, 2, 3, \ldots, b$ to $b, b-1, b-2, \ldots, 1$ in all the words. This relabelling is clearly self-inverse and therefore bijective. Note that our relabelled words are now down/up and all have first letter $b - c + 1$. Therefore $P_{0,-n}(b, b - c + 1)$ must be at least $P_{1,1-n}(b, c)$. By the same relabelling argument $P_{1,1-n}(b, c) \geq P_{0,-n}(b, b - c + 1)$ hence

$$P_{1,1-n}(b, c) = P_{0,-n}(b, b - c + 1).$$

This relationship can also be shown for odd n and is mimicked in the polynomials $F_{n,i}$ as is detailed in the following theorem.

Theorem 5.1.7. *Let n and i be integers with $i \leq n$ and f a real number in the interval $[0, 1]$, then $F_{n,i}(1 - f) = F_{n+1,i+1}(f)$.*

Proof. By definition $F_{n,n} = 1 = F_{n+1,n+1}$, so let us suppose that for all integers j such that $n \geq j > i$ we have

$$F_{n,j}(1 - f) = F_{n+1,j+1}(f).$$

Using the integral definition of $F_{n,i}$, we have that, when i is even:

$$\begin{aligned} F_{n,i}(1 - f) &= \int_0^{1-f} F_{n,i+1}(f')\, df' \\ &= \int_0^{1-f} F_{n+1,i+2}(1 - f')\, df' \end{aligned}$$

by applying the inductive hypothesis. If we now perform the substitution $g = 1 - f'$

we have

$$\begin{aligned}
F_{n,i}(1-f) &= -\int_1^f F_{n+1,i+2}(g)\,dg \\
&= \int_f^1 F_{n+1,i+2}(g)\,dg \\
&= F_{n+1,i+1}(f).
\end{aligned}$$

When i is odd

$$\begin{aligned}
F_{n,i}(1-f) &= \int_{1-f}^1 F_{n,i+1}(f')\,df' \\
&= \int_{1-f}^1 F_{n+1,i+2}(1-f')\,df'.
\end{aligned}$$

Once again we make the substitution $g = 1 - f'$, thus

$$\begin{aligned}
F_{n,i}(1-f) &= -\int_f^0 F_{n+1,i+2}(g)\,dg \\
&= \int_0^f F_{n+1,i+2}(g)\,dg \\
&= F_{n+1,i+1}(f).
\end{aligned}$$

□

We may now focus our attention on just one of the families of polynomials, given that they are so simply related, but this still is fairly awkward. Either family at each recursion alternately generates a polynomial that describes up/down words or down/up words, the coefficients of which bear no resemblance to each other. We would do better to look at the sequence of functions $\{F_{n,1}(f)\}_{n=1}^{\infty}$, since this sequence describes polynomials counting only up/down words; moreover, once we understand this sequence determining a general $F_{n,i}(f)$ is trivial.

5.1.3 Implementing the counting method on a computer

The purpose of this section is to calculate and inspect the first few terms of the aforementioned sequence of functions $\{F_{n,1}(f)\}_{n=1}^{\infty}$. The analysis that follows motivates the rest of the chapter. Firstly we shall use the result in Theorem 5.1.7 to derive a useful relation between the functions $F_{n,1}(f)$.

Lemma 5.1.8. *Let n be a nonnegative integer then*

$$F_{n+1,1}(f) = \int_0^{1-f} F_{n,1}(f')\,df'.$$

Proof. By definition we have

$$F_{n+1,1}(f) = \int_f^1 F_{n+1,2}(g)\, dg.$$

If we apply Theorem 5.1.7 and substitute $f' = 1 - g$ we find

$$\begin{aligned} F_{n+1,1}(f) &= \int_f^1 F_{n,1}(1-g)\, dg \\ &= -\int_{1-f}^0 F_{n,1}(f')\, df' \\ &= \int_0^{1-f} F_{n,1}(f')\, df'. \end{aligned}$$

□

We make use of Lemma 5.1.8 to calculate the functions $F_{n,1}(f)$, the first ten of which are shown in Table 5.1. The MatLab code used to generate the polynomials is given in Appendix B.

As n increases, the coefficients of the polynomials in Table 5.1 appear to tend to zero. Unfortunately, this does not readily allow us to draw any conclusions about the behaviour of the functions over the interval $[0, 1]$, but does motivate the following definition.

Definition 5.1.9. *Let $F_{n,1}(f)$ be as in Definition 5.1.1. The "normalised" polynomial $\tilde{F}_{n,1}(f)$ is defined*

$$\tilde{F}_{n,1}(f) := \frac{F_{n,1}(f)}{F_{n,1}(0)}.$$

The purpose of normalising our sequence of polynomials in this way, as opposed to say making them monic, is two fold. Firstly, since we are only interested in the polynomials over the domain $(0, 1]$, the constant coefficient will have a greater relative effect than higher order terms. Secondly, we expect $F_{n,1}(f)$ to be a non-strictly decreasing function since when b is fixed the following lemma shows this to be the case for $P_{n,1}(b, c)$.

Lemma 5.1.10. *Let b, n and c be positive integers with $n > 1$ and $c < b$. Then*

$$P_{n,1}(b, c) \geq P_{n,1}(b, c+1).$$

Proof. This proof is very much along the lines of the proof of Lemma 4.1.1. Let D and E be the sets of words in $A_{n,b}$ with first letters c and $c+1$ respectively. Define the map $\Psi : E \to D$ by $e \mapsto d$, where $d_1 = c$ and for i from 2 to n, $d_i = e_i$.

Let e be an element of E, then e has first letter $c+1$ and is up/down. Let $d = \Psi(e)$, then $d_1 = c$. Therefore $d_1 < d_2$ since $e_1 < e_2$ and $e_1 = c$ and $e_2 = d_2$.

Table 5.1: The coefficients of the polynomials in the sequence $\{F_{n,1}(f)\}_{n=1}^{10}$

	f^9	f^8	f^7	f^6	f^5	f^4	f^3	f^2	f	1
$F_{1,1}$	0	0	0	0	0	0	0	0	0	1
$F_{2,1}$	0	0	0	0	0	0	0	0	-1	1
$F_{3,1}$	0	0	0	0	0	0	0	-.50000	0	.50000
$F_{4,1}$	0	0	0	0	0	0	.16667	-.50000	0	.33333
$F_{5,1}$	0	0	0	0	0	.41667e-1	0	-.25000	0	.20833
$F_{6,1}$	0	0	0	0	-.83333e-2	.41667e-1	0	-.16667	0	.13333
$F_{7,1}$	0	0	0	-.13889e-2	0	.20833e-1	0	-.10417	0	.84722e-1
$F_{8,1}$	0	0	.19841e-3	-.13889e-2	0	.13889e-1	0	-.66667e-1	0	.53968e-1
$F_{9,1}$	0	.24802e-4	0	-.69444e-3	0	.86806e-2	0	-.42361e-1	0	.34350e-1
$F_{10,1}$	-.27557e-5	.24802e-4	0	-.46296e-3	0	.55556e-2	0	-.26984e-1	0	.21869e-1

Since e and d agree in their tails of length $n-1$ and e is up/down, d must be up/down. Therefore Ψ is well defined.

Let e and e' be words in E. Suppose $\Psi(e) = \Psi(e')$, then since e and $\Psi(e)$ agree in all letters but the first one, as do e' and $\Psi(e')$, e and e' must agree in at least all letters but the first one. However, by definition $e_1 = c+1 = e'_1$, so we must have that $e = e'$; hence Ψ is injective and $|D| \geq |E|$. Since $P_{n,1}(b,c)$ and $P_{n,1}(b,c+1)$ enumerate D and E respectively, the result follows. □

We now give the normalised functions $\tilde{F}_{n,1}(f)$ in Table 5.2. This time the coefficients strongly suggest some sort of convergence and this is our motivation for looking at what function if any they converge to.

5.2 Convergence of the counting method

5.2.1 Background on Hilbert spaces

Throughout the remainder of this chapter we shall be talking of convergence of sequences of functions and will use the theory of functions as set out in [Hal57], [Lar73] and [Tit39]. We shall be working in the Hilbert space L^2 with the standard inner product and it is to be taken that we are referring to convergence over the metric derived from the inner product norm.

Definition 5.2.1. *Let F and G be two functions on $\mathbb{R}$ and let μ be the Lebesgue measure. Let S be the set of points s such that $F(s) \neq G(s)$. If S has zero measure, i.e. if $\mu(S) = 0$, then we say F and G are* equal almost everywhere *and write*

$$F \underset{ae}{=} G.$$

Remark 5.2.2. *Titchmarsh provides us with the tools we need in measure theory and Lebesgue integration with [Tit39], though we should explicitly observe that the binary relation $=_{ae}$ is an equivalence relation.*

Definition 5.2.3. *Let $c, d \in \mathbb{R}$, the space $L^2[c,d]$ (also written $L^2_{\mathbb{C}}[c,d]$) is the set of equivalence classes of functions $F(f)$ over $[c,d]$ such that the Lebesgue integral $\frac{1}{d-c}\int_c^d F(f)\overline{F(f)}\,df$ is defined. The equivalence classes are of course determined by $=_{ae}$.*

There is a standard practice of "abuse of notation" in this field whereby the statement

$$F \in L^2[c,d]$$

is taken to mean that F is a representative of its equivalence class, which in turn is a member of $L^2[c,d]$. One of the consequences of this is as follows: when we say $F \in L^2[c,d]$, we do not mean that F evaluated at a point is representative of

Table 5.2: The coefficients of the normalised polynomials in the sequence $\{\tilde{F}_{n,1}(f)\}_{n=1}^{10}$

	f^9	f^8	f^7	f^6	f^5	f^4	f^3	f^2	f	1
$\tilde{F}_{1,1}$	0	0	0	0	0	0	0	0	0	1
$\tilde{F}_{2,1}$	0	0	0	0	0	0	0	0	-1	1
$\tilde{F}_{3,1}$	0	0	0	0	0	0	0	-1	0	1
$\tilde{F}_{4,1}$	0	0	0	0	0	0	.50000	-1.5000	0	1
$\tilde{F}_{5,1}$	0	0	0	0	0	.20000	0	-1.2000	0	1
$\tilde{F}_{6,1}$	0	0	0	0	-.62500e-1	.31250	0	-1.2500	0	1
$\tilde{F}_{7,1}$	0	0	0	-.16393e-1	0	.24590	0	-1.2295	0	1
$\tilde{F}_{8,1}$	0	0	.36765e-2	-.25735e-1	0	.25735	0	-1.2353	0	1
$\tilde{F}_{9,1}$	0	.72202e-3	0	-.20217e-1	0	.25271	0	-1.2332	0	1
$\tilde{F}_{10,1}$	-.12601e-3	.11341e-2	0	-.21169e-1	0	.25403	0	-1.2339	0	1

the values other functions equivalent to F take when evaluated at the same point. Moreover, F is identified with functions that take every conceivable value at this point and as such F evaluated at a point is not considered well defined.

Definition 5.2.4. *For two functions $F(f)$ and $G(f)$ in $L^2[c,d]$, we denote their inner product, $\frac{1}{d-c}\int_c^d F(f)\overline{G(f)}\,df$, as $\langle F(f), G(f)\rangle$, or more simply $\langle F, G\rangle$.*

Definition 5.2.5. *For a function F in $L^2[c,d]$ define the norm of F to be the nonnegative square root of $\langle F, F\rangle$ and denote it by $\|F\|$. This is indeed a norm; see* [Hal57] *for a treatment of this.*

We define normality and orthogonality in the standard way with the above definitions. So F and G are orthogonal if and only if $\langle F, G\rangle = 0$ and we say F is normal when $\|F\| = 1$.

Definition 5.2.6. *A set S of functions is an* orthogonal set *when its elements are pairwise orthogonal and we say S is an* orthonormal set *when, in addition, the members are all normal.*

Definition 5.2.7. *An* orthonormal basis *of a vector space V over a field Φ is an orthonormal set $B = \{e_i : i \in I\}$ such that for any vector v in V, we may write*

$$v = \sum_{i \in I} \alpha_i e_i$$

where the α_i are in Φ.

Definition 5.2.8. *A* complete orthonormal set *is a maximal orthonormal set.*

Theorem 5.2.9. *In a Hilbert space, a set is an orthonormal basis if and only if it is a complete orthonormal set.*

Proof. Larsen proves this in [Lar73]. □

5.2.2 Assuming convergence to obtain a hypothetical family of limits

We shall have need of the following lemmata for Theorem 5.2.14.

Lemma 5.2.10. *Consider a sequence of functions $\{G_n(f)\}_{n=1}^{\infty}$, each with domain* $[0,1]$. *Suppose $G_1(f) = 1$ and that $G_{n+1}(f) = \int_0^{1-f} G_n(f')\,df'$, then each of the functions in the sequence is bounded above by 1 and is nonnegative over* $[0,1]$.

Proof. Clearly each of the functions $G_n(f)$ is real over $[0,1]$. Now, the function $G_1(f)$ is bounded above by 1 and is nonnegative over $[0,1]$, so let us suppose that $G_n(f)$ is also. Therefore

$$0 = \int_0^{f-1} 0\,df' \le \int_0^{f-1} G_n(f')\,df'\,df = G_{n+1}(f).$$

The upper bound is found similarly, thus

$$1 \geq \int_0^{f-1} 1 \, df' \geq \int_0^{f-1} G_n(f') \, df' = G_{n+1}(f).$$

□

Lemma 5.2.11. *Let $\{G_n(f)\}_{n=1}^{\infty}$ be defined as in Lemma 5.2.10. Each of the functions is strictly decreasing over $[0,1]$ except for $G_1(f)$.*

Proof. Recall Lemma 5.2.10 which proves that each of the functions is nonnegative. Let f_1 and f_2 be two points in $[0,1]$ such that $f_1 < f_2$. Let n be an integer such that $n \geq 2$ and let us assume that $G_n(f)$ is a strictly decreasing function over $[0,1]$. Let g be a point in $(1-f_2, 1-f_1)$ then $G_n(g) > G_n(1-f_1)$, so

$$\begin{aligned}\int_{1-f_2}^{1-f_1} G_n(g) \, dg &> (f_2 - f_1) G_n(1-f_1) \\ &\geq 0.\end{aligned}$$

By definition of $G_{n+1}(f)$ we have

$$\begin{aligned}G_{n+1}(f_1) &= \int_0^{1-f_1} G_n(g) \, dg \\ &= \int_{1-f_2}^{1-f_1} G_n(g) \, dg + \int_0^{1-f_2} G_n(g) \, dg \\ &> \int_0^{1-f_2} G_n(g) \, dg \\ &= G_{n+1}(f_2).\end{aligned}$$

Therefore $G_{n+1}(f)$ is a strictly decreasing function over $[0,1]$. It is not hard to see that $G_2(f)$ is $1-f$ and that it is also strictly decreasing. This completes the induction. □

A rather simple consequence of Lemma 5.2.10 and Lemma 5.2.11 is that each function $G_n(f)$ described in the lemmata is positive over the interval $[0,1]$ apart from at the point 1 where $G_n(1) = 0$ for $n \geq 2$.

Lemma 5.2.12. *Let μ be the Lebesgue measure and $\mathscr{H}$ be a function in $L^2[0,1]$ that is real almost everywhere. Define $Z := \{f \in [0,1] : \mathscr{H}(f) \in \mathbb{R}, \mathscr{H}(f) < 0\}$. If $\mu(Z) > 0$ then we can find an $\epsilon > 0$ and a set $Y \subseteq Z$ such that $\mu(Y) > 0$ and $\inf_{f \in Y}\{\mathscr{H}(f)^2\} > \epsilon$.*

Proof. For all positive δ define

$$Z_\delta := \{f \in Z : \mathscr{H}(f)^2 > \delta\}.$$

Since $\mathscr{H}$ is a measurable function, Z_δ is a measurable set. If now for some δ we have that $\mu(Z_\delta) > 0$ then the proof is complete with $Y := Z_\delta$ and $\epsilon := \delta$.

Suppose then that all Z_δ have zero measure. Since μ is a measure and Z_δ and $Z \backslash Z_\delta$ are disjoint

$$\mu(Z) = \mu(Z_\delta) + \mu(Z \backslash Z_\delta),$$

and so

$$\int_Z \mathscr{H}^2 = \int_{Z_\delta} \mathscr{H}^2 + \int_{Z \backslash Z_\delta} \mathscr{H}^2.$$

An integral over a set of zero measure is always zero so we need only consider the integral over $Z \backslash Z_\delta$. Now, $\mathscr{H}^2(f) < \delta$ for all $f \in Z \backslash Z_\delta$ so

$$\int_Z \mathscr{H}^2 = \int_{Z \backslash Z_\delta} \mathscr{H}^2 \leq \int_{Z \backslash Z_\delta} \delta = \delta \mu(Z \backslash Z_\delta) = \delta \mu(Z).$$

Let $\chi_Z(f)$ be the characteristic function of Z over $[0, 1]$ so that it takes the value 1 when $f \in Z$ and 0 otherwise. We may now write

$$\|\chi_Z \mathscr{H}\|^2 = \int_Z \mathscr{H}^2 \leq \delta \mu(Z).$$

This means that $\chi_Z \mathscr{H}$ must be zero almost everywhere. Our function $\mathscr{H}$ is nonzero on Z and so it must follow that Z has zero measure. $\square$

Lemma 5.2.13. *Let $\{H_n\}_{n=1}^{\infty}$ be a sequence of real and nonnegative functions on the interval $[0, 1]$ that converges inside $L^2[0, 1]$ to a function $\mathscr{H}$, then $\mathscr{H}$ is real and nonnegative almost everywhere.*

Proof. Suppose that $\mathscr{H}$ is not nonnegative almost everywhere. Firstly we show that indeed $\mathscr{H}$ is real almost everywhere. Suppose that the imaginary part of $\mathscr{H}$, $Im(\mathscr{H})$, is not almost everywhere zero. Then $\|Im(\mathscr{H})\| \neq 0$ and since the functions H_n are real

$$\begin{aligned} |\mathscr{H} - H_n|^2 &= |Re(\mathscr{H}) + Im(\mathscr{H}) - H_n|^2 \\ &= (Re(\mathscr{H}) - H_n)^2 + (Im(\mathscr{H}))^2 . \end{aligned}$$

Hence

$$\begin{aligned} \|\mathscr{H} - H_n\|^2 &= \int |\mathscr{H} - H_n|^2 \\ &= \int (Re(\mathscr{H}) - H_n)^2 + \int (Im(\mathscr{H}))^2 \\ &\geq \int (Im(\mathscr{H}))^2 \\ &= \|Im(\mathscr{H})\|^2 . \end{aligned}$$

This is preposterous as $\{H_n\}_{n=1}^{\infty}$ converges in $L^2[0, 1]$ to $\mathscr{H}$.

Let μ be the Lebesgue measure and define Z as in Lemma 5.2.12, then $\mu(Z) > 0$ and so by Lemma 5.2.12 there exists an $\epsilon > 0$ and a set $Y \subseteq Z$ such that $\mu(Y) > 0$ and $\mathscr{H}(y)^2 > \epsilon$ for all $y \in Y$. Let y be a point in Y , then

$$\begin{aligned} |H_n(y) - \mathscr{H}(y)| &= H_n(y) - \mathscr{H}(y) \\ &\geq -\mathscr{H}(y) \\ &\geq \sqrt{\epsilon}. \end{aligned}$$

Now, proceeding in a similar fashion to the first half of this proof,

$$\begin{aligned} \|H_n - \mathscr{H}\|^2 &= \int |H_n - \mathscr{H}|^2 \\ &\geq \int_Y |H_n - \mathscr{H}|^2 \\ &\geq \int_Y \mathscr{H}^2 \\ &\geq \int_Y \epsilon \\ &= \epsilon\mu(Y). \end{aligned}$$

This is also preposterous and for the same reason. □

Theorem 5.2.14. *If the normalised sequence of polynomials $\{\tilde{F}_{n,1}(f)\}_{n=1}^{\infty}$ converges in $L^2[0,1]$ to a function $\mathscr{F}(f)$ that is nontrivial (i.e. $\|\mathscr{F}\| \neq 0$), then $\mathscr{F}(f) = \cos\frac{\pi f}{2}$.*

Proof. Recall Lemma 5.1.8 which states that

$$F_{n+1,1}(f) = \int_0^{1-f} F_{n,1}(f')\, df'.$$

We normalise the $F_{n,1}(f)$ by dividing through by the constant coefficient of the polynomial (or equivalently the value the polynomial takes at $f = 0$). As we mentioned earlier, Lemma 5.2.10 and Lemma 5.2.11 tell us that for $n \geq 2$, the $F_n(f)$ are nonnegative, strictly decreasing functions and from their definition we can see that $F_n(1) = 0$. This means that they are positive for the rest of the interval over which they are defined, in particular they are always positive at the origin. Our normalised polynomials are defined thus

$$\begin{aligned} \tilde{F}_{n,1}(f) &= \frac{F_{n,1}(f)}{F_{n,1}(0)} \\ &= \frac{\int_0^{1-f} F_{n-1,1}(f')\, df'}{\int_0^1 F_{n-1,1}(f')\, df'} \\ &= \frac{\int_0^{1-f} \tilde{F}_{n-1,1}(f')\, df'}{\int_0^1 \tilde{F}_{n-1,1}(f')\, df'}. \end{aligned}$$

Of course $\tilde{F}_{1,1}(f)$ is still $1(f)$ as this function is already normalised. For ease of notation let us define

$$\kappa := \frac{1}{\int_0^1 \mathscr{F}(f)\, df}.$$

Appealing to Lemma 5.2.13, $\mathscr{F}(f)$ is nonnegative almost everywhere and by assumption is nontrivial. The value κ is therefore well defined, real and positive. By assuming our sequence converges in $L^2[0,1]$ to $\mathscr{F}(f)$, in the limit we have $\mathscr{F}(f)$ satisfying

$$\mathscr{F}(f) \underset{\text{ae}}{=} \kappa \int_0^{1-f} \mathscr{F}(f')\, df'.$$

We cannot hope for $\mathscr{F}$ to be differentiable as, since it is a general member of $L^2[0,1]$ and hence representative of its equivalence class, it may be discontinuous everywhere! What we shall prove however is that it is equal almost everywhere to a function that is. Let us then define this function thus:

$$F(f) := \kappa \int_0^{1-f} \mathscr{F}(f')\, df'.$$

Certainly F is equal almost everywhere to $\mathscr{F}$, but in order to show that it is also differentiable we first prove that it is bounded, continuous and monotonically decreasing.

Since $\mathscr{F}$ is nonnegative almost everywhere, its integral is nonnegative and so F is nonnegative. Let f_1 and f_2 be two points in the interval $[0,1]$ such that $f_1 < f_2$. Since

$$\begin{aligned} F(f_1) - F(f_2) &= \int_0^{1-f_1} \mathscr{F}(f')\, df' - \int_0^{1-f_2} \mathscr{F}(f')\, df' \\ &= \int_{1-f_2}^{1-f_1} \mathscr{F}(f')\, df' \\ &\geq 0, \end{aligned}$$

the function F is monotonically decreasing. From the definition of F however, it is clear that $F(0)$ is 1, therefore F is bounded above by 1.

The functions F and $\mathscr{F}$ are equal almost everywhere, so their integrals over any set will be identical. Let ϵ be any positive real and let f and g be two points in the interval $[0,1]$ such that $f < g$ and

$$|f - g| < \frac{\epsilon}{\kappa}.$$

Then

$$\begin{aligned}
|F(f) - F(g)| &= \left| \kappa \int_{1-g}^{1-f} \mathscr{F}(f')\, df' \right| \\
&= \kappa \left| \int_{1-g}^{1-f} \mathscr{F}(f')\, df' \right| \\
&= \kappa \left| \int_{1-g}^{1-f} F(f')\, df' \right| \\
&\leq \kappa \, |f - g| \\
&< \epsilon.
\end{aligned}$$

This proves that F is continuous and we are now in a position to prove differentiability. Let f be a point in $[0,1]$ and let δ be a nonzero real number such that $f + \delta \in [0,1]$. Consider the quotient

$$\begin{aligned}
\frac{F(f+\delta) - F(f)}{\delta} &= \frac{\kappa \int_0^{1-(f+\delta)} \mathscr{F}(f')\, df' - \kappa \int_0^{1-f} \mathscr{F}(f')\, df'}{\delta} \\
&= \kappa \frac{\int_{1-f}^{1-(f+\delta)} \mathscr{F}(f')\, df'}{\delta} \\
&= \kappa \frac{\int_{1-f}^{1-(f+\delta)} F(f')\, df'}{\delta} \\
&= -\kappa \frac{\int_{1-(f+\delta)}^{1-f} F(f')\, df'}{\delta}.
\end{aligned}$$

Recall that F is a monotonically decreasing function. When δ is positive, the integral is bounded above and below by $\delta F(1-f)$ and $\delta F(1-(f+\delta))$ and the quotient by $-\kappa F(1-(f+\delta))$ and $-\kappa F(1-f)$ respectively. When δ is negative, the integral is bounded above and below by $\delta F(1-(f+\delta))$ and $\delta F(1-f)$ but still the quotient is bounded by $-\kappa F(1-(f+\delta))$ and $-\kappa F(1-f)$ respectively. Furthermore by the continuity of F, as δ tends to 0 the bounds of the quotient come together and the differential of F, which we shall write as F', is $-\kappa F(1-f)$.

Note that, since F is differentiable in $[0,1]$, $-\kappa F(1-f)$ is also differentiable. In other words, F is twice differentiable. Therefore

$$\begin{aligned}
-F'(1-f) &= \kappa F(f) \\
F'' &= \kappa F'(f) \\
&= -\kappa^2 F(1-f),
\end{aligned}$$

so $F''(f) = -\kappa^2 F(f)$. This gives us the auxiliary equation $\lambda^2 + \kappa^2 = 0$, which has roots $\lambda = \pm i\kappa$. Recall that κ is nonzero and so the general solution for $F(f)$ is $Ae^{i\kappa f} + Be^{-i\kappa f}$, for some constants A and B.

Of course, our function F is real and so, by equating imaginary parts, we have for any f

$$\begin{aligned} 0 &= \mathrm{Im}(Ae^{i\kappa f} + Be^{-i\kappa f}) \\ &= A\sin(\kappa f) + B\sin(-\kappa f) \\ &= (A - B)\sin(\kappa f)\,. \end{aligned}$$

This is only possible if $A = B$ since κ is positive and unchanged by our choice of f. We may now write

$$F(f) = 2A\cos(\kappa f)\,.$$

As mentioned earlier, $F(0) = 1$ by definition of F. This proves that $2A = 1$. Also from the definition of F we can see that $F(1) = 0$ and so $\cos\kappa = 0$. This means that $\kappa = n\pi + \frac{\pi}{2}$ for some integer n, so the family of possible solutions are

$$F(f) = \cos\bigl(\pi(n + \tfrac{1}{2})f\bigr)\,.$$

We know that $F(f)$ is monotonically decreasing over $[0, 1]$ and so cannot oscillate. When $n > 0$ we can see that the solutions oscillate (go from being positive to negative or viceversa) precisely n times and so $n \leq 0$.

For n negative let us write $m = -n$, then some simple trigonometry yields

$$\begin{aligned} F(f) &= \cos\bigl(\pi(n + \tfrac{1}{2})f\bigr) \\ &= \cos\bigl(\pi(-m + \tfrac{1}{2})f\bigr) \\ &= \cos\bigl(\pi(m - \tfrac{1}{2})f\bigr) \\ &= \cos\bigl(\pi((m-1) + \tfrac{1}{2})f\bigr) \end{aligned}$$

and so the solution given by a negative n is the same one as given by $-n - 1$, a nonnegative integer. Hence we discard solutions for negative n. This leaves only the solution given by $n = 0$ and so

$$F(f) = \cos\bigl(\tfrac{\pi}{2}f\bigr)\,.$$

Recall that F and $\mathscr{F}$ are equal almost everywhere and so our sequence of normalised polynomials $\left\{\tilde{F}_{n,1}(f)\right\}_{n=1}^{\infty}$ converges in $L^2[0, 1]$ to $\cos\bigl(\frac{\pi}{2}f\bigr)$. □

5.2.3 Harmonic analysis with a generalised Fourier series

Towards the end of the proof of Theorem 5.2.14 we came across the following family of functions as potential solutions for our limit:

$$\bigl\{\cos\bigl(\pi(n + \tfrac{1}{2})f\bigr) : n \text{ a nonnegative integer}\bigr\}\,.$$

These functions are special for two reasons: firstly, when operated upon by our recursive procedure for the polynomials $F_{n,1}(f)$ (as given in Lemma 5.1.8) the result is a scalar multiple of itself; secondly, they strongly resemble the $\cos(nf)$ functions from the Fourier series of harmonic analysis.

Lemma 5.2.15. *The set of functions*

$$\{\sin\big((n+\tfrac{1}{2})f\big)\,,\cos\big((n+\tfrac{1}{2})f\big) : n \in \mathbb{N}\}$$

is a set of orthogonal functions over $[-\pi, \pi]$.

Proof. Firstly, we shall show that $\big\|\sin\big((n+\frac{1}{2})f\big)\big\|$ and $\big\|\cos\,(n+\frac{1}{2})f\big\|$ are both nonzero.

$$\begin{aligned}
\big\|\sin\big((n+\tfrac{1}{2})f\big)\big\|^2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\big(\sin\big((n+\tfrac{1}{2})f\big)\big)^2\,df \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{1-\cos((2n+1)f)}{2}\,df \\
&= \frac{1}{2}.
\end{aligned}$$

$$\begin{aligned}
\big\|\cos\big((n+\tfrac{1}{2})f\big)\big\|^2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\big(\cos\big((n+\tfrac{1}{2})f\big)\big)^2\,df \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}1-\big(\sin\big((n+\tfrac{1}{2})f\big)\big)^2\,df \\
&= 1-\big\|\sin\big((n+\tfrac{1}{2})f\big)\big\|^2 \\
&= \frac{1}{2}.
\end{aligned}$$

Secondly, we must prove that the inner product of any two distinct functions from our set is zero. We break this part of the proof down into two parts: sines with sines and cosines with cosines; cosines with sines.

In order to do this we shall deduce some trigonometric identities. So, for no obvious reason consider the following integrals:

$$\begin{aligned}
\int_{-\pi}^{\pi}\cos((n+m+1)f)\,df &= \int_{-\pi}^{\pi}\cos((n+\tfrac{1}{2})f)\cos((m+\tfrac{1}{2})f)\,df \\
&\quad-\int_{-\pi}^{\pi}\sin((n+\tfrac{1}{2})f)\sin((m+\tfrac{1}{2})f)\,df; \\
\int_{-\pi}^{\pi}\cos((n-m)f)\,df &= \int_{-\pi}^{\pi}\cos(((n+\tfrac{1}{2})-(m+\tfrac{1}{2}))f)\,df \\
&= \int_{-\pi}^{\pi}\cos((n+\tfrac{1}{2})f)\cos((m+\tfrac{1}{2})f)\,df \\
&\quad+\int_{-\pi}^{\pi}\sin((n+\tfrac{1}{2})f)\sin((m+\tfrac{1}{2})f)\,df.
\end{aligned}$$

When n and m are different then

$$\int_{-\pi}^{\pi} \cos((n-m)f)\,df = 0.$$

Also

$$\int_{-\pi}^{\pi} \cos((n+m+1)f)\,df = 0$$

since $n+m+1 > 0$. Therefore

$$\int_{-\pi}^{\pi} \cos((n+\tfrac{1}{2})f)\cos((m+\tfrac{1}{2})f)\,df - \int_{-\pi}^{\pi} \sin((n+\tfrac{1}{2})f)\sin((m+\tfrac{1}{2})f)\,df$$

and

$$\int_{-\pi}^{\pi} \cos((n+\tfrac{1}{2})f)\cos((m+\tfrac{1}{2})f)\,df + \int_{-\pi}^{\pi} \sin((n+\tfrac{1}{2})f)\sin((m+\tfrac{1}{2})f)\,df$$

are both zero. This gives us

$$\int_{-\pi}^{\pi} \cos((n+\tfrac{1}{2})f)\cos((m+\tfrac{1}{2})f)\,df = 0$$

and

$$\int_{-\pi}^{\pi} \sin((n+\tfrac{1}{2})f)\sin((m+\tfrac{1}{2})f)\,df = 0.$$

Finally, notice that $\sin((m+\frac{1}{2})f)\cos((n+\frac{1}{2})f)$ is an odd function and so gives us 0 when integrated over an interval that is symmetric about the origin. □

5.2.4 Proving convergence of the recursive counting method

Larsen shows in [Lar73] that the set of functions $\{e^{inf} : n \in \mathbb{Z}\}$ is a complete orthonormal set in $L^2[-\pi,\pi]$. This together with Theorem 5.2.9 shows us that such a set is equivalently a basis. We shall manipulate this set of functions until we can show that the functions $\sin((n+\frac{1}{2})f)$ and $\cos((n+\frac{1}{2})f)$, for nonnegative n, also form a basis.

Theorem 5.2.16. *The set of functions $B = \{\sin((n+\frac{1}{2})f), \cos((n+\frac{1}{2})f) : n \in \mathbb{N}\}$ is a complete orthogonal set over $[-\pi,\pi]$.*

Proof. Firstly we will show that $S := \{e^{i(n+\frac{1}{2})f} : n \in \mathbb{Z}\}$ is a complete orthogonal set in $L^2[-\pi,\pi]$. Certainly S forms an orthogonal set, as

$$\left\langle e^{i(n+\frac{1}{2})f}, e^{i(m+\frac{1}{2})f} \right\rangle = 0 \text{ if and only if } n \neq m.$$

Let $F \in L^2[-\pi,\pi]$ be such that $\left\langle F, e^{i\left(n+\frac{1}{2}\right)f} \right\rangle = 0$ for all integers n. Consider the function $F(f)e^{-if/2}$.

$$\begin{aligned}
\left\langle F(f)e^{-if/2}, F(f)e^{-if/2}\right\rangle &= \int_{-\pi}^{\pi} F(f)e^{-if/2}\overline{F(f)e^{-if/2}}\, df \\
&= \int_{-\pi}^{\pi} F(f)e^{-if/2}\overline{F(f)}e^{if/2}\, df \\
&= \int_{-\pi}^{\pi} F(f)\overline{F(f)}\, df \\
&= \langle F, F\rangle
\end{aligned}$$

which is real since $F \in L^2[-\pi,\pi]$ and so $F(f)e^{-if/2} \in L^2[-\pi,\pi]$. Recall that $\left\langle F, e^{i\left(n+\frac{1}{2}\right)f}\right\rangle = 0$ means that $\int_{-\pi}^{\pi} F(f)\overline{e^{i\left(n+\frac{1}{2}\right)}}\, df = 0$, so

$$\begin{aligned}
\left\langle F(f)e^{-if/2}, e^{inf}\right\rangle &= \int_{-\pi}^{\pi} F(f)e^{-if/2}\overline{e^{inf}} \\
&= \int_{-\pi}^{\pi} F(f)e^{-if/2}e^{-inf} \\
&= \int_{-\pi}^{\pi} F(f)e^{-if\left(n+\frac{1}{2}\right)} \\
&= \int_{-\pi}^{\pi} F(f)\overline{e^{if\left(n+\frac{1}{2}\right)}} \\
&= 0.
\end{aligned}$$

Therefore $Fe^{-if/2}$ is orthogonal to e^{inf} for all integers n, but because $\{e^{inf} : n \in \mathbb{Z}\}$ is a maximal orthogonal set it must be the case that $F(f)e^{-if/2} =_{\text{ae}} 0$. The function $e^{-if/2}$ is never zero however, so our function F must be zero almost everywhere. This proves that S is also a complete orthogonal set.

Recall that in Lemma 5.2.15 we showed that B is an orthogonal set. Suppose that $G \in L^2[-\pi,\pi]$ is such that

$$\left\langle G(f), \cos\left(\left(n+\tfrac{1}{2}\right) f\right)\right\rangle = 0 \text{ and} \tag{5.1}$$
$$\left\langle G(f), \sin\left(\left(n+\tfrac{1}{2}\right) f\right)\right\rangle = 0 \tag{5.2}$$

for all nonnegative integers n. Then linear combinations of (5.1) and (5.2) are also zero. In particular

$$\begin{aligned}
0 &= \left\langle G(f), \cos\left(\left(n+\tfrac{1}{2}\right) f\right)\right\rangle - i\left\langle G(f), \sin\left(\left(n+\tfrac{1}{2}\right) f\right)\right\rangle \\
&= \left\langle G(f), \cos\left(\left(n+\tfrac{1}{2}\right) f\right) + i\sin\left(\left(n+\tfrac{1}{2}\right) f\right)\right\rangle \\
&= \left\langle G(f), e^{if\left(n+\frac{1}{2}\right)}\right\rangle
\end{aligned}$$

and

$$
\begin{aligned}
0 &= \left\langle G(f), \cos\left(\left(n+\tfrac{1}{2}\right)f\right)\right\rangle + i\left\langle G(f), \sin\left(\left(n+\tfrac{1}{2}\right)f\right)\right\rangle \\
&= \left\langle G(f), \cos\left(\left(n+\tfrac{1}{2}\right)f\right) - i\sin\left(\left(n+\tfrac{1}{2}\right)f\right)\right\rangle \\
&= \left\langle G(f), e^{-if\left(n+\frac{1}{2}\right)}\right\rangle \\
&= \left\langle G(f), e^{if\left(-n-\frac{1}{2}\right)}\right\rangle \\
&= \left\langle G(f), e^{if\left(-(n+1)+\frac{1}{2}\right)}\right\rangle .
\end{aligned}
$$

Therefore, since S is a complete orthogonal set and we have shown that G is orthogonal to every element of it, G must be zero almost everywhere and so the orthonormal set B is maximal. That is, B is a complete orthonormal set. □

Now that we have shown $\{\sin((n+\frac{1}{2})f), \cos((n+\frac{1}{2})f) : n \in \mathbb{N}\}$ is a complete orthogonal set in $L^2[-\pi,\pi]$, we appeal to [Car66] where Carleson proved that the Fourier series of a function in $L^2[-\pi,\pi]$ converges in $L^2[-\pi,\pi]$ to the function, i.e. the function and its Fourier series are equal almost everywhere.

The function $F_{1,1}(f)$ is only defined over the interval $[0,1]$ and our complete orthogonal system is over the interval $[-\pi,\pi]$. In order to overcome this, we shall look at generating the generalised Fourier series for the function 1 over the interval $[-\pi,\pi]$ and then shrink it to one over $[-1,1]$. When necessary, there will be an implicit restriction of this function to the interval $[0,1]$.

The function $1(f)$ is obviously in $L^2[-\pi,\pi]$, so there exists a generalised Fourier series for it using our basis $\{\sin((n+\frac{1}{2})f), \cos((n+\frac{1}{2})f) : n \in \mathbb{N}\}$. Hence we may write

$$
1 = \sum_{n=0}^{\infty} \left(\alpha_n \cos((n+\tfrac{1}{2})f) + \beta_n \sin((n+\tfrac{1}{2})f)\right).
$$

If we take the inner product of this function with the basis element $\cos((m+\frac{1}{2})f)$ say, we find

$$
\begin{aligned}
\langle 1, \cos((m+\tfrac{1}{2})f)\rangle &= \langle \sum_{n=0}^{\infty} \left(\alpha_n \cos((n+\tfrac{1}{2})f) + \beta_n \sin((n+\tfrac{1}{2})f)\right), \\
&\qquad \cos((m+\tfrac{1}{2})f)\rangle \\
&= \sum_{n=0}^{\infty} \alpha_n \langle \cos((n+\tfrac{1}{2})f), \cos((m+\tfrac{1}{2})f)\rangle \\
&\qquad + \sum_{n=0}^{\infty} \beta_n \langle \sin((n+\tfrac{1}{2})f), \cos((m+\tfrac{1}{2})f)\rangle \\
&= \alpha_m \langle \cos((m+\tfrac{1}{2})f), \cos((m+\tfrac{1}{2})f)\rangle.
\end{aligned}
$$

Hence we may determine the constant α_m:

$$\begin{aligned}
\alpha_m &= \frac{\langle 1, \cos((m+\frac{1}{2})f)\rangle}{\langle \cos((m+\frac{1}{2})f), \cos((m+\frac{1}{2})f)\rangle} \\
&= \frac{\int_{-\pi}^{\pi} \cos((m+\frac{1}{2})f)\,df}{\int_{-\pi}^{\pi} \cos^2((m+\frac{1}{2})f)\,df} \\
&= \frac{\left[\frac{\sin((m+\frac{1}{2})f)}{m+\frac{1}{2}}\right]_{-\pi}^{\pi}}{\int_{-\pi}^{\pi} \frac{\cos((2m+1)f)+1}{2}\,df} \\
&= \frac{2\sin((m+\frac{1}{2})\pi)}{(m+\frac{1}{2})\left[\frac{\sin((2m+1)f)}{2(2m+1)} + f/2\right]_{-\pi}^{\pi}} \\
&= \frac{2(-1)^m}{(m+\frac{1}{2})\pi}.
\end{aligned}$$

Note that the coefficients β_m are all zero as the function $1(f)$ is symmetric about the origin — this can also be seen by calculating the inner product of the function with our sine functions: $\int_{-\pi}^{\pi} \sin((m+\frac{1}{2})f)\,df = 0$.

Lemma 5.2.17. *We may write the function $1(f)$ over $[-1,1]$ as the sum*

$$\sum_{n=0}^{\infty} \frac{2(-1)^n \cos(\pi(n+\frac{1}{2})f)}{(n+\frac{1}{2})\pi}.$$

Proof. We have just shown the coefficients, α_n and β_n, of our generalised Fourier series over the interval $[-\pi,\pi]$ to be $\frac{2(-1)^n}{\pi(n+\frac{1}{2})}$ and zero respectively, so:

$$1 = \sum_{n=0}^{\infty} \frac{2(-1)^n \cos((n+\frac{1}{2})f)}{(n+\frac{1}{2})\pi}.$$

It follows that, shrinking the function 1 to the interval $[-1,1]$, we may write:

$$1 = \sum_{n=0}^{\infty} \frac{2(-1)^n \cos(\pi(n+\frac{1}{2})f)}{(n+\frac{1}{2})\pi}.$$

□

Lemma 5.2.18. *The function $F_{n,1}(f)$ has the following representation for any point f in $[0,1]$:*

$$F_{n,1}(f) = \sum_{m=0}^{\infty} \frac{2(-1)^{mn}}{(\pi(m+\frac{1}{2}))^n} \cos\big(\pi(m+\tfrac{1}{2})f\big).$$

Proof. Consider the following integral:

$$\begin{aligned}\int_0^{1-f} \cos\bigl(\pi(m+\tfrac{1}{2})f\bigr)\ df &= \left[\frac{\sin\bigl(\pi(m+\frac{1}{2})f\bigr)}{\pi(m+\frac{1}{2})}\right]_0^{1-f} \\ &= \frac{\sin\bigl(\pi(m+\frac{1}{2})(1-f)\bigr)}{\pi(m+\frac{1}{2})} \\ &= \frac{\sin\bigl(m\pi+\frac{\pi}{2}-\pi(m+\frac{1}{2})f\bigr)}{\pi(m+\frac{1}{2})} \\ &= \frac{(-1)^m\sin\bigl(\frac{\pi}{2}-\pi(m+\frac{1}{2})f\bigr)}{\pi(m+\frac{1}{2})} \\ &= \frac{(-1)^m}{\pi(m+\frac{1}{2})}\cos\bigl(\pi(m+\tfrac{1}{2})f\bigr)\,.\end{aligned}$$

From Lemma 5.2.17 we may write:

$$F_{1,1}(f) = \sum_{m=0}^{\infty} \frac{2(-1)^m}{\pi(m+\frac{1}{2})}\cos\bigl(\pi(m+\tfrac{1}{2})f\bigr)\,.$$

Since we perform an integration, from 0 to $1-f$, a total of $n-1$ times to generate $F_{n,1}(f)$ from $1(f)$, we obtain

$$\begin{aligned}F_{n,1}(f) &= \sum_{m=0}^{\infty} \frac{2(-1)^m}{\pi(m+\frac{1}{2})}\left(\frac{(-1)^m}{\pi(m+\frac{1}{2})}\right)^{n-1}\cos\bigl(\pi(m+\tfrac{1}{2})f\bigr) \\ &= \sum_{m=0}^{\infty} \frac{2(-1)^{mn}}{\bigl(\pi(m+\frac{1}{2})\bigr)^n}\cos\bigl(\pi(m+\tfrac{1}{2})f\bigr)\,.\end{aligned}$$

□

Theorem 5.2.19. *The normalised sequence of polynomials* $\{\tilde{F}_{n,1}(f)\}_{n=1}^{\infty}$ *completely converges to* $\cos(\frac{\pi f}{2})$.

Proof. From Lemma 5.2.18 we know that

$$F_{n,1}(f) = \sum_{m=0}^{\infty} \frac{2(-1)^{mn}}{(\pi(m+\frac{1}{2}))^n}\cos\bigl(\pi(m+\tfrac{1}{2})f\bigr)\,.$$

Recall from Definition 5.1.9 the definition of our normalised polynomial is

$$\tilde{F}_{n,1}(f) := \frac{F_{n,1}(f)}{F_{n,1}(0)}.$$

Let us define another type of *normalised* polynomial, $\hat{F}_{n,1}(f)$, to be

$$\hat{F}_{n,1}(f) := \frac{\pi^n}{2^{n+1}}F_{n,1}(f).$$

Firstly, we shall show that $\{\hat{F}_{n,1}(f)\}_{n=1}^{\infty}$ completely converges to $\cos(\frac{\pi f}{2})$. Then we show that $\{\hat{F}_{n,1}(f) - \tilde{F}_{n,1}(f)\}_{n=1}^{\infty}$ completely converges to zero and the theorem will follow.

Let n be at least 2. Then

$$\begin{aligned}
\left|\hat{F}_{n,1}(f) - \cos(\frac{\pi}{2}f)\right| &= \left|\frac{\pi^n}{2^{n+1}}\sum_{m=0}^{\infty}\frac{2(-1)^{mn}}{(\pi(m+\frac{1}{2}))^n}\cos\big(\pi(m+\tfrac{1}{2})f\big) - \cos(\frac{\pi}{2}f)\right| \\
&= \left|\frac{1}{2^n}\sum_{m=0}^{\infty}\frac{(-1)^{mn}}{(m+\frac{1}{2})^n}\cos\big(\pi(m+\tfrac{1}{2})f\big) - \cos(\frac{\pi}{2}f)\right| \\
&= \left|2^{-n}\sum_{m=1}^{\infty}\frac{(-1)^{mn}}{(m+\frac{1}{2})^n}\cos\big(\pi(m+\tfrac{1}{2})f\big)\right| \\
&\le 2^{-n}\sum_{m=1}^{\infty}\left|\frac{(-1)^{mn}}{(m+\frac{1}{2})^n}\cos\big(\pi(m+\tfrac{1}{2})f\big)\right| \\
&\le 2^{-n}\sum_{m=1}^{\infty}(m+\tfrac{1}{2})^{-n}.
\end{aligned}$$

If we can show that $\sum_{m=1}^{\infty}(m+\frac{1}{2})^{-n}$ is bounded, we will have shown that $\{\hat{F}_{n,1}(f)\}_{n=1}^{\infty}$ completely converges to $\cos(\frac{\pi f}{2})$.

Now consider the difference between the two types of normalised polynomials. Again let n be at least 2. Then

$$\begin{aligned}
\left|\hat{F}_{n,1}(f) - \tilde{F}_{n,1}(f)\right| &= \left|\frac{F_{n,1}(f)}{\frac{2^{n+1}}{\pi^n}} - \frac{F_{n,1}(f)}{F_{n,1}(0)}\right| \\
&= \left|F_{n,1}(f)\frac{F_{n,1}(0) - \frac{2^{n+1}}{\pi^n}}{\frac{2^{n+1}}{\pi^n}F_{n,1}(0)}\right| \\
&= \frac{\pi^n}{2^{n+1}}\left|\frac{F_{n,1}(f)}{F_{n,1}(0)}\right|\left|\sum_{m=0}^{\infty}\frac{2(-1)^{mn}}{(\pi(m+\frac{1}{2}))^n} - \frac{2^{n+1}}{\pi^n}\right| \\
&= \frac{\pi^n}{2^n}\left|\tilde{F}_{n,1}(f)\right|\left|\sum_{m=1}^{\infty}\frac{(-1)^{mn}}{(\pi(m+\frac{1}{2}))^n}\right| \\
&\le 2^{-n}\left|\tilde{F}_{n,1}(f)\right|\sum_{m=1}^{\infty}\left|\frac{(-1)^{mn}}{(m+\frac{1}{2})^n}\right| \\
&= 2^{-n}\left|\tilde{F}_{n,1}(f)\right|\sum_{m=1}^{\infty}(m+\tfrac{1}{2})^{-n}.
\end{aligned}$$

Recall that Lemma 5.2.10 and Lemma 5.2.11 tell us, for $n > 1$, the $F_{n,1}(f)$ are nonnegative and strictly decreasing. Therefore $0 \le \tilde{F}_{n,1}(f) \le 1$ and so

$$\left|\hat{F}_{n,1}(f) - \tilde{F}_{n,1}(f)\right| \le 2^{-n}\sum_{m=1}^{\infty}(m+\tfrac{1}{2})^{-n}.$$

To prove the result then, it suffices to show that $\sum_{m=1}^{\infty}(m+\frac{1}{2})^{-n}$ is bounded.

Clearly

$$(m + \tfrac{1}{2})^{-n} > (m + \tfrac{1}{2})^{-(n+1)},$$

so

$$\sum_{m=1}^{\infty} (m + \tfrac{1}{2})^{-n} > \sum_{m=1}^{\infty} (m + \tfrac{1}{2})^{-(n+1)}.$$

Therefore we only need that $\sum_{m=1}^{\infty}(m + \frac{1}{2})^{-2}$ exists, since it is also clear that $\sum_{m=1}^{\infty}(m + \frac{1}{2})^{-n} > 0$. Dunham gives Euler's proof of $\sum_{m=1}^{\infty}(m + \frac{1}{2})^{-2} = \frac{\pi^2}{6}$ in [Dun99].

$\square$

Chapter 6

Bounding the size of Pebody codes

So far we have methods for obtaining the polynomial $P_{n,1}(b, c)$, which counts the number of words in $A_{n,b}$ with first letter c. Setting $f = c/b$ so that our polynomial becomes $P_{n,1}(b, bf)$ provides us with a way of "isolating" the most significant term in the polynomial, $F_{n,1}(f)b^{n-1}$. This makes it possible for us to maximise and bound expressions such as $P_{n,i}(b, c) + O(b^{n-i-1})$.

Recall that at the end of Chapter 4 we said that the difference in the b^{n-2} term between the sizes of the Peboy code and the alternating code is

$$\sum_{k=1}^{n-1} C_{n,k}(f_k)b^{n-2}.$$

At the start of the following chapter we translated this notation from the language of Chapter 4 to what we currently use. We now express this difference as

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)b^{n-2}.$$

By the end of this chapter we shall have upper bounds, lower bounds and approximations for the product $F_{-1,-i}(f)F_{n,i+1}(f)$ so that we can estimate how much larger the Pebody code is than the alternating code.

6.1 Approximating the number of words with a single repeat

6.1.1 Approximating the recursive counting method by its limit

Shortly, we shall give a way of approximating the polynomials $F_{n,1}(f)$ and the errors will be given in terms of the Hurwitz zeta function. Lang tells us that the Hurwitz zeta function as it is defined here is absolutely convergent in [Lan99].

Definition 6.1.1. *Let s and a be complex numbers such that $Re(s) > 1$ and $Re(a) > 0$. Define the Hurwitz zeta function thusly:*

$$\zeta(s, a) := \sum_{m=0}^{\infty} (a + m)^{-s}.$$

Except for the case where $n = 1$, we may approximate the function $F_{n,1}(f)$ by $\frac{2^{n+1}}{\pi^n} \cos\left(\frac{\pi}{2} f\right)$ as we show below. For the case $n = 1$, recall that by definition $F_{1,1} = 1$.

Lemma 6.1.2. *Let f be in $[0, 1]$ and let n be an integer such that $n > 1$. The following inequality then holds:*

$$\left| F_{n,1}(f) - \frac{2^{n+1} \cos\left(\frac{\pi f}{2}\right)}{\pi^n} \right| \leq \frac{2\zeta(n, \frac{3}{2})}{\pi^n}.$$

Proof. From Lemma 5.2.18 we have the following representation for $F_{n,1}(f)$:

$$F_{n,1}(f) = \sum_{m=0}^{\infty} \frac{2(-1)^{mn}}{(\pi(m + \frac{1}{2}))^n} \cos\left(\pi(m + \tfrac{1}{2})f\right).$$

Thus we may bound the error in our approximation as so:

$$\begin{aligned}
\left| F_{n,1}(f) - \frac{2^{n+1} \cos\left(\frac{\pi f}{2}\right)}{\pi^n} \right| &= \left| \sum_{m=1}^{\infty} \frac{2(-1)^{mn}}{(\pi(m + \frac{1}{2}))^n} \cos\left(\pi(m + \tfrac{1}{2})f\right) \right| \\
&\leq \frac{2}{\pi^n} \sum_{m=1}^{\infty} \left| \frac{(-1)^{mn}}{(m + \frac{1}{2})^n} \cos\left(\pi(m + \tfrac{1}{2})f\right) \right| \\
&\leq \frac{2}{\pi^n} \sum_{m=1}^{\infty} (m + \tfrac{1}{2})^{-n} \\
&\leq \frac{2}{\pi^n} \sum_{m=0}^{\infty} (m + \tfrac{3}{2})^{-n} \\
&= \frac{2\zeta(n, \frac{3}{2})}{\pi^n}.
\end{aligned}$$

□

Definition 6.1.3. *Let s be a complex number such that $Re(s) > 1$. The Riemann zeta function $\zeta(s)$ is defined by*

$$\zeta(s) := \sum_{k=1}^{\infty} k^{-s}.$$

The Hurwitz zeta function is a generalisation of the Riemann zeta function $\zeta(s)$.

This is clear to see with a relabelling of the dummy variables $k = m + 1$.

$$\begin{aligned}
\zeta(s, 1) &= \sum_{m=0}^{\infty} (1 + m)^{-s} \\
&= \sum_{k=1}^{\infty} k^{-s} \\
&= \zeta(s).
\end{aligned}$$

The absolute convergence of the Hurwitz zeta function is therefore enough to reassure us that the Riemann zeta function is likewise convergent. There is a further link between the two zeta functions that we shall need to exploit for the case where $a = \frac{1}{2}$. This is given in the following lemma.

Lemma 6.1.4. *Let s again be a complex number with real component larger than 1. Then*

$$\zeta(s, \tfrac{3}{2}) = (2^s - 1)\,\zeta(s) - 2^s.$$

Proof. A small amount of manipulation on the definition yields the result:

$$\begin{aligned}
\zeta(s, \tfrac{3}{2}) &= \sum_{m=0}^{\infty} (m + \tfrac{3}{2})^{-s} \\
&= \sum_{m=1}^{\infty} (m + \tfrac{1}{2})^{-s} \\
&= \sum_{m=0}^{\infty} (m + \tfrac{1}{2})^{-s} - 2^{-s} \\
&= 2^s \sum_{m=0}^{\infty} (2m + 1)^{-s} - 2^{-s} \\
&= 2^s \left(\sum_{k=1}^{\infty} k^{-s} - \sum_{k=1}^{\infty} (2k)^{-s} \right) - 2^{-s} \\
&= 2^s \left(\zeta(s) - 2^{-s}\zeta(s) \right) - 2^{-s} \\
&= (2^s - 1)\,\zeta(s) - 2^{-s}.
\end{aligned}$$

□

Although Lemma 6.1.4 provides us with another way of thinking about $\zeta(s, \frac{3}{2})$, any feeling for how the function behaves, say for large integers s, remains somewhat intangible. We shall only be interested in the case where s is a positive integer larger than 2 and so this enables us to form a simple bound.

Lemma 6.1.5. *Let n be an integer greater than 1. Then we may bound $\zeta(n, \frac{3}{2})$ above by $\left(\frac{2}{3}\right)^{n-2} \zeta(2, \frac{3}{2})$.*

Proof. Trivially, the bound holds for the case where $n = 2$. Assume then that $n > 2$.

Recall the definition of the Hurwitz zeta function, then

$$\zeta(n, \tfrac{3}{2}) = \sum_{m=0}^{\infty} \left(m + \tfrac{3}{2}\right)^{-n}.$$

For each value the dummy variable m takes, the summand satisfies $m + \frac{3}{2} \geq \frac{3}{2}$. Hence $\left(m + \frac{3}{2}\right)^{-1} \leq \frac{2}{3}$ and so

$$\sum_{m=0}^{\infty} \left(m + \tfrac{3}{2}\right)^{-n} \leq \sum_{m=0}^{\infty} \tfrac{2}{3} \left(m + \tfrac{3}{2}\right)^{-n+1}.$$

Therefore $\zeta(n, \frac{3}{2}) \leq \frac{2}{3}\zeta(n-1, \frac{3}{2})$. If we exploit this inequality $n-2$ times the result follows. □

Lemma 6.1.6. [*The Basel problem*] *The Riemann zeta function evaluated at* 2 *is*

$$\sum_{m=0}^{\infty} (1+m)^{-2} = \tfrac{\pi^2}{6}.$$

Proof. In [Dun99], Dunham provides us with detailed accounts of both Euler's first, though somewhat questionable, proof and his alternative which is fully accepted by the mathematical community. □

The following corollary simply ties together some of the above results in order to provide us with a palpable analytic bound for the error in approximation introduced in Lemma 6.1.2.

Corollary 6.1.7. *Let n be an integer greater than* 1 *and f a point of the interval* $[0, 1]$. *The error in approximating $F_{n,1}(f)$ by $\frac{2^{n+1}}{\pi^n} \cos\left(\frac{\pi}{2} f\right)$ is as such:*

$$\left| F_{n,1}(f) - \frac{2^{n+1} \cos\left(\frac{\pi f}{2}\right)}{\pi^n} \right| \leq \left(\frac{2}{3\pi}\right)^{n-2} \frac{\pi^2 - 8}{\pi^2}.$$

Proof. Recall that Lemma 6.1.6 tells us that $\zeta(2) = \frac{\pi^2}{6}$. Let us apply Lemma 6.1.5

and then Lemma 6.1.4 to the inequality derived in Lemma 6.1.2.

$$\begin{aligned}\left|F_{n,1}(f)-\frac{2^{n+1}\cos\left(\frac{\pi f}{2}\right)}{\pi^n}\right| &\leq \frac{2\zeta(n,\frac{3}{2})}{\pi^n}\\ &\leq \left(\frac{2}{3}\right)^{n-2}\frac{2\zeta(2,\frac{3}{2})}{\pi^n}\\ &= \left(\frac{2}{3}\right)^{n-2}\frac{2(2^2-1)\zeta(2)-2^3}{\pi^n}\\ &= \left(\frac{2}{3\pi}\right)^{n-2}\frac{6\zeta(2)-8}{\pi^2}\\ &= \left(\frac{2}{3\pi}\right)^{n-2}\frac{6\frac{\pi^2}{6}-8}{\pi^2}\\ &= \left(\frac{2}{3\pi}\right)^{n-2}\frac{\pi^2-8}{\pi^2}.\end{aligned}$$

□

6.1.2 Approximating the number of words with one repeat at a given position

In this section we shall take the approximation proved in Corollary 6.1.7 and use it to derive a similar approximation for the number of words with one repeat at a given position. Firstly, however, we will need to generalise our approximation to any polynomial $F_{n,i}(f)$. In order to facilitate this process, we introduce the following notation.

Definition 6.1.8. *Let i be an integer. Define $\mathscr{F}_i(f)$ over $[0,1]$ by*

$$\begin{aligned}\mathscr{F}_i(f) &:= \cos\left(\tfrac{\pi f}{2}\right) && \text{when } i \text{ is odd,}\\ \mathscr{F}_i(f) &:= \cos\left(\tfrac{\pi(1-f)}{2}\right) && \text{when } i \text{ is even.}\end{aligned}$$

Lemma 6.1.9. *Let n and i be integers such that $n > i$ and let f be a point in the interval $[0,1]$. Then*

$$\left|F_{n,i}(f)-\frac{2^{n-i+2}\mathscr{F}_i(f)}{\pi^{n-i+1}}\right| \leq \left(\frac{2}{3\pi}\right)^{n-i-1}\frac{\pi^2-8}{\pi^2}.$$

Proof. If i is odd then define $g := f$ and if i is even then define $g := 1-f$. In both cases g belongs to the interval $[0,1]$. Recall that repeated applications of Theorem 5.1.7 yields

$$F_{n-i+1,1}(g) = F_{n,i}(f).$$

By definition it is clear that $\mathscr{F}_i(f) = \cos\left(\frac{\pi g}{2}\right)$. This information together with

Corollary 6.1.7 tells us that

$$\begin{aligned}\left|F_{n,i}(f) - \frac{2^{n-i+2}\mathscr{F}_i(f)}{\pi^{n-i+1}}\right| &= \left|F_{n-i+1,1}(g) - \frac{2^{n-i+2}\cos\left(\frac{\pi g}{2}\right)}{\pi^{n-i+1}}\right| \\ &\leq \left(\frac{2}{3\pi}\right)^{n-i-1}\frac{\pi^2-8}{\pi^2}.\end{aligned}$$

□

We shall have the need for bounds on the sizes of the functions $\mathscr{F}_i$ and $F_{n,i}$ over our interval $[0,1]$. From its definition we can see that $\mathscr{F}_i$ is nonnegative and at most 1. Recall Lemma 5.2.10 states $F_{n,i}$ is nonnegative over $[0,1]$.

Lemma 6.1.10. *Let n and i be integers with $n > i$ then*

$$F_{n,i}(f) \leq \frac{2^{n-i+2}}{\pi^{n-i+1}}\left(1 + \frac{\pi^2-8}{8\times 3^{n-i-1}}\right).$$

Proof. Recall that in Lemma 5.2.18 we showed that for any positive integer N

$$F_{N,1}(f) = \sum_{m=0}^{\infty}\frac{2(-1)^{mN}}{(\pi(m+\frac{1}{2}))^N}\cos\left(\pi(m+\tfrac{1}{2})f\right).$$

Again let us define g to be f when i is odd and $1-f$ when i is even. Then

$$\begin{aligned}F_{n,i}(f) &= F_{n-i+1,1}(g) \\ &= \sum_{m=0}^{\infty}\frac{2(-1)^{m(n-i+1)}}{(\pi(m+\frac{1}{2}))^{n-i+1}}\cos\left(\pi(m+\tfrac{1}{2})g\right) \\ &\leq \sum_{m=0}^{\infty}\left|\frac{2(-1)^{m(n-i+1)}}{(\pi(m+\frac{1}{2}))^{n-i+1}}\cos\left(\pi(m+\tfrac{1}{2})g\right)\right| \\ &\leq \frac{2}{\pi^{n-i+1}}\sum_{m=0}^{\infty}(m+\tfrac{1}{2})^{-n+i-1} \\ &= \frac{2}{\pi^{n-i+1}}\left(2^{n-i+1} + \sum_{m=1}^{\infty}(m+\tfrac{1}{2})^{-n+i-1}\right) \\ &= \frac{2}{\pi^{n-i+1}}\left(2^{n-i+1} + \zeta(n-i+1,\tfrac{3}{2})\right).\end{aligned}$$

Recall that in the proof of Corollary 6.1.7 we showed, for any integer m larger than 1, that

$$\frac{2\zeta(m,\frac{3}{2})}{\pi^m} \leq \left(\frac{2}{3\pi}\right)^{m-2}\frac{\pi^2-8}{\pi^2}.$$

Since $n > i$, we have that $n - i + 1$ is larger than 1. So

$$\begin{aligned} F_{n,i}(f) &\leq \frac{2^{n-i+2}}{\pi^{n-i+1}} + \left(\frac{2}{3\pi}\right)^{n-i-1} \frac{\pi^2 - 8}{\pi^2} \\ &= \frac{2^{n-i+2}}{\pi^{n-i+1}} \left(1 + \frac{\pi^2 - 8}{8 \times 3^{n-i-1}}\right). \end{aligned}$$

□

Recall that towards the end of Chapter 4, in the proof of Theorem 4.2.9, we employed the product $P_{-1,-i}(b, y_i)P_{n,i+1}(b, y_i)$ to count the number of words in B_y with a single repeat at position i, which is a polynomial. We shall soon be requiring estimates for the product $F_{-1,-i}(f)F_{n,i+1}(f)$ when we talk more about the leading term of this polynomial.

Although we have somewhat optimised the calculation of the polynomials $F_{n,i}$, generating them is computationally expensive. So, we now extend the approximation given in Lemma 6.1.9 to an approximation for the product $F_{-1,-i}(f)F_{n,i+1}(f)$.

Theorem 6.1.11. *Let n and i be integers with $n - i > 1$ and $i > 1$. Let f be a point in $[0, 1]$. The polynomial $F_{-1,-i}(f)F_{n,i+1}(f)$ can be approximated by*

$$\frac{2^{n+1}}{\pi^n} \sin(\pi f)$$

and the error in approximating is at most

$$\frac{2^{n-1}(\pi^2 - 8)}{\pi^n 3^{n-4}} \left(3^{i-2} + \frac{\pi^2 - 8}{8} + 3^{n-i-2}\right).$$

Proof. Let a_1, a_2, α_1 and α_2 be reals such that

$$|a_1 - \alpha_1| \leq \epsilon_1 \text{ and} \tag{6.1}$$

$$|a_2 - \alpha_2| \leq \epsilon_2 \tag{6.2}$$

for some reals ϵ_1 and ϵ_2. Then

$$\begin{aligned} |a_1 a_2 - \alpha_1 \alpha_2| &= |a_1 a_2 - a_1 \alpha_2 + a_1 \alpha_2 - \alpha_1 \alpha_2| \\ &\leq |a_1 a_2 - a_1 \alpha_2| + |a_1 \alpha_2 - \alpha_1 \alpha_2| \\ &= |a_1||a_2 - \alpha_2| + |\alpha_2||a_1 - \alpha_1| \\ &\leq |a_1|\epsilon_2 + |\alpha_2|\epsilon_1. \end{aligned} \tag{6.3}$$

Lemma 6.1.9 gives us the following two inequalities:

$$\left|F_{-1,-i}(f) - \frac{2^{i+1}\mathscr{F}_{-i}(f)}{\pi^i}\right| \le \left(\frac{2}{3\pi}\right)^{i-2} \frac{\pi^2-8}{\pi^2} \text{ and} \tag{6.4}$$

$$\left|F_{n,i+1}(f) - \frac{2^{n-i+1}\mathscr{F}_{i+1}(f)}{\pi^{n-i}}\right| \le \left(\frac{2}{3\pi}\right)^{n-i-2} \frac{\pi^2-8}{\pi^2}. \tag{6.5}$$

In the same way we derived (6.3) from (6.1) and (6.2), we derive from (6.4) and (6.5) the following:

$$\begin{aligned}
&\left|F_{-1,-i}(f)F_{n,i+1}(f) - \frac{2^{n+2}\mathscr{F}_{-i}(f)\mathscr{F}_{i+1}(f)}{\pi^n}\right| \\
&\le F_{-1,-i}(f)\left(\frac{2}{3\pi}\right)^{n-i-2}\frac{\pi^2-8}{\pi^2} + \frac{2^{n-i+1}\mathscr{F}_{i+1}(f)}{\pi^{n-i}}\left(\frac{2}{3\pi}\right)^{i-2}\frac{\pi^2-8}{\pi^2}.
\end{aligned} \tag{6.6}$$

Recall that Lemma 6.1.10 tells us

$$F_{-1,-i}(f) \le \frac{2^{i+1}}{\pi^i}\left(1 + \frac{\pi^2-8}{8\times 3^{i-2}}\right).$$

Bringing these last two inequalities together (along with the observation made earlier that $\mathscr{F}_{i+1} < 1$) we see that (6.6) is

$$\begin{aligned}
&\le \frac{2^{i+1}}{\pi^i}\left(1 + \frac{\pi^2-8}{8\times 3^{i-2}}\right)\left(\frac{2}{3\pi}\right)^{n-i-2}\frac{\pi^2-8}{\pi^2} \\
&\quad + \frac{2^{n-i+1}\mathscr{F}_{i+1}(f)}{\pi^{n-i}}\left(\frac{2}{3\pi}\right)^{i-2}\frac{\pi^2-8}{\pi^2} \\
&\le \frac{2^{n-1}}{\pi^n}\left(1 + \frac{\pi^2-8}{8\times 3^{i-2}}\right)\frac{\pi^2-8}{3^{n-i-2}} + \frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{i-2}} \\
&= \frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(3^{i-2} + \frac{\pi^2-8}{8} + 3^{n-i-2}\right).
\end{aligned}$$

Now, certainly $-i$ and $i+1$ are of opposite parity, so

$$\begin{aligned}
\mathscr{F}_{-i}(f)\mathscr{F}_{i+1}(f) &= \cos\left(\tfrac{\pi f}{2}\right)\cos\left(\tfrac{\pi(1-f)}{2}\right) \\
&= \cos\left(\tfrac{\pi f}{2}\right)\sin\left(\tfrac{\pi f}{2}\right) \\
&= \frac{\sin(\pi f)}{2}.
\end{aligned}$$

This observation then gives us that

$$\frac{2^{n+2}\mathscr{F}_{-i}(f)\mathscr{F}_{i+1}(f)}{\pi^n} = \frac{2^{n+1}}{\pi^n}\sin(\pi f)$$

and so the result follows. □

The error bound given in Theorem 6.1.11 depends only on n and i and the approximation is best when i is close to $\frac{n}{2}$. However, when i is close to 1 or n the approximation is poor and so in these cases we require a better method. This motivates the rest of this chapter.

6.2 A better approximation

6.2.1 Centralising the use of the approximation

Consider the difference in the b^{n-2} terms of the Pebody code and the alternating code as discussed at the start of this chapter:

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i). \tag{6.7}$$

It is possible to exactly determine this value, since we have already given an efficient way of calculating the $F_{n,i}(f)$. However, as we have already pointed out, this is still computationally expensive. Here we look at using the various approximations given earlier in this chapter to efficiently obtain bounds on this difference for some fixed tolerance.

Let l be an integer between 1 and $n-1$ inclusively. We consider the above sum term by term and approximate $F_{m,j}(f)$ when $m-j \geq l$. Depending on i and the choice of l we may approximate $F_{-1,-i}$, $F_{n,i+1}$, the product of these polynomials or we may not approximate the ith term at all. The following definition gives us the function $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ with which we shall approximate (6.7).

Definition 6.2.1. *Let n and l be positive integers so that $1 \leq l \leq n-1$. Define $R_{n,n-1}(f_1, f_2, \ldots, f_{n-1})$ as:*

$$R_{n,n-1} := \sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i).$$

Suppose $\frac{n-1}{2} < l < n-1$, then define $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ to be

$$\begin{aligned} R_{n,l} = \sum_{i=1}^{n-l-1} & F_{-1,-i}(f_i)\frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \\ &+ \sum_{i=n-l}^{l} F_{-1,-i}(f_i)F_{n,i+1}(f_i) \\ &+ \sum_{i=l+1}^{n-1} \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^i}F_{n,i+1}(f_i). \end{aligned}$$

Note that, since $l > n-l-1$ we have that $l \geq n-l$ and so the limits of the sums (in particular the central sum) are in the usual order of size.

Suppose now that $l < \frac{n-1}{2}$. Define then $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ to be

$$\begin{aligned} R_{n,l} =& \sum_{i=1}^{l} F_{-1,-i}(f_i) \frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \\ &+ \sum_{i=l+1}^{n-l-1} \frac{2^{n+1}}{\pi^n} \sin(\pi f_i) \\ &+ \sum_{i=n-l}^{n-1} \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^i} F_{n,i+1}(f_i). \end{aligned}$$

Again note that the limits of the sums are in the usual order.

Finally, suppose that $l = \frac{n-1}{2}$. Here $l = n-l-1$ so then we define

$$\begin{aligned} R_{n,l} :=& \sum_{i=1}^{n-l-1} F_{-1,-i}(f_i) \frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \\ &+ \sum_{i=l+1}^{n-1} \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^i} F_{n,i+1}(f_i). \end{aligned}$$

Lemma 6.2.2. *Let $l \geq \frac{n-1}{2}$. Then the error in approximating the sum*

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i) F_{n,i+1}(f_i)$$

by the function $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ is at most

$$\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}} \left((n-l-1)\frac{\pi^2-8}{4} + 3^{n-l-2} - \tfrac{1}{3} \right).$$

Proof. Evaluating the above formula at $l = n-1$ gives us 0. By definition

$$R_{n,n-1}(f_1, f_2, \ldots, f_{n-1}) = \sum_{i=1}^{n-1} F_{-1,-i}(f_i) F_{n,i+1}(f_i)$$

so there is no error in approximating in this case and the result is trivially true for $l = n-1$. For the rest of this proof we shall assume then that $l < n-1$.

Using the triangle inequality we can see that

$$\left| \sum_{i=1}^{n-1} F_{-1,-i}(f_i) F_{n,i+1}(f_i) - R_{n,l}(f_1, f_2, \ldots, f_{n-1}) \right| \tag{6.8}$$

is at most

$$\sum_{i=1}^{n-l-1} F_{-1,-i}(f_i) \left| F_{n,i+1}(f_i) - \frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \right|$$
$$+ \sum_{i=n-l}^{l} |F_{-1,-i}(f_i)F_{n,i+1}(f_i) - F_{-1,-i}(f_i)F_{n,i+1}(f_i)|$$
$$+ \sum_{i=l+1}^{n-1} F_{n,i+1}(f_i) \left| F_{-1,-i}(f_i) - \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^{i}} \right|$$

when $l > \frac{n-1}{2}$ and at most

$$\sum_{i=1}^{n-l-1} F_{-1,-i}(f_i) \left| F_{n,i+1}(f_i) - \frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \right|$$
$$+ \sum_{i=l+1}^{n-1} F_{n,i+1}(f_i) \left| F_{-1,-i}(f_i) - \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^{i}} \right|$$

when $l = \frac{n-1}{2}$. Recall that Lemma 6.1.9 tells us that

$$\left| F_{n,i}(f) - \frac{2^{n-i+2}\mathscr{F}_i(f)}{\pi^{n-i+1}} \right| \leq \left(\frac{2}{3\pi}\right)^{n-i-1} \frac{\pi^2 - 8}{\pi^2}.$$

Then (6.8) is at most

$$\sum_{i=1}^{n-l-1} F_{-1,-i}(f_i) \left(\frac{2}{3\pi}\right)^{n-i-2} \frac{\pi^2 - 8}{\pi^2}$$
$$+ \sum_{i=l+1}^{n-1} F_{n,i+1}(f_i) \left(\frac{2}{3\pi}\right)^{i-2} \frac{\pi^2 - 8}{\pi^2}.$$

Lemma 6.1.10 gives us

$$F_{n,i}(f) \leq \frac{2^{n-i+2}}{\pi^{n-i+1}} \left(1 + \frac{\pi^2 - 8}{8 \times 3^{n-i-1}}\right),$$

so (6.8) is at most

$$
\begin{aligned}
&\sum_{i=1}^{n-l-1} \frac{2^{i+1}}{\pi^i}\left(1+\frac{\pi^2-8}{8\times 3^{i-2}}\right)\left(\frac{2}{3\pi}\right)^{n-i-2}\frac{\pi^2-8}{\pi^2} \\
&+\sum_{i=l+1}^{n-1} \frac{2^{n-i+1}}{\pi^{n-i}}\left(1+\frac{\pi^2-8}{8\times 3^{n-i-2}}\right)\left(\frac{2}{3\pi}\right)^{i-2}\frac{\pi^2-8}{\pi^2} \\
=&\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(\sum_{i=1}^{n-l-1}\left(3^{i-2}+\frac{\pi^2-8}{8}\right)+\sum_{i=l+1}^{n-1}\left(3^{n-i-2}+\frac{\pi^2-8}{8}\right)\right) \\
=&\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(\frac{\pi^2-8}{8}2(n-l-1)+\sum_{i=1}^{n-l-1}3^{i-2}+\sum_{i=l+1}^{n-1}3^{n-i-2}\right).
\end{aligned}
$$

By a change of dummy variable ($j = n-i$) we can see that the two sums $\sum_{i=1}^{n-l-1} 3^{i-2}$ and $\sum_{j=l+1}^{n-1} 3^{n-j-2}$ are equal. Summing the geometric series we have that

$$\sum_{i=1}^{n-l-1} 3^{i-2} = \frac{3^{n-l-2}-\frac{1}{3}}{3-1} = \frac{3^{n-l-2}-\frac{1}{3}}{2},$$

and so (6.8) is at most

$$\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left((n-l-1)\frac{\pi^2-8}{4}+3^{n-l-2}-\tfrac{1}{3}\right).$$

□

Lemma 6.2.3. *Let $l < \frac{n-1}{2}$. Then the error in approximating the sum*

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)$$

by the function $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ is at most

$$\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left((n-1)\frac{\pi^2-8}{8}+3^{n-l-2}-\tfrac{1}{3}\right).$$

Proof. Using the triangle inequality we can see that

$$\left|\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i) - R_{n,l}(f_1, f_2, \ldots, f_{n-1})\right| \tag{6.9}$$

is at most

$$\begin{aligned}
&\sum_{i=1}^{l} F_{-1,-i}(f_i) \left| F_{n,i+1}(f_i) - \frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \right| \\
&+ \sum_{i=l+1}^{n-l-1} \left| F_{-1,-i}(f_i)F_{n,i+1}(f_i) - \frac{2^{n+1}}{\pi^n}\sin(\pi f_i) \right| \\
&+ \sum_{i=n-l}^{n-1} F_{n,i+1}(f_i) \left| F_{-1,-i}(f_i) - \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^i} \right|.
\end{aligned}$$

Recall that Lemma 6.1.9 tells us that

$$\left| F_{n,i}(f) - \frac{2^{n-i+2}\mathscr{F}_i(f)}{\pi^{n-i+1}} \right| \leq \left(\frac{2}{3\pi}\right)^{n-i-1} \frac{\pi^2-8}{\pi^2}$$

and Theorem 6.1.11 tells us that $\frac{2^{n+1}}{\pi^n}\sin(\pi f_i)$ approximates $F_{-1,-i}(f_i)F_{n,i+1}(f_i)$ with an error of at most

$$\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}} \left(3^{i-2} + \frac{\pi^2-8}{8} + 3^{n-i-2}\right).$$

Then (6.9) is at most

$$\begin{aligned}
&\sum_{i=1}^{l} F_{-1,-i}(f_i) \left(\frac{2}{3\pi}\right)^{n-i-2} \frac{\pi^2-8}{\pi^2} \\
&+ \sum_{i=l+1}^{n-l-1} \frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}} \left(3^{i-2} + \frac{\pi^2-8}{8} + 3^{n-i-2}\right) \\
&+ \sum_{i=n-l}^{n-1} F_{n,i+1}(f_i) \left(\frac{2}{3\pi}\right)^{i-2} \frac{\pi^2-8}{\pi^2}.
\end{aligned}$$

Lemma 6.1.10 gives us

$$F_{n,i}(f) \leq \frac{2^{n-i+2}}{\pi^{n-i+1}} \left(1 + \frac{\pi^2-8}{8 \times 3^{n-i-1}}\right),$$

so (6.9) is at most

$$\begin{aligned}
&\sum_{i=1}^{l} \frac{2^{i+1}}{\pi^i}\left(1+\frac{\pi^2-8}{8\times 3^{i-2}}\right)\left(\frac{2}{3\pi}\right)^{n-i-2}\frac{\pi^2-8}{\pi^2}\\
&+\sum_{i=l+1}^{n-l-1}\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(3^{i-2}+\frac{\pi^2-8}{8}+3^{n-i-2}\right)\\
&+\sum_{i=n-l}^{n-1}\frac{2^{n-i+1}}{\pi^{n-i}}\left(1+\frac{\pi^2-8}{8\times 3^{n-i-2}}\right)\left(\frac{2}{3\pi}\right)^{i-2}\frac{\pi^2-8}{\pi^2}\\
=&\sum_{i=1}^{l}\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(3^{i-2}+\frac{\pi^2-8}{8}\right)\\
&+\sum_{i=l+1}^{n-l-1}\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(3^{i-2}+\frac{\pi^2-8}{8}+3^{n-i-2}\right)\\
&+\sum_{i=n-l}^{n-1}\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(3^{n-i-2}+\frac{\pi^2-8}{8}\right)\\
=&\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left((n-1)\frac{\pi^2-8}{8}+\sum_{i=1}^{n-l-1}3^{i-2}+\sum_{i=l+1}^{n-1}3^{n-i-2}\right).
\end{aligned}$$

By the same change of dummy variable as previously we can see that the two sums $\sum_{i=1}^{n-l-1} 3^{i-2}$ and $\sum_{i=l+1}^{n-1} 3^{n-i-2}$ are equal. Summing the geometric series we have that

$$\sum_{i=1}^{n-l-1} 3^{i-2} = \frac{3^{n-l-2}-\frac{1}{3}}{3-1} = \frac{3^{n-l-2}-\frac{1}{3}}{2},$$

and so (6.9) is at most

$$\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left((n-1)\frac{\pi^2-8}{8}+3^{n-l-2}-\tfrac{1}{3}\right)$$

□

When l approaches $\frac{n-1}{2}$ we see the bounds in the above lemmata come together. Indeed it is a simple task to represent them both as a single bound for all possible values of l (that is $1 \leq l \leq n-1$):

$$\frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(\left((n-1)+(\tfrac{n-1}{2}-l)-\left|\tfrac{n-1}{2}-l\right|\right)\frac{\pi^2-8}{8}+3^{n-l-2}-\tfrac{1}{3}\right).$$

6.2.2 Generating simple analytic upper and lower bounds

Up to now, we have given a method for calculating $\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)$ precisely and another for approximating it. This approximation, together with the bounds given for our approximations, enables us to calculate upper bounds for our sum, trading computation time for precision as required. However, short of delving into mathematical software and computing these calculations we have no *simple tangible formula* of an upper bound for our sum. Our aim therefore is to derive such a bound.

We shall be referring again to the bound introduced at the end of the previous section, so let us give it a name.

Definition 6.2.4. *Let n and l be positive integers such that $1 \leq l \leq n-1$. Define $\epsilon_{n,l}$ to be*

$$\epsilon_{n,l} := \frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}} \left(\left(\frac{3(n-1)}{2} - l - \left| \tfrac{n-1}{2} - l \right| \right) \frac{\pi^2-8}{8} + 3^{n-l-2} - \tfrac{1}{3} \right).$$

As was proved in the last section, with Lemma 6.2.2 and Lemma 6.2.3, we have that $\epsilon_{n,l}$ bounds the following:

$$\left| \sum_{i=1}^{n-1} F_{-1,-i}(f_i) F_{n,i+1}(f_i) - R_{n,l}(f_1, f_2, \ldots, f_{n-1}) \right| \leq \epsilon_{n,l}.$$

Obviously then

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i) F_{n,i+1}(f_i) \leq R_{n,l} + \epsilon_{n,l} \tag{6.10}$$

for all l such that $1 \leq l \leq n-1$. If we pick $l = 1$ then we shall need no calculations, since by definition $F_{-1,-1}(f)$ and $F_{n,n}(f)$ are both 1 for all $f \in [0,1]$ and we shall approximate everything else.

Lemma 6.2.5. *Let n be an integer greater than or equal to 3. Then the sum $\sum_{i=1}^{n-1} F_{-1,-i}(f_i) F_{n,i+1}(f_i)$ is at most*

$$\left(\frac{2}{\pi}\right)^n \left(2n + c_1 + c_2 n 3^{-n} - c_3 3^{-n}\right),$$

where the constants c_1, c_2 and c_3 are:

$$\begin{aligned} c_1 &= 2\pi - 6 + \frac{3(\pi^2-8)}{2}; \\ c_2 &= \frac{81(\pi^2-8)^2}{16}; \\ c_3 &= \frac{27(\pi^2-8)(3\pi^2-16)}{16}. \end{aligned}$$

Proof. To prove this result, we shall simply evaluate (6.10) at $l = 1$. Since we have already a formula for $\epsilon_{n,1}$ we shall obtain one for $R_{n,1}$. Recall that by definition

$$\begin{aligned}
R_{n,1} =& \sum_{i=1}^{1} F_{-1,-i}(f_i)\frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \\
&+ \sum_{i=2}^{n-2} \frac{2^{n+1}}{\pi^n}\sin(\pi f_i) \\
&+ \sum_{i=n-1}^{n-1} \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^i} F_{n,i+1}(f_i) \\
=&\frac{2^n}{\pi^{n-1}}(\mathscr{F}_2(f_1) + \mathscr{F}_{-n+1}(f_{n-1})) + \frac{2^{n+1}}{\pi^n}\sum_{i=2}^{n-2}\sin(\pi f_i) \\
\leq&\frac{2^n}{\pi^{n-1}}\left(1+1+\frac{2}{\pi}(n-3)\right) \\
=&\left(\frac{2}{\pi}\right)^n(2n+2\pi-6)\,.
\end{aligned}$$

Now, since $n \geq 3$, we have that $1 \leq \frac{n-1}{2}$ and so $\left|\frac{n-1}{2}-1\right| = \frac{n-1}{2}-1$. Hence $\epsilon_{n,1}$ is

$$\begin{aligned}
\epsilon_{n,1} &= \frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left(\left(\frac{3(n-1)}{2}-1-\left|\tfrac{n-1}{2}-1\right|\right)\frac{\pi^2-8}{8}+3^{n-3}-\tfrac{1}{3}\right) \\
&= \frac{2^{n-1}(\pi^2-8)}{\pi^n 3^{n-4}}\left((n-1)\frac{\pi^2-8}{8}+3^{n-3}-\tfrac{1}{3}\right) \\
&= \left(\frac{2}{\pi}\right)^n\left(\frac{3(\pi^2-8)}{2}+\frac{(\pi^2-8)^2}{16}\frac{n-1}{3^{n-4}}-\frac{\pi^2-8}{2\times 3^{n-3}}\right) \\
&= \left(\frac{2}{\pi}\right)^n\left(\frac{3(\pi^2-8)}{2}+\frac{81(\pi^2-8)^2}{16}\frac{n}{3^n}-\frac{81(\pi^2-8)^2}{16\times 3^n}-\frac{27(\pi^2-8)}{2\times 3^n}\right) \\
&= \left(\frac{2}{\pi}\right)^n\left(\frac{3(\pi^2-8)}{2}+\frac{81(\pi^2-8)^2}{16}\frac{n}{3^n}-\frac{27(\pi^2-8)(3\pi^2-24+8)}{16\times 3^n}\right) \\
&= \left(\frac{2}{\pi}\right)^n\left(\frac{3(\pi^2-8)}{2}+\frac{81(\pi^2-8)^2}{16}\frac{n}{3^n}-\frac{27(\pi^2-8)(3\pi^2-16)}{16}3^{-n}\right).
\end{aligned}$$

Bringing the formulae for $R_{n,1}$ and $\epsilon_{n,1}$ together, we have

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i) \leq \left(\frac{2}{\pi}\right)^n\left(2n+c_1+c_2 n 3^{-n}-c_3 3^{-n}\right).$$

□

If we try to obtain a lower bound for $\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)$ then the best we can do is 0, since the f_i could be chosen to be alternately 0 and 1. This corresponds to the Pebody code $B_{1b1b\ldots}$ which is just $A_{n,b}$.

More usefully, we shall look at a lower bound for $\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)$ once it has been maximised for $(f_1, f_2, f_3, \ldots, f_{n-1})$ over $[0,1]^{n-1}$. Let us say for each i

that $\hat{f}_i$ maximises $F_{-1,-i}(\hat{f}_i)F_{n,i+1}(\hat{f}_i)$ over $[0, 1]$. Then we want a lower bound for

$$\sum_{i=1}^{n-1} F_{-1,-i}(\hat{f}_i)F_{n,i+1}(\hat{f}_i).$$

Lemma 6.2.6. *Let n be an integer greater than or equal to 3. The sum*

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)$$

when it is maximised for $(f_1, f_2, f_3, \ldots, f_{n-1})$ over $[0, 1]^{n-1}$ is at least

$$\left(\frac{2}{\pi}\right)^n \left(2n + c_1' - c_2 n 3^{-n} + c_3 3^{-n}\right),$$

where the constants c_1', c_2 and c_3 are:

$$\begin{aligned} c_1' &= 2\pi - 6 - \frac{3(\pi^2 - 8)}{2}; \\ c_2 &= \frac{81(\pi^2 - 8)^2}{16}; \\ c_3 &= \frac{27(\pi^2 - 8)(3\pi^2 - 16)}{16}. \end{aligned}$$

Proof. Firstly note that, since $[0, 1]^{n-1}$ is a closed set in the complete space $\mathbb{R}^{n-1}$ and the polynomial $\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i)$ is continuous, such a maximising point, $(\hat{f}_1, \hat{f}_2, \hat{f}_3, \ldots, \hat{f}_{n-1})$ say, exists.

We may certainly start in a similar fashion to obtaining the upper bound, in that for all $(n-1)$-tuples $(f_1, f_2, f_3, \ldots, f_{n-1})$ in $[0, 1]^{n-1}$ we have

$$\sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i) \geq R_{n,l}(f_1, f_2, f_3, \ldots, f_{n-1}) - \epsilon_{n,l}.$$

Since $\sum_{i=1}^{n-1} F_{-1,-i}(\hat{f}_i)F_{n,i+1}(\hat{f}_i)$ is maximum, we have

$$\begin{aligned} \sum_{i=1}^{n-1} F_{-1,-i}(\hat{f}_i)F_{n,i+1}(\hat{f}_i) &\geq \sum_{i=1}^{n-1} F_{-1,-i}(f_i)F_{n,i+1}(f_i) \\ &\geq R_{n,l}(f_1, f_2, f_3, \ldots, f_{n-1}) - \epsilon_{n,l}, \end{aligned}$$

for all $f_i \in [0, 1]$. We shall proceed then to maximise $R_{n,l}(f_1, f_2, f_3, \ldots, f_{n-1})$. Again, selecting l to be 1, recall that $R_{n,1}$ is by definition

$$\begin{aligned}
R_{n,1} =& \sum_{i=1}^{1} F_{-1,-i}(f_i) \frac{2^{n-i+1}\mathscr{F}_{i+1}(f_i)}{\pi^{n-i}} \\
&+ \sum_{i=2}^{n-2} \frac{2^{n+1}}{\pi^n} \sin(\pi f_i) \\
&+ \sum_{i=n-1}^{n-1} \frac{2^{i+1}\mathscr{F}_{-i}(f_i)}{\pi^i} F_{n,i+1}(f_i) \\
=& \frac{2^n}{\pi^{n-1}} (\mathscr{F}_2(f_1) + \mathscr{F}_{-n+1}(f_{n-1})) + \frac{2^{n+1}}{\pi^n} \sum_{i=2}^{n-2} \sin(\pi f_i)\,.
\end{aligned}$$

Recall that in Definition 6.1.8 $\mathscr{F}_i(f)$ is defined as $\cos\left(\frac{\pi f}{2}\right)$ when i is odd and $\cos\left(\frac{\pi(1-f)}{2}\right)$ when i is even. Let $f_1 = 1$, let $f_{n-1} = 0$ when n is even and $f_{n-1} = 1$ when n is odd and let $f_i = \frac{1}{2}$ for all i from 2 to $n-2$. Now, $\mathscr{F}_2(f_1) = 1$, $\mathscr{F}_{-n+1}(f_{n-1}) = 1$ and $\sin(\pi f_i) = 1$ for all i from 2 to $n-2$. So

$$\begin{aligned}
R_{n,1} &= \frac{2^n}{\pi^{n-1}} \left(2 + \frac{2}{\pi}(n-3) \right) \\
&= \left(\frac{2}{\pi}\right)^n (2n + 2\pi - 6).
\end{aligned}$$

In the proof of Lemma 6.2.5 we showed that $\epsilon_{n,1}$ is

$$\left(\frac{2}{\pi}\right)^n \left(\frac{3(\pi^2-8)}{2} + \frac{81(\pi^2-8)^2}{16} \frac{n}{3^n} - \frac{27(\pi^2-8)(3\pi^2-16)}{16} 3^{-n} \right).$$

So

$$\begin{aligned}
\sum_{i=1}^{n-1} F_{-1,-i}(\hat{f}_i) F_{n,i+1}(\hat{f}_i) \geq& R_{n,1}(f_1, f_2, f_3, \ldots, f_{n-1}) - \epsilon_{n,1} \\
=& \left(\frac{2}{\pi}\right)^n \left(2n + c_1' - c_2 n 3^{-n} + c_3 3^{-n}\right).
\end{aligned}$$

□

Theorem 6.2.7. *The difference in the b^{n-2} coefficients of the maximised Pebody code and the alternating code, $R_{n,n-1}$, is*

$$\left(\frac{2}{\pi}\right)^n (2n + O(1)).$$

Proof. The result follows immediately from lemmata 6.2.5 and 6.2.6 □

A natural question to ask is: how does the difference $R_{n,n-1}$ compare with the original b^{n-2} coefficient of the alternating code? Recall that in Corollary 3.2.12 we showed that this coefficient is

$$\frac{1}{(n-2)!}\left(|U_n^{[2,2]}| + |U_n^{[3]}| - \frac{|U_n^{[2]}|(n-2)}{2} + \frac{|U_n|(n-2)(3n-1)}{24}\right).$$

We shall tackle this question in the next chapter.

Chapter 7

A programmatic implementation and analysis of the bounds on weakly consecutive repeat-free codes of the form B_y

Chapters 5 and 6 proposed and justified a method for approximating the size of a Pebody code (replacing an exact component $R_{n,n-1}$ by an approximation $R_{n,l}$ for some $1 \leq l < n-1$). This chapter reports on computer experiments to verify that our error bounds on $|R_{n,n-1} - R_{n,l}|$ are reasonable and to demonstrate that a good approximation for $R_{n,l}$ may be computed significantly faster than the corresponding exact value $R_{n,n-1}$. The computer experiments also generate data on the sizes of alternating codes and maximised Pebody codes, so we get a feel for how quickly they grow and their relative sizes.

7.1 Technical specifications of hardware and software and motivation for their selection

All of the tabulated data provided in this chapter was generated by *Mathematica* code executed in the *Wolfram Mathematica* environment version 7.0.0. The platform for this environment was *Microsoft Windows XP Professional* with *Service Pack 3* applied and nothing else was installed. The operating system was installed on a virtual machine using the open source virtualisation package *VirtualBox* version 3.0.2 developed by *Sun Microsystems*. The host operating system was again *Microsoft Windows XP Professional* with *Service Pack 3* applied running on an *AMD Athlon 64 X2* dual core 2.21GHz processor with 2GB of RAM. The guest operating system had 1GB of RAM and one of the cores dedicated to it.

The decision to perform the measurements on a virtual machine was twofold: cpu intensive processes such as automatic software updates, virus scans and file indexing could be safely turned off or simply would not be present; and since execution states of virtual machines can be saved, they can be reloaded for each set of experiments. This should have provided more consistent conditions for our observations. It is also important to underline that RAM usage was closely monitored for the duration of the code's execution and the 1GB available was comfortably in excess of what Mathematica's kernel requested. This means that there was no need to use virtual memory which would have involved reading and writing to disk for some of the calculations and most likely would have distorted the execution times measured.

7.2 Computing the difference in size of alternating codes and largest Pebody codes

Recall that in Definition 1.2.2 we said for fixed n and $b \to \infty$, a largest Pebody code, $B_{n,b}$, has magnitude

$$|B_{n,b}| = \alpha_{n,n} b^n + \alpha_{n,n-1} b^{n-1} + \beta_{n,n-2} b^{n-2} + O(b^{n-3}).$$

We also said, in Definition 1.2.1, that the alternating code $A_{n,b}$ has magnitude

$$|A_{n,b}| = \alpha_{n,n} b^n + \alpha_{n,n-1} b^{n-1} + \alpha_{n,n-2} b^{n-2} + O(b^{n-3}).$$

Let us define

$$\Delta_n = \alpha_{n,n-2} - \beta_{n,n-2}.$$

In Chapter 4 we develop theory that enables us to calculate Δ_n and in Chapter 6 we give a range of approximations to Δ_n, with error bounds, that are theoretically quicker to compute.

We pick two approximations to Δ_n from the range offered to us by Chapter 6, let us call them $\hat{\Delta}_n$ and $\bar{\Delta}_n$, to illustrate the tradeoff between precision and computation time in practice. Table 7.1 shows that significant savings in computation time are gained by using our approximation, without losing too much accuracy. The table also vindicates the claims of the previous chapter by validating that our approximations indeed lie within the theoretical error bounds. We note that, since the theoretical error bounds are reasonably close to the actual errors, further work in this area is unlikely to produce significant improvements to our approximations.

At this point we enter a more technical discussion of what Δ_n, $\hat{\Delta}_n$ and $\bar{\Delta}_n$ are. Recall that in Definition 6.2.1 we defined $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ as an approximation to the difference in the b^{n-2} terms between the alternating code and a Pebody code. The higher the integer l the better the approximation is and for $l = n-1$ we have the difference exactly. When we maximise the function $R_{n,n-1}$ over its domain $[0,1]^{n-1}$

Table 7.1: Calculations of Δ_n, the actual measured errors incurred by approximating it, the theoretical maximum such errors and relevant computation times of approximation in seconds

n	Δ_n	Compute time in seconds	Actual error measured $\Delta_n - \hat{\Delta}_n$	Theoretical maximum error for $\lvert\Delta_n - \hat{\Delta}_n\rvert$	Compute time in seconds	Actual error measured $\Delta_n - \bar{\Delta}_n$	Theoretical maximum error for $\lvert\Delta_n - \bar{\Delta}_n\rvert$	Compute time in seconds
20	4.7694e(-3)	2.123	-2.0084e(-9)	1.7045e(-8)	1.082	-1.2897e(-6)	1.2419e(-5)	0.350
40	1.1421e(-6)	8.873	-4.0998e(-18)	3.4494e(-17)	3.104	-1.8677e(-11)	1.6498e(-10)	0.611
60	2.0492e(-10)	24.645	-8.3017e(-27)	6.9844e(-26)	6.570	8.0190e(-16)	6.5754e(-15)	0.792
80	3.2676e(-14)	49.791	-1.6810e(-35)	1.4142e(-34)	12.027	-3.0615e(-20)	2.6206e(-19)	0.912
100	4.8843e(-18)	89.909	-3.4037e(-44)	2.8636e(-43)	19.648	-4.1127e(-25)	3.4815e(-24)	1.282
120	7.0085e(-22)	149.696	-6.8919e(-53)	5.7983e(-52)	31.485	-4.9174e(-29)	4.1626e(-28)	1.282
140	9.7770e(-26)	231.100	-1.3955e(-61)	1.1741e(-60)	45.364	1.9789e(-33)	1.6590e(-32)	1.503
160	1.3360e(-29)	350.762	-2.8257e(-70)	2.3773e(-69)	58.892	-7.8417e(-38)	6.6119e(-37)	1.732
180	1.7972e(-33)	651.667	-5.7215e(-79)	4.8136e(-78)	80.034	3.1362e(-42)	2.6352e(-41)	1.953
200	2.3876e(-37)	1168.760	-1.1585e(-87)	9.7468e(-87)	106.955	-1.2473e(-46)	1.0502e(-45)	2.163

we have Δ_n. If we choose l to be $\lfloor \frac{n}{2} \rfloor$ then $R_{n,l}$ maximised over $[0,1]^{n-1}$ gives us $\hat{\Delta}_n$ and, with $l = \lfloor \sqrt{n} \rfloor$, we obtain $\bar{\Delta}_n$ when $R_{n,l}$ is maximised over $[0,1]^{n-1}$.

Recall also that we proved the theoretical maximum error in approximating our $R_{n,n-1}(f_1, f_2, \ldots, f_{n-1})$ by $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ is at most $\epsilon_{n,l}$ which was defined in Definition 6.2.4. The motivation behind this approximation is that it is far easier to compute the functions $\cos(\frac{\pi f}{2})$ or $\cos(\frac{\pi(1-f)}{2})$ than the recursively defined $F_{n,i}(f)$. Furthermore, in the cases where $l < \frac{n-1}{2}$ and $l+1 \leq i \leq n-l-1$, when trying to maximise the part of $R_{n,l}$ involving f_i over $[0,1]$ there is an analytic solution of $\frac{2^{n+1}}{\pi^n}$. For $R_{n,n-1}$ we are still faced with maximising $F_{-1,-i}(f_i)F_{n,i+1}(f_i)$.

Note that the points at where the functions $R_{n,n-1}$, $R_{n,\lfloor \frac{n}{2} \rfloor}$ and $R_{n,\lfloor \sqrt{n} \rfloor}$ are maximised are most likely not the same. Their respective maxima are still of course within their error bounds as can be seen from the following short lemma.

Lemma 7.2.1. *Let F and G be real functions over some domain. Suppose that their maxima exist and are at points f and g respectively. If for some ϵ we may write $\|F - G\|_\infty < \epsilon$, then $|F(f) - G(g)| < \epsilon$.*

Proof. Suppose the result is false. Without loss of generality suppose that $F(f) \geq G(g)$. Then

$$\begin{aligned} F(f) &\geq G(g) + \epsilon \\ &\geq G(f) + \epsilon. \end{aligned}$$

This contradicts that $\|F - G\|_\infty < \epsilon$. □

7.3 Comparing the difference to the size of the alternating code

In studying the difference Δ_n in the b^{n-2} term between the alternating code and a largest Pebody code, a natural question is: how does this difference compare to $\alpha_{n,n-2}$? In other words by what proportion can a Pebody code be larger than the alternating code of the same length? Table 7.2, describes the size of the alternating code alongside Δ_n for comparison. The last two columns of the table give us the ratios $\frac{\alpha_{n,n-2}}{\Delta_n}$ and $\frac{\alpha_{n,n-2}}{n\Delta_n}$ in order to show us how $\alpha_{n,n-2}$ and Δ_n relate in size for large n.

In our measurements Δ_n is smaller than $\alpha_{n,n-2}$, but the rate at which the ratio $\frac{\alpha_{n,n-2}}{\Delta_n}$ grows is far slower than that at which $\alpha_{n,n-2}$ and Δ_n shrink. The right hand column, $\frac{\alpha_{n,n-2}}{n\Delta_n}$, seems convergent. Thus, for some $\lambda \approx 0.0795$, it appears we have

$$\Delta_n \approx \frac{\alpha_{n,n-2}}{\lambda n}$$

Table 7.2: Calculations of coefficients of $A_{n,b} = \alpha_{n,n}b^n + \alpha_{n,n-1}b^{n-1} + \alpha_{n,n-2}b^{n-2} + O(b^{n-3})$, Δ_n and how its magnitude compares to $\alpha_{n,n-2}$

n	$\alpha_{n,n}$	$\alpha_{n,n-1}$	$\alpha_{n,n-2}$	Δ_n	$\frac{\alpha_{n,n-2}}{\Delta_n}$	$\frac{\alpha_{n,n-2}}{n\Delta_n}$
20	1.52234e-4	-1.52234e-3	7.46589e-3	4.76811e-3	1.56580	0.0782899
40	1.82018e-8	-3.64037e-7	3.61485e-6	1.14206e-6	3.16520	0.0791301
60	2.17629e-12	-6.52888e-11	9.75316e-10	2.04923e-10	4.75942	0.0793237
80	2.60208e-16	-1.04083e-14	2.07570e-13	3.26761e-14	6.35236	0.0794045
100	3.11116e-20	-1.55558e-18	3.88045e-17	4.88430e-18	7.94473	0.0794473
120	3.71985e-24	-2.23191e-22	6.68390e-21	7.00852e-22	9.53682	0.0794735
140	4.44762e-28	-3.11333e-26	1.08806e-24	9.77697e-26	11.1288	0.0794912
160	5.31777e-32	-4.25422e-30	1.69953e-28	1.33604e-29	12.7206	0.0795038
180	6.35817e-36	-5.72235e-34	2.57219e-32	1.79718e-33	14.3124	0.0795132
200	7.60212e-40	-7.60212e-38	3.79730e-36	2.38762e-37	15.9041	0.0795206

as $n \to \infty$. So when n is large we may approximate (the three leading terms of) the the size of a largest Pebody code by that of the alternating code and incur only small error in just the third term.

7.4 Discussion of algorithms used

For Table 7.1 we needed to calculate $R_{n,l}$ and measure the computation time for each calculation. The code was broken down into the following three functions:$getF(n,i)$, $getMaximisedR(n,l)$ and $getEpsilon(n,l)$.

The simplest of these is $getEpsilon(n,l)$ which used relatively low precision floating point arithmetic (called *machine precision* in *Mathematica* — about 16 significant figures) to give us an idea of the magnitude of the error we could expect from the approximation $R_{n,l}$. It computes $\epsilon_{n,l}$ from the formula as defined in Definition 6.2.4 and is very quick to evaluate.

The function $getF(n,i)$ returns $F_{n,i}(f)$, defined in Definition 5.1.1. The definition is a recursive one and so $F_{n,i}(f)$ must be calculated from $F_{n,i+1}(f)$ and so on from $F_{n,n}(f) = 1(f)$. However, $getF(n,i)$ makes use of the identities

$$\begin{aligned} F_{n,i}(f) &= F_{n-2,i-2}(f), \\ F_{n+1,1}(f) &= \int_0^{1-f} F_{n,1}(g)dg, \\ F_{n,i+1}(f) &= F_{n,i}(1-f), \end{aligned}$$

proved in Theorem 5.1.7 and Lemma 5.1.8. By using these results, $getF(n,i)$ would only calculate $F_{n-i,0}(f)$ or $F_{n-i+1,1}(f)$ if i is odd or even respectively. We shall therefore only talk in terms of $getF(n,0$ and $getF(n,1)$. In calculating $getF(n,1)$, for all positive integers $m < n$ one gets $F_{m,1}(f)$ for free. Each of these values is stored for (nearly) immediate future retrieval. Any future request for, say, $getF(m,0)$ is then calculated by computing the substitution $f \mapsto 1-f$ on the result of $getF(m+1,1)$, which is already stored. Again, $getF(m,0)$ would be stored for future retrieval.

Although we did not wipe these results between each successive calculation of $R_{n,l}$, $getF$ actually times itself and stores this time along with the one-time computed value. Fresh experiments then have their computation times artificially corrected with the relevant stored times. This means the displayed computation times are as if the results from previous experiments had been wiped but the experiments actually take considerably less time to perform.

Finally, $getMaximisedR(n,l)$ returns the function $R_{n,l}(f_1, f_2, \ldots, f_{n-1})$ maximised over $[0,1]^{n-1}$. The first thing this function does is determine the accuracy it must work to and does this by calling $getEpsilon(n,l)$. Mathematica then doubles the number of decimal places to get the number it decides to work to (this is the default in Mathematica). This ensures that we calculate our answer to a suitably high precision and so can rely on the output for the $R_{n,n-1} - R_{n,l}$ (referred to as "Actual error measured") columns in Table 7.1.

The function then follows the definition of $R_{n,l}$ given in Definition 6.2.1 by using the approximation for $getF(n,i)$ when $n - i \geq l$. The function is maximised using Mathematica's built-in tools to a suitably high precision with reasonably large cap on the maximum number of iterations allowed until the answer lies within our tolerance. This tolerance is set to be a few decimal places within the maximum error $\epsilon_{n,l}$ (this varies, but about 3 or 4). As previously discussed, if any of the $getF(n,i)$ values were used from earlier experiments, the computation time is corrected. This is not a matter for concern since the times for these calculation do not vary between runs by more than a couple of milliseconds. The Mathematica code for these functions is given in Appendix C.

The columns of coefficients in Table 7.2 are generated by the recursive function $scrfull(n, i, parity)$, which calculates some of the polynomial $P_{n,i}(b,c)$ as defined in Definition 3.1.3. This is again recursively defined and as with $getF(n,i)$, results are stored for future retrieval and similar identities are used to prevent duplicate calculation of identical polynomials.

If $scrfull(n, 0, parity)$ were to calculate the entire polynomial $P_{n,0}$, which is a polynomial in b and c with combined degree n, it would be dealing with coefficient for each term $b^j c^k$ where $0 \leq j + k \leq n - i$. There are

$$\begin{aligned}\sum_{j=0}^{n}(n-j) &= n(n+1) - \sum_{j=0}^{n} j \\ &= n(n+1) - \frac{n(n+1)}{2} \\ &= \frac{n(n+1)}{2}\end{aligned}$$

such terms. As such, calculation to values of $n = 200$ would have been impractical to compute with the hardware outlined at the beginning of this chapter.

Since we only need the terms $b^j c^k$ where $j + k \geq n - 2$ there is some optimisation we can do. From Definition 3.1.3 we have that, for even i,

$$P_{n,i}(b, w_i) := \sum_{w_{i+1}=1}^{w_i - 1} P_{n,i+1}(b, w_{i+1}).$$

Apply to that Lemma 2.0.7 and we see that a term $\alpha_{j,k} b^j w_{i+1}^k$ will contribute

$$\alpha_{j,k} b^j \left(\frac{1}{k+1}(w_i - 1)^{k+1} + \frac{1}{2}(w_i - 1)^k + \frac{k}{12}(w_i - 1)^{k-1} + O(w_i^{k-2}) \right)$$

to $P_{n,i}(b, w_i)$. Thus, if we are only interested in the polynomial with terms with combined degree $(n-i)$, $(n-i-1)$ or $(n-i-2)$ then we need only store and compute these terms. Rather than computing the entire sum, using say Mathematica's built-in *Sum* function, we use Lemma 2.0.7. This means that we need only manage the three homogenous polynomials that we are interested in. So, for $P_{n,0}$ we are computing polynomials with $n+1$, n and $n-1$ terms. This is only $3n$ terms, which is significantly quicker to compute than the aforementioned $\frac{n(n+1)}{2}$ terms and requires a great deal less memory to store it. Using this method we were (finally) about to compute the coefficient columns in Table 7.2. The other columns are just quotients of these numbers and so were simple to compute. The code for the function $scrfull(n, i, parity)$ is given in Appendix D.

The columns for Table 3.2 were similarly easy to compute once the three leading coefficients for the polynomial $|A_{n,b}|$ had been determined. The formulae linking the coefficients to the values $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}|+|U_n^{[3]}|$ given in Corollary 3.2.12 were solved to give $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}|+|U_n^{[3]}|$. The values of the coefficients were then substituted into the new formulae to generate the results for Table 3.2.

Appendix A

MatLab programming code

The following *MatLab* code details the recursive function $scrf(n, parity, c)$ that was used to calculate the polynomial $|A_{n,b}|$ for n from 1 to 10. The output is shown in Table 3.1.

The variable *parity* takes the values 0 or 1 and $scrf(n, parity, c)$ gives the polynomial in b and c counting the number of down/up or up/down words respectively of length n with first letter c. Thus $scrf(n, 1, c)$ counts the number of up/down words of length n with first letter c.

We are interested in $|A_{n,b}|$, the total number of up/down words of length n. There is however a simple bijection between the up/down words of length n and the number of down/up words of length $n-1$ with first letter b. In order to calculate the polynomial $|A_{n,b}|$ therefore, we need only to make the substitution $c = b$ on the polynomial $scrf(n+1, 0, c)$.

```
function sym_poly=scrf(n,parity,c)
%returns number of words of length n that start with the letter c and that
%are up/down, if parity=1, and down/up, if parity=0
%scrf_poly_table(:,parity+1) is a col of polys for increasing
%n with parity 'parity'
global scrf_poly_table;
known_rows=size(scrf_poly_table,1);
if known_rows==0
    scrf_poly_table=sym([1 1 1 1]);
    known_rows=1;
end
syms d b;
if n==1
    sym_poly=sym(1);
    return
end
if known_rows>=n
    if scrf_poly_table(n,parity+3)==1
        sym_poly=scrf_poly_table(n,parity+1);
        return
    end
end

%we don't know this poly yet
if parity==1
```

```
    sym_poly=subs(symsum(scrf(n-1,1-parity,c),c,d+1,b),d,c);
else
    sym_poly=subs(symsum(scrf(n-1,1-parity,c),c,1,d-1),d,c);
end
scrf_poly_table(n,parity+1)=sym_poly;
scrf_poly_table(n,parity+3)=1;
return

%Declare variables used in the calculations
syms b c;
scrf(10,0,c);
scrf(11,0,c);

%access global variable scrf_poly_table
global scrf_poly_table;

%Display calculated polynomials
subs(scrf_poly_table(:,1),c,b);
```

Appendix B

MatLab code generating $F_{n,1}(f)$ and $\tilde{F}_{n,1}(f)$

The code here generates the data for Table 5.1 and Table 5.2 showing the polynomials $F_{n,1}(f)$ and the normalised $\tilde{F}_{n,1}(f)$ respectively, for n from 2 to 10. When $n = 1$ we have $F_{1,1}(f) = 1 = \tilde{F}_{1,1}(f)$ by definition.

Most of the complicated computation is done in $scrf(n, parity, c)$, which was described in Appendix A. By definition 5.1.1, $F_{n,1}(f)$ is the coefficient of the highest power of b in $P_{n,1}(b, bf)$. This is precisely how we generate the $F_{n,1}(f)$ for Table 5.1. We employ the function $lcoef(poly, variable)$, which obtains the leading coefficient of the polynomial *poly* in the variable *variable*, to perform this task.

The normalised polynomials $\tilde{F}_{n,1}(f)$ were obtained by dividing the $F_{n,1}(f)$ by $F_{n,1}(0)$ as per Definition 5.1.9. They are displayed in Table 5.2.

```
%Populate the variable scrf_poly_table with polynomials P_{n,1}(b,c).
global scrf_poly_table;
syms b c f;
scrf(10,0,c);
scrf(11,0,c);

%Substitute c=bf so we can extract leading coefficient: a polynomial in f.
polys=subs(scrf_poly_table(:,2),c,b*f);

%print out in scientific notation
format short e;

for i=2:n
    %Coefficients of F_{n,1}(f).
    %(not normalised)
    sym2poly(lcoef(polys(i,1),b))
end for i=2:n
    %Coefficients of \tilde{F}_{n,1}(f)
    %(normalised)
    sym2poly(lcoef(polys(i,1),b))/subs(lcoef(polys(i,1),b),f,0)
end


%Function that returns the leading coefficient of a symbolic polynomial
```

```
function coefficient=lcoef(poly,variable)
%variable must be positive for log to work, so substitute for it 'pos'
pos=sym('pos','positive');
poly=subs(poly,variable,pos);

%the next few lines gets the leading coefficient
poly=expand(poly);
poly=collect(poly,pos);
[coeffs_row,terms_row]=coeffs(poly,pos);

terms_row=eval(simplify(log(terms_row)/log(pos)));
[c,i]=max(terms_row);
coefficient=coeffs_row(1,i);
return
```

Appendix C

Mathematica code for calculating and timing maximisation of $R_{n,l}$

The code given here was written for *Mathematica* and was used to demonstrate the effectiveness, in terms of accuracy versus computation time, of the approximations $R_{n,l}$ to the difference in size between the alternating code and largest Pebody codes. The values approximated, errors in approximation and compute times are shown in Table 7.1. Each of the functions is described in some detail in 7.4 and it is here the reader is referred for further discussion.

```
getF[n_, i_] := (
  (*
  Return {integrationTime, substitutionTime,
  the polynomial F_ {n,i}(f) as defined in 4.1.1}.
  Justification for storing two times:
  For each getF[n,i] where i==1 we perform an integration but for
  each subsequent getF[m,i] with m<n there is no additional
  computation required.
  For i==0 a fresh substitution must be made for each value of n.
  Therefore we want to sum the substitutionTimes but only use the
  largest integrationTime.
  *)
  Module[{
    parity = Mod[i, 2],
    prevIntegrationTime, prevResult,
    integrationTime = 0, substitutionTime = 0, result
   },
   If[i != parity,
    (*
    We will only store getF[m,0] and getF[m,1] for some integer m.
    *)
    getF[n - i + parity, parity],
    (*
    We now have i==parity.
    *)
    If[i == 1,
     (*Using 4.1.8*)
     (*
     The substitutionTime will be 0, so just get elements {1,3}
     *)
```

```
     {prevIntegrationTime, prevResult} = getF[n - 1, 1][[{1, 3}]];
     {integrationTime, result} = Timing[Expand[
      Integrate[prevResult, {f, 0, 1 - f}]
     ]];,
     (*
     No sense in recalculating, since a substitution of "f->1-f"
     is much quicker; we shall however store the result for
     future use.
     *)
     (*
     The substitutionTime will be 0, so just get elements {1,3}
     *)
     {prevIntegrationTime, prevResult} = getF[n + 1, 1][[{1, 3}]];
     {substitutionTime, result} = Timing[Expand[
      prevResult /. f -> 1 - f
     ]];
    ];
    getF[n, i] = {prevIntegrationTime + integrationTime,
      substitutionTime, result}
   ]
  ]
)
(*
{IntegrationTime=0,substitutionTime=0, F_ {1,1}(f):=1}
*)
getF[1,1] = {0, 0, 1};


getMaximisedR[n_, l_] := (
  (*
  Return R_ {n,l}, as defined in 5.2.1,
  maximised over f_ 1,f_ 2,...,f_ {n-1}
  *)
  If[l < 1 || l > n || n < 2,
   Print["Bad input parameters. n=" <> ToString[n]
     <> " l=" <> ToString[l]
   ];
   Interrupt[];
  ];
  Clear[f];
  Module[
   {
    total = 0,
    (*
    Accuracy to be number of decimal places of error plus
    extra error incurred due to summing n times. If l==n-1 then
    we are not approximating, so use getEpsilon[n,l-1], the
    smallest error for this n.
    *)
    accuracyGoal = Max[6,
      If[l == n - 1,
       Ceiling[-Log[10, getEpsilon[n, l - 1]] + Log[10, n]],
       Ceiling[-Log[10, getEpsilon[n, l]] + Log[10, n]]
       ]
      ],
    substitutionTimes = Table[0, {temp1, n}, {temp2, 2}],
    substitutionTime = 0,
    integrationTime = 0,
```

```
 maxIntegrationTime = 0,
 totalTime = 0
},
totalTime = First[Timing[
   Do[
     (*
     We can optimise here for when i-1>=l and n-i-1>=l:
     just use f_i=1/2. Hence total+=2^{n+1}/pi^n.
     *)
     If[i - 1 >= l && n - i - 1 >= l,
      total += N[2^(n + 1)/Pi^n, 2*accuracyGoal],

      (*
      Calculate lhs of summand of R_ {n,l} given in 5.2.1.
      *)
      If[i - 1 < l,
       {integrationTime, substitutionTime, lhs} = getF[-1, -i],
       lhs = 2^(i + 1)/Pi^(i) Cos[Pi f/2];
       If[EvenQ[i], lhs = lhs /. f -> 1 - f];
      ];
      maxIntegrationTime = Max[maxIntegrationTime, integrationTime];
      (*
      Record substitutionTime for (i-1,Mod[-i,2]). We actually
      use (i-1+1,Mod[-i,2]+1) because Mathematica starts arrays
      at 1 not 0.
      *)
      substitutionTimes[[i, Mod[-i, 2] + 1]] = substitutionTime;

      (*
      Calculate rhs of summand of R_ {n,l} given in 5.2.1.
      *)
      If[n - i - 1 < l,
       {integrationTime, substitutionTime, rhs} = getF[n, i + 1],
       rhs = 2^(n - i + 1)/Pi^(n - i) Cos[Pi f/2];
       If[OddQ[i], rhs = rhs /. f -> 1 - f];
      ];
      maxIntegrationTime = Max[maxIntegrationTime, integrationTime];
      (*
      Record substitutionTime for (n-i-1,Mod[i+1,2]). We actually
      use (n-i,Mod[i+1,2]+1) because Mathematica starts arrays
      at 1 not 0.
      *)
      substitutionTimes[[n - i, Mod[i + 1, 2] + 1]] = substitutionTime;

      total += NMaxValue[
        {
         lhs*rhs,
         0 <= f && f <= 1
         }, {f},
        AccuracyGoal -> accuracyGoal,
        PrecisionGoal -> Infinity,
        MaxIterations -> 999,
        WorkingPrecision -> 2*accuracyGoal
        ];

      ],
     (*
     Do this for i=1..n-1.
```

```
        *)
        {i, n - 1}
        ];
      ]];
   totalTime += maxIntegrationTime + Total[substitutionTimes, 2];
   Return[{totalTime, total}];
   ]
  )

getEpsilon[n_, l_] := (
  (*
  A fast/rough estimate of epsilon_ {n,l}.
  *)
  If[l == n - 1, Return[0];];
  N[
    2^(n - 1) (Pi^2 - 8)/Pi^n/3^(n - 4) (
      (3 (n - 1)/2 - l - Abs[(n - 1)/2 - l]) (Pi^2 - 8)/8
      + 3^(n - l - 2)
      - 1/3
    ), 5
  ]
)
```

Appendix D

Mathematica code for calculating the three leading terms of $\left|A_{n,b}\right|$ as a polynomial in b

The function $scrfull(n, i, level)$, written for *Mathematica*, calculates parts of the polynomial $P_{n,i}(b, c)$. It returns the homogenous polynomial who's terms are those in $P_{n,i}(b, c)$ with combined degree $n - i - level$. With $level = 0$, for example, we obtain the terms of highest combined degree. Since $|A_{n,b}| = P_{n,0}(b, b)$, we can use this polynomial to give us the three leading terms of the alternating code. This is how we generated these coefficients in Table 7.2. Furthermore, one can use Corollary 3.2.12 to calculate the values $|U_n|$, $|U_n^{[2]}|$ and $|U_n^{[2,2]}| + |U_n^{[3]}|$. We give the results for these values in Table 3.2. A more thorough discussion of how this recursive function works, and why it is such an improvement on $scrf(n, parity, c)$ in Appendix A, is given in 7.4.

```
%Clear[scrfull];(*Remove any previous data*)
%(*
%level (=0,1 or 2) tells us which of the top 3 homogeneous polynomials
%we are referring to:
%   0 - poly in P_{n,i}(b,c) of highest combined degree.
%   1 - poly in P of second highest combined degree.
%   2 - poly in P of third highest combined degree.
%*)
%scrfull[n_, i_, level_] := (
%  Clear[b, c];
%  If[n < i,
%   Print["'n'(" <> ToString[n] <> ") is less than 'i'(" <>
%     ToString[i]];
%   Interrupt[]
%   ];
%  parity = Mod[i, 2];
%  If[i != parity,
%   Return[scrfull[n - i + parity, parity, level]]
%   ];
%  If[level == 0,
%   Return[
%    scrfull[n, i, 0] =
```

```
%     If[i == 1,
%      Integrate[scrfull[n - 2, 0, 0], {c, c, b}],
%      Integrate[scrfull[n, 1, 0], {c, 0, c}]
%      ]
%    ]
%   ];
%  If[level == 1,
%   Return[
%    scrfull[n, i, 1] =
%     If[i == 1,
%      (Integrate[scrfull[n - 2, 0, 1], {c, c, b}]
%        + 1/2 (
%          Replace[scrfull[n - 2, 0, 0], c -> b, {-1}]
%           - scrfull[n - 2, 0, 0]
%          )
%       ),
%      (Integrate[scrfull[n, 1, 1], {c, 0, c}]
%        - 1/2 (
%          scrfull[n, 1, 0]
%           + Replace[scrfull[n, 1, 0], c -> 0, {-1}]
%          )
%       )
%      ]
%    ]
%   ];
%  If[level == 2,
%   Return[
%    scrfull[n, i, 2] =
%     If[i == 1,
%      (Integrate[scrfull[n - 2, 0, 2], {c, c, b}]
%        + 1/2 (
%          Replace[scrfull[n - 2, 0, 1], c -> b, {-1}]
%           - scrfull[n - 2, 0, 1]
%          )
%        + 1/12 (
%          Replace[D[scrfull[n - 2, 0, 0], c], c -> b, {-1}]
%           - D[scrfull[n - 2, 0, 0], c]
%          )
%      ),
%      (Integrate[scrfull[n, 1, 2], {c, 0, c}]
%        - 1/2 (
%          scrfull[n, 1, 1]
%           + Replace[scrfull[n, 1, 1], c -> 0, {-1}]
%          )
%        + 1/12 (
%          D[scrfull[n, 1, 0], c]
%           - Replace[D[scrfull[n, 1, 0], c], c -> 0, {-1}]
%          )
%      )
%     ]
%   ]
%  ];
%)
%scrfull[0, 0, 0] = 1;
%scrfull[0, 0, 1] = 0;
%scrfull[0, 0, 2] = 0;
%scrfull[1, 1, 0] = 1;
%scrfull[1, 1, 1] = 0;
```

```
%scrfull[1, 1, 2] = 0;
```

Bibliography

[Bea96] Alan F Beardon. Sums of powers of integers. *The American Mathematical Monthly*, 1996.

[BR90] Bruce Bauslaugh and Frank Ruskey. Generating alternating permutations lexicographically. *BIT*, 30(1):17–26, March 1990.

[BR09] Yuliy Baryshnikov and Dan Romik. Enumeration formulas for Young tableaux in a diagonal strip. *Israel Journal of Mathematics*, 2009. forthcoming.

[Car66] Lennart Carleson. On convergence and growth of partial sums of Fourier series. *Acta Mathematica*, 116(1):135–157, December 1966.

[Dun99] William Dunham. *Euler: the master of us all.* Mathematical Association of America, 1999.

[Hal57] Paul Richard Halmos. *Introduction to Hilbert space and the theory of spectral multiplicity.* Chelsea publishing company, second edition, 1957.

[KB67] Donald E Knuth and Thomas K Buckholtz. Computation of tangent, Euler and Bernoulli numbers. *Mathematics of Computation*, 21(100):663–688, October 1967.

[Lan99] Serge Lang. *Complex analysis.* Springer, fourth edition, 1999.

[Lar73] Ronald Larsen. *Functional analysis an introduction.* Marcel Dekker, 1973.

[Pas54] Blaise Pascal. Sommation des puissances numeriqués. 1654.

[Peb06] Luke Pebody. The largest strongly consecutive repeat-free code. *Journal of Combinatorial Theory, Series A*, 113(3):551–555, Apr 2006.

[Slo] N J A Sloane. An on-line version of the encyclopedia of integer sequences. Website. `http://www.research.att.com/~njas/sequences/`.

[Tit39] Edward Charles Titchmarsh. *The theory of functions.* Oxford University Press, second edition, 1939.