Санкт–Петербургский государственный университет

Факультет прикладной математики – процессов управления

Кафедра управления медико-биологическими системами

Боровой Иван Иванович

Образовательная программа: МК.3005.2014 «Математическая кибернетика»

Направление подготовки: 02.06.01 «Компьютерные и информационные науки»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Методы рациональной интерполяции в задаче помехоустойчивого кодирования

Заведующий кафедрой,
доктор физ.-мат. наук,
профессор Александров А.Ю.

Научный руководитель,
доктор физ.-мат. наук,
профессор Утешев А.Ю.

Рецензент,
кандидат техн. наук,
доцент Ежгуров В.Н.

Санкт-Петербург
2017

Оглавление

Глава 1. Введение

Разбиение математики на разделы, как, впрочем, и всего знания на науки, весьма условно. Средневековый учёный мог всю жизнь заниматься одним делом, а потомки называли бы его физиком, астрономом, математиком и философом одновременно.

Однако, не упрекая и не оправдывая существующий порядок вещей, стоит обратить внимание на одно его очевидное достоинство. Многие вопросы, лежащие на пересечении смежных областей знания, открываются под разными углами и, порой, демонстрируют совершенно неожиданные для приверженцев конкретной точки зрения свойства.

Данная работа, в целом, посвящена теории кодирования, подразделу теории информации[1]. Задача помехоустойчивого кодирования не нова, и классические подходы к её решению были сформулированы ещё Р. Хэммингом в середине прошлого века.

В последние десятилетия вычислительная техника сделала огромный шаг вперёд. Стало возможным применение сложных алгоритмов на широком классе устройств. Сам список этих устройств удлинился на порядок, что, в свою очередь, расширило спектр задач и предъявило более высокие требования к алгоритмам их решения и программным реализациям.

Ещё не так давно закон Мура обеспечивал стабильный рост вычислительных мощностей, а внедрению нового эффективного алгоритма зачастую предпочитался переход на более производительную архитектуру. Сегодня этот закон потерял свою актуальность, чем можно объяснить резко возросшую популярность технологий параллельного программирования и гетерогенных вычислительных комплексов.

Такое развитие событий благотворно повлияло на теоретическую сторону информационных технологий: быстрые алгоритмы стали особенно востребова-

[1]Клод Шеннон (1916 – 2001) – американский инженер и математик, «отец» теории информации. Его результаты не использованы здесь в явном виде, однако в силу значительного вклада в эту науку его имя, несомненно, стоит упоминания.

ны. Однако их сложность настолько возросла, что выбрать лучшее решение из удовлетворительных может оказаться непозволительно затратно и долго.

Стремясь всё же удовлетворить теоретический интерес и практическую потребность в хороших кодах, данная работа ставит перед собой следующие цели:

- описать основные идеи и методы теории кодирования;
- рассмотреть применение методов рациональной интерполяции в теории кодирования;
- оценить перспективы использования методов рациональной интерполяции в этой области.

Глава 2. Задача помехоустойчивого кодирования

Прежде чем перейти к описанию предметной области, давайте договоримся о терминах.[1]

Начнём с простого. Слово «помехоустойчивый» будет употребляться в своём естественном и интуитивном смысле — устойчивый к помехам. С той лишь разницей, что степень, способ устойчивости, а также характер помех формализуются и оговариваются в каждом случае особо.

Слово же «кодирование» не столь очевидно.

Вообще говоря, кодирование не подразумевает сокрытия информации. Подобное кодирование правильно называть шифрованием. То есть шифрование — способ кодирования, ставящий целью сокрытие информации.

Кодирование также не есть сжатие. Понятие сжатия так таковое допускает весьма вольные толкования, однако наиболее часто оно обозначает существенное уменьшение размера данных при пренебрежимо малой потере их информационной составляющей. Стоит отметить, что кодирование не допускает потери информации. То есть, обратимость процесса кодирования существенна и принципиальна.

Короче, кодирование — изменение представления информации, сохраняющее её содержание.

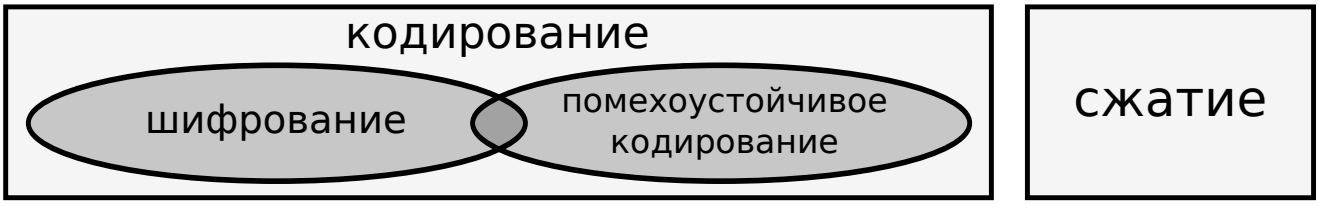

Рис. 2.1: Что есть кодирование?

Отметим здесь наличие пересечения шифрования и помехоустойчивого

[1]Эта здравая мысль в разных вариациях была высказана ещё Сократом («Давайте спорить, но прежде давайте условимся в терминах») и Р. Декартом («Определимся в терминах, и половина человеческих споров исчезнет»).

кодирования. Этот момент может служить своего рода проверкой на понимание структуры кодирования в целом. Читателю предлагается поразмыслить над данным фактом. Строго же непустота такого пересечения показана далее в замечании 1, после введения соответствующих определений и формализма.

Основные понятия

2.0.1 Канал связи

Ключевым понятием теории помехоустойчивого кодирования является *двусторонний симметричный канал связи без памяти.*

Эта абстракция описывает любые способы передачи данных, включая произвольные их композиции, а также, возможно, этапы промежуточного хранения и преобразования информации. То есть, способ «доставки» информации из точки A в точку B обобщённо называют каналом связи, независимо от его реализации или отсутствия таковой.

Нас будут интересовать свойства канала; поясним сначала явно указанные выше.

- *Двусторонний* канал подразумевает один вход и один выход. Влияние третьей стороны на процесс передачи исключено[2].

- *Симметричность* гарантирует одинаковые свойства канала в обоих направлениях[3]. Это означает, что каждый вход двустороннего канала со свойством симметричности является также и выходом.

- *Отсутствием памяти* у канала называют принципиальную невозможность буферизации передаваемых данных, а также повторной их пересылки без участия передающей стороны.

Из прочих свойств канала связи наиболее важными являются следующие два.

- *Качественный характер помех,* которым подвержен канал. Это может быть, например, искажение информации или её частичная потеря.

[2]Здесь имеется в виду активное целенаправленное вмешательство. Совокупность случайных факторов, затрудняющих передачу, принято учитывать свойством самого канала — помехоустойчивостью.

[3]Вообще говоря, симметричность не накладывает ограничения на передачу лишь в одном направлении в каждый момент времени.

- *Максимально возможный уровень помех.* Обычно задаётся верхней границей количества испорченных или потерянных сообщений.

2.0.2 Отправитель и получатель

Назовём сущности, заинтересованные в передаче и приёме информации, отправителем и получателем соответственно. Они находятся на разных концах канала связи и могут обмениваться информацией исключительно посредством него.

Предполагается, однако, что о форме представления передаваемой информации отправитель и получатель условились заранее. То есть данные могут быть поданы на вход канала в модифицированном виде, но при этом получатель в состоянии выполнить обратное преобразование.

2.0.3 Кодировщик и декодировщик

Для описания упомянутых преобразований данных вводятся посредники между отправителем и каналом, между каналом и получателем. Логично назвать их кодировщиком и декодировщиком соответственно.

Выделив таким образом функции кодирования информации в отдельные сущности, будем понимать их в дальнейшем как алгоритмы. Исследование типов подобных алгоритмов и структур данных, с которыми они работают, составляет содержание данной работы.

2.0.4 Код

Структура данных, которую формируют и поддерживают алгоритмы кодирования / декодирования и свойства которой обеспечивают возможность восстановления зашумлённой информации, называется кодом.

Важно различать код как таковой и его алгоритм. Код первичен, он может иметь несколько алгоритмических реализаций. В то время как каждому из этих алгоритмов может соответствовать целый ряд программных реализаций.

Понимание такой иерархии существенно для проведённого исследования. В литературе же не редко конкретный код подразумевает единственный или лучший алгоритм его построения и наоборот.

2.0.5 Шум

Шум является формализацией помех различной природы.

Особо отметим, что шумам подвержен лишь канал связи. Результаты работы кодировщика и декодировщика детерминированы и полностью определяются поданными на их входы данными.

2.0.6 Общая схема связи

На следующем рисунке можно проследить взаимосвязь между введёнными в предыдущих пунктах понятиями.

Рис. 2.2: Схема связи

Линейные коды

Перейдём к более строгому изложению материала.

Рассмотрим линейное пространство $\mathbb{S}^k$ размерности k. Положим его элементы векторами-строками[4] с компонентами из множества $\{0, 1\}$. Зададим операции сложения $(+)$ и умножения $(*)$ на этом множестве традиционным образом с использованием таблиц.

$+$	0	1
0	0	1
1	1	0

$*$	0	1
0	0	0
1	0	1

Легко видеть, что операции сложения и умножения в таком поле равносильны логическим XOR и AND соответственно.

Будем называть $\mathbb{S}^k$ пространством информационных слов, подлежащих передаче по каналу связи.

Определение 1 (Вес Хэмминга)**.** *Весом Хэмминга $w(x)$ вектора x называется число его ненулевых компонент.*

Определение 2 (Расстояние Хэмминга)**.** *Расстоянием Хэмминга $d(x, y)$ называется количество компонент, в которых x и y различны.*

[4]Строчная запись векторов является стандартом де-факто в теории кодирования.

С учётом таблицы сложения и определения 1, получим

$$d(x, y) \equiv w(x + y). \tag{2.1}$$

Лемма 1. *Расстояние Хэмминга d является мерой.*

Доказательство. Покажем выполнение трёх условий:

(a) $d(x, y) \geq 0, \quad d(x, y) = 0 \iff x = y,$

(b) $d(x, y) = d(y, x),$

(c) $d(x, z) \leq d(x, y) + d(y, z), \quad \forall x, y, z.$

(a) Согласно определению 2, $d(x, y) = 0$ тогда и только тогда, когда все компоненты x и y совпадают. А это и означает $x = y$.

(b) Из коммутативности операции сложения в поле и (2.1) имеем: $d(x, y) \equiv w(x + y) \equiv w(y + x) \equiv d(y, x)$.

(c) В силу определения 2, $d(x, z)$ есть минимальное количество компонент, инвертирование которых переводит x в z. В свою очередь, инвертирование $d(x, y)$ и $d(y, z)$ компонент переводит x в y и y в z соответственно. А тогда из минимальности $d(x, z)$ следует искомое $d(x, z) \leq d(x, y) + d(y, z)$.

В силу произвольности выбора x, y, z, показанное справедливо для любых x, y, z. □

Пусть $n \geq k$, тогда подпространство

$$\mathbb{D}_k^n \subset \mathbb{D}^n, \quad \dim(\mathbb{D}_k^n) = \dim(\mathbb{S}^k) = k \tag{2.2}$$

есть пространство кодовых слов.

Зададим биекцию $f : \mathbb{S}^k \leftrightarrow \mathbb{D}_k^n$. Таким образом, отображение $f : \mathbb{S}^k \to \mathbb{D}_k^n$ является этапом кодирования, поскольку переводит информационные вектора в кодовые; f^{-1} производит декодирование.

Однако рассмотрения такого отображения не достаточно, поскольку оно не учитывает возможное зашумление кодовых слов при передаче по каналу связи. Введём для этого сюръекцию $\xi : \mathbb{D}^n \to \mathbb{D}_k^n$.

Как видно из рисунка 2.3 и (2.2), пространство кодовых векторов $\mathbb{D}_k^n$ является подпространством зашумлённых векторов $\mathbb{D}^n$.

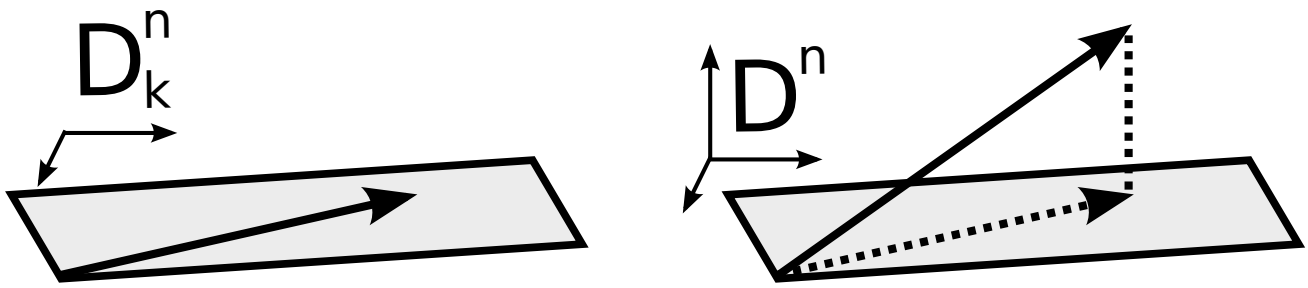

Рис. 2.3: Пространство кодовых слов: незашумлённый и зашумлённый векторы

Пусть на вход канала был подан вектор x. Наличие ошибки в принятом векторе y декодер может определить по факту $y \notin \mathbb{D}_k^n$. Таким образом, суть работы принимающей стороны описывается отображением

$$f \circ \xi : \mathbb{D}^n \to \mathbb{S}^k,$$

где $\circ$ обозначает левую композицию функций.

Здесь есть тонкий момент относительно результирующего вектора y. Казалось бы, мы пренебрегаем случаями, когда

$$y \equiv x + e \in \mathbb{D}_k^n. \tag{2.3}$$

Здесь и далее e есть вектор ошибки. Чтобы прояснить этот вопрос, введём

Определение 3 (Кодовое расстояние)**.** *Кодовым расстоянием кода* $\mathbb{S}$ *называется*

$$d(\mathbb{S}) = \min\left(d(x, y) : x, y \in \mathbb{S}, x \neq y\right). \tag{2.4}$$

Отсюда тотчас же следует, что одновременное выполнение $w(e) < d(\mathbb{D}_k^n)$ и (2.3) невозможно. Считая известной оценку максимально допустимого веса вектора ошибки (свойство помехоустойчивости канала связи), можно *гарантировать* детектирование сбоя в передаче путём выбора $\mathbb{D}_k^n : d(\mathbb{D}_k^n) > w(e_{max})$. Иначе, код с расстоянием d может обнаружить не более $d - 1$ ошибок.

Аналогичную оценку для ситуации исправления ошибок даёт

Лемма 2. *Код* $\mathbb{D}$ *с кодовым расстоянием* d *способен исправить не более* $\lfloor \frac{d-1}{2} \rfloor$ *ошибок.*

Доказательство. Пусть принят вектор $y = x + e, w(e) \leq \lfloor \frac{d-1}{2} \rfloor$. И пусть нашлось два вектора $x_1, x_2 \in \mathbb{D}$ таких, что $d(x_1, y) \leq \frac{d-1}{2}, d(y, x_2) \leq \frac{d-1}{2}$. Но тогда

по лемме 1 $d(x_1, x_2) \leq d(x_1, y) + d(y, x_2) \leq d - 1$, что противоречит предположению $d(\mathbb{D}) = d$.

С другой стороны, при передаче вектора x может произойти ситуация $w(e) = \frac{d-1}{2} + 1 = \frac{d+1}{2}$. Но тогда для $y = x + e$ найдётся $x' \in \mathbb{D} : d(x', y) = d - \frac{d+1}{2} = \frac{d-1}{2}$. То есть ошибка будет исправлена неверно. □

Определение 4. *(n,k,d)-кодом называется код* $\mathbb{D}_k^n \subset \mathbb{D}^n : d(\mathbb{D}_k^n) = d$.

Обобщим выше написанное: код длины n с k информационными компонентами и кодовым расстоянием d способен детектировать не более $d - 1$ или исправить не более $\lfloor \frac{d-1}{2} \rfloor$ ошибок.

Определение 5. *Скоростью r кода* $\mathbb{D}_k^n$ *называется отношение* k/n *количества информационных компонент к его длине.*

Как видно, скорость кода никак не связана со скоростью передачи информации по каналу связи. Первая — безразмерная величина, в то время как вторая имеет размерность *бит/с*.

2.0.7 Матричное представление

Простейшим примером кода является *код повторения*. Кодирование сообщения в таком случае заключается в многократной повторной пересылке компонент информационного слова.

Предположим, что на вход кодировщику была подана единица. Тогда для длины кода n на выходе получим $\underbrace{1 \ldots 1}_{n}$. Таким образом, код повторения длины n является $(n, 1, n)$-кодом.

Это следует из факта наличия лишь двух кодовых слов: $\underbrace{0 \ldots 0}_{n}$ и $\underbrace{1 \ldots 1}_{n}$.

Однако этап кодирования можно также удобно описать, используя матричный формализм.

$$\Big(0\Big)_{1\times1} * \Big(1 \;\; 1 \;\; \ldots \;\; 1\Big)_{1\times n} = \Big(0 \;\; 0 \;\; \ldots \;\; 0\Big)_{1\times n}$$

$$\Big(1\Big)_{1\times1} * \Big(1 \;\; 1 \;\; \ldots \;\; 1\Big)_{1\times n} = \Big(1 \;\; 1 \;\; \ldots \;\; 1\Big)_{1\times n}$$

Здесь информационная вектор-строка умножается на матрицу кода. В результате получается кодовый вектор.

Исходный вектор принадлежит пространству $\mathbb{D}_1^n$, в то время как результирующий принадлежит $\mathbb{D}_n^n \equiv \mathbb{D}^n$, что согласуется с определением $(n, 1, n)$-кода. Итак, код повторения длины n способен исправить до $\lfloor \frac{d-1}{2} \rfloor$ ошибок, принимая за верное наиболее частое значение. Так, кодовое слово 10101 будет декодировано в информационное 1, а 10001 — уже в 0.

Однако матричный подход оказывается подходящим и для более сложных случаев, в чём и заключаются его удобство и универсальность.

Рассмотрим матрицу

$$G_{2\times 3} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Она задаёт, так называемый *код чётности*. Этот $(3, 2, 2)$-код способен обнаруживать только одну ошибку, и назван так потому, что в третьем разряде кодового слова будет содержаться 0 или 1 в зависимости от чётности или нечётности суммы двух первых компонент.

$$\begin{pmatrix} 0 & 0 \end{pmatrix}_{1\times 2} * G_{2\times 3} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}_{1\times 3}$$

$$\begin{pmatrix} 0 & 1 \end{pmatrix}_{1\times 2} * G_{2\times 3} = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}_{1\times 3}$$

$$\begin{pmatrix} 1 & 0 \end{pmatrix}_{1\times 2} * G_{2\times 3} = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}_{1\times 3}$$

$$\begin{pmatrix} 1 & 1 \end{pmatrix}_{1\times 2} * G_{2\times 3} = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}_{1\times 3}$$

Возникает следующий вопрос: как проверить принадлежность принятого слова пространству кодовых слов? Ведь при больших значениях k их число может достигать 2^k, и хранить такой объём информации видится маловозможным.

Здесь оказывается уместен введённый формализм линейных пространств. Поскольку $\mathbb{D}_k^n$ является подпространством $\mathbb{D}^n$, то

$$x \cdot y = 0 \qquad \forall x \in \mathbb{D}_k^n, \quad y \in \hat{\mathbb{D}}_k^n, \tag{2.5}$$

где $\hat{\mathbb{D}}_k^n$ является ортогональным дополнением $\mathbb{D}_k^n$ до $\mathbb{D}^n$.

Легко видеть, что можно построить матрицу H из базисных векторов $\hat{\mathbb{D}}_k^n$, и тогда будет возможно проверять наличие ошибок по весу результирующего вектора

$$z \equiv x \cdot H \neq \bar{0} \quad \Leftrightarrow \quad w(z) \neq 0. \tag{2.6}$$

Такая проверка куда удобнее, чем (2.5), поскольку позволяет хранить в памяти лишь базисные вектора, количество которых, как известно, совпадает с размерностью пространства.

Определение 6 (Порождающая матрица кода)**.** *Матрица G, составленная из базисных векторов пространства $\mathbb{D}_k^n$ называется* порождающей матрицей кода.

Определение 7 (Проверочная матрица кода)**.** *Матрица H, составленная из базисных векторов пространства $\hat{\mathbb{D}}_k^n : \mathbb{D}_k^n \oplus \hat{\mathbb{D}}_k^n \equiv \mathbb{D}^n$ называется* проверочной матрицей кода $\mathbb{D}_k^n$.

Введённые обозначения G и H являются традиционными в теории кодирования.

Из (2.5) и (2.6) также следует

$$G \cdot H \equiv 0. \tag{2.7}$$

Заметим, что для приведённого выше кода чётности

$$G_{2\times 3} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad H_{3\times 1} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

В заключение пункта сделаем обещанное к рисунку 2.1

Замечание 1. *Покажем непустоту пересечения шифрования и помехоустойчивого кодирования на простом примере.*

Для начала схематично опишем процесс шифрования как перестановку $\zeta(\Omega) : \mathbb{S}^k \leftrightarrow \mathbb{S}^k$. Причём ζ^{-1} может быть легко найдена по самой ζ и набору параметров Ω. При отсутствии Ω обращение функции шифрования крайне затруднительно. Потому Ω называется ключом шифра и держится в секрете.

Теперь легко видеть, что $\zeta \circ f \circ \xi : \mathbb{S}^k \leftrightarrow \mathbb{D}^n$ обладает свойствами как помехоустойчивого кодирования, так и шифрования. Отметим, что правая композиция этим свойствам удовлетворять не будет.

Постановка задачи

Возможно, это покажется нелогичным на первый взгляд, однако краеугольным камнем теории кодирования является не создание быстрых алгоритмов (де)кодирования.

Как было замечено ранее в пункте 2.0.4, первичен сам код. А потенциальная возможность существования быстрых алгоритмов определяется уже затем его структурой.

- Количество информационных компонент k. Чем больше k, тем выше (согласно определению 5) скорость кода. То есть, тем меньшее время потребуется для передачи информации. Разумно выбирать коды с большим k.

- Длина кода n. Рассуждения для больших значений k справедливы и для малых n, с учётом того, что количество информационных компонент не может превышать общую длину кода. Иными словами, желательна минимизация n.

- В то же время, эффективность обнаружения и исправления ошибок напрямую зависит от кодового расстояния d. Однако для задания сложной структуры требуется большое число служебных компонент, что снижает скорость кода.

Прискорбно, пусть и не удивительно, однако приведённые требования противоречивы.

К счастью, в прикладных задачах хотя бы один из этих параметров известен: k при фиксированной длине сообщения (SMS), n для заданного размера ячейки данных (сектора и блоки НЖМД[5]), d в случае предсказуемого характера помех (любая стационарная сеть датчиков).

В таких случаях задачу можно переформулировать одним из следующих образов.

[5]Накопители на жёстких магнитных дисках.

- *При известном размере сообщения* k построить наиболее короткий код $\mathbb{D}_k^n, n \to \min$ с учётом $d(\mathbb{D}_k^n) = d$.

- *Для заданной длины кода* n построить код $\mathbb{D}_k^n : d(\mathbb{D}_k^n) = d$ с наибольшим количеством информационных компонент.

- *По фиксированному кодовому расстоянию* d построить код с наибольшей скоростью $k/n \to \max$.

Несмотря на наличие многих эффективных и проверенных временем кодов, вопрос построения новых по заданным параметрам (n, k, d) остаётся на сегодняшний день открытым.

Главную роль в этом играют всё более высокие запросы прикладных исследований, как то: цифровые носители информации повышенной ёмкости и надёжности, каналы связи сверхвысокой пропускной способности, передача данных исследовательскими зондами с других планет.

Основным способом — или, точнее, попыткой — решения указанной задачи является построение новых *классов* кодов, имеющих удобные для рассмотрения закономерности структуры. С одной стороны, несомненное достоинство такого подхода — возможность построения ряда кодов по общей схеме в рамках класса. С другой же стороны, существует не менее серьёзный недостаток: под различными классами кодов часто лежит совершенно, казалось бы, не связанная теория. А значит, для выбора наилучшего кода среди представителей нескольких классов необходимо иметь одновременно широкую и глубокую теоретическую базу.

Так, например, в последние годы всё чаще стали появляться публикации на тему применения в помехоустойчивом кодировании аппарата вейвлетов. Для адекватного анализа вейвлетных кодов и сравнения их с широко используемыми кодами Рида–Соломона и LDPC[6] следовало бы иметь знания и опыт работы с вейвлетами, арифметикой полиномов полей Галуа и теорией графов соответственно.

Задачей данной работы является разработка и применение методов рациональной интерполяции в задаче помехоустойчивого кодирования.

Рассмотрим задачу интерполяции для рациональных функций.

[6]Low Density Parity Check codes

Задача. Для таблицы значений переменных x и y

$$\begin{array}{c|c|c|c|c} x & x_1 & x_2 & \dots & x_N \\ \hline y & y_1 & y_2 & \dots & y_N \end{array} \quad , \quad \{x_j\}_{j=1}^N \text{ все различны}\,, \tag{2.8}$$

построить ее рациональный интерполянт, т. е. рациональную функцию такую, что

$$\{r(x_j) = y_j\}_{j=1}^N \,. \tag{2.9}$$

Здесь $r(x) = p(x)/q(x)$ при

$$p(x) = p_0 x^n + p_1 x^{n-1} + \ldots + p_n,\ q(x) = q_0 x^m + q_1 x^{m-1} + \ldots + q_m,\ p_0 \neq 0, q_0 \neq 0,$$

и $N = n + m + 1$.

Замечание 2. *В дальнейшем не будем делать различия между интерполянтами, числитель и знаменатель которых домножены на общий числовой множитель.*

Следует упомянуть о некоторых особенностях задачи рациональной интерполяции, отличающей ее от интерполяции полиномиальной. В то время как последняя всегда имеет решение, задача интерполяции рациональной не всегда разрешима в указанной постановке. Для иллюстрации этого, преобразуем (2.9) в систему уравнений

$$\{p(x_j) = y_j q(x_j)\}_{j=1}^N \,, \tag{2.10}$$

или, в развернутом виде,

$$\left\{p_n + \cdots + p_1 x_j^{n-1} + p_0 x_j^n = q_m y_j + q_{m-1} x_j y_j + \cdots + q_1 x_j^{m-1} y_j + q_0 x_j^m y_j\right\}_{j=1}^N \,, \tag{2.11}$$

которая линейна относительно $N+1$ коэффициентов $p(x)$ и $q(x)$. Конструктивная разрешимость этой системы может быть установлена средствами линейной алгебры.

Пример 1. *При* $\deg p(x) = 1, \deg q(x) = 3$ *найти рациональный интерполянт для таблицы*

$$\begin{array}{c|c|c|c|c|c} x & -1 & 0 & 1 & 2 & 3 \\ \hline y & 1 & 1 & 1/3 & 3 & 1/13 \end{array}.$$

Решение. Решая систему (2.11), получаем $p(x) \equiv x-2,\ q(x) \equiv x^3-x^2-x-2$. Однако $p(2)=0$ и $q(2)=0$, а потому условие $r(2)=3$ не выполнено. Оно не будет выполнено, даже если сократить $p(x)$ и $q(x)$ на их общий линейный делитель. □

Природа этого феномена кроется в неэквивалентности перехода от (2.9) к (2.10), поскольку для некоторого узла x_j может существовать решение линейной системы (2.11), удовлетворяющее обоим условиям: $p(x_j)=0$ и $q(x_j)=0$. Вместе с тем, в случае существования решение задачи может оказаться не единственным.

Пример 2. *Для таблицы*

x	-1	0	1	2	3
y	1	1	$1/3$	$1/7$	$1/13$

значений функции $1/(x^2+x+1)$ *существует бесконечно много интерполянтов при* $\deg p(x)=1, \deg q(x)=3$ *в форме* $(x-\lambda)/((x-\lambda)(x^2+x+1))$ *при* $\lambda \notin \{-1,0,1,2,3\}$. □

Настоящая статья посвящена подходу к решению задачи, основанному на идее, предложенной в 1846 г. Карлом Якоби [18]. Такой подход заключается в представлении полиномов числителя и знаменателя в виде определителей специального вида: так называемых *ганкелевых полиномов*. Элементами этих определителей, помимо мономов по x, являются некоторые рациональные симметрические функции элементов таблицы (2.8). Работа Якоби может считаться практически забытой в XX в.: ссылки на нее немногочисленны (и не всегда корректны). Возможно, данное обстоятельство связано с сомнениями в практической значимости предложенного подхода. Действительно, определитель большого порядка, зависящий от параметра, крайне неудобен для практических расчетов[7]. Однако же, удалось обнаружить процедуру, позволяющую обойти упомянутую проблему: для ганкелевых полиномов существует рекурсивная по порядку определителей процедура их вычисления, сводящая расчёт такого полинома k-го порядка к получению двух полиномов меньших порядков. Удивительно то, что автором идеи такой процедуры оказался тот же Якоби, а ее исчерпывающее обоснование было осуществлено его учеником Фердинандом

[7]Достаточно вспомнить проблему вычисления характеристического полинома матрицы.

Йоахимшталем [19]. Приведем этот результат в п. 5. В п. 6 излагается основной теоретический результат по разрешимости задачи рациональной интерполяции в терминах ганкелевых полиномов и на примере иллюстрируется применение результата Якоби–Йоахимшталя для получения всего семейства решений рассматриваемой задачи — при всех возможных комбинациях степеней числителя и знаменателя.

Глава 3. Обзор литературы

Теория кодирования — довольно молодая наука, и классические труды по ней были написаны лишь в XX веке.

Прежде всего, это работы уже упомянутых К. Шеннона [28] и Р. Хэмминга [11, 12]. В качестве подтверждения классического статуса данных работ можно привести тот факт, что коды Хэмминга и по сей день не только являются базовым построением большинства теоретических курсов кодирования [27, 34], но и находят применение в современных высокотехнологичных решениях [23, 29].

Целая веха также связана с фамилиями Боуза, Чоудхури, Хоквингема[1] [7, 14] (далее – БЧХ) и появлением одноимённого класса кодов. Эти коды получили широкое распространение благодаря удобству построения [14] и скорости декодирования [7], породив впоследствии более узкие подклассы кодов со значительно лучшими свойствами в пределах области их применимости.

К 80-м годам теоретические основы помехоустойчивого кодирования были сформированы. С практической стороны это подтверждается успешным использованием методов исправления ошибок в ходе космических программ Mariner 9 (1971) и Voyager (1977)[2]. Исчерпывающее описание состояния теории кодирования на тот момент дают монографии [2, 5].

В более поздний период работ столь широкого профиля не наблюдается. Это можно объяснить явно оформившимся к тому времени разделением кодирования на две части: сугубо практическую, обусловленную появлением доступных вычислительных машин, и теоретическую. В первом случае внимание преимущественно уделялось модификации зарекомендовавших себя кодов и разработке эффективных алгоритмов их реализации на существующих аппаратных архитектурах [5, 31]; во втором – построению новых субоптимальных кодов, приближающихся к указанной Р. Хэммингом теоретической границе [3, 10].

[1] Bose R.C., Ray-Chaudhury D.K., Hocquenghem A.

[2] Были использованы коды Адамара (Hadamard [32, 6, 16]) и Голея (Golay [23, 12, 7]) соответственно.

В настоящее время коды, управляющие ошибками, вышли далеко за границы высоконадёжных и высокотехнологичных систем. Так, коды Рида–Соломона используются в двумерных штрих-кодах (QR code) [15, 25], вариации контрольных сумм «чётности» используются при формировании международного стандартного книжного номера[3] (ISBN) [16].

Что касается интерполяции вообще и рациональной интерполяции в частности, здесь в первую очередь стоит отметить Коши [8], аппарат которого послужил прекрасным примером одновременной изящности и строгости математических выкладок. Его результаты в дальнейшем были развиты Кронекером [21, 22] и Нетто [24].

Интерполяция многочленами специальных видов, как-то: Ганкеля и Лагранжа – подробна рассмотрена в следующих публикациях [20, 26, 33].

[3]International Standard Book Number

Глава 4. Коды, контролирующие ошибки

Типы кодов и их свойства

Рис. 4.1: Типы кодов

В помехоустойчивом кодировании традиционно рассматривается два типа кодов: свёрточные и блочные.

Свёрточные коды применяются, когда данные поступают непрерывным потоком, и важно обрабатывать их в режиме реального времени. Принцип их работы заключается в кодировании каждой компоненты с учётом нескольких предыдущих. Такой метод хорошо подходит, например, для передачи информации по сети.

Блочные же коды, как можно догадаться из названия, преобразуют целиком набор информационных слов фиксированного размера; причём необходимым условием является наличие всего блока информации на момент начала кодирования.

Далее в работе будут рассматриваться только блочные коды.

Большое значение для свойств кода имеет количество его кодовых слов. Это следует из взаимооднозначного соответствия информационных и кодовых векторов. При недостаточном количестве кодовых слов, отправитель будет вынужден использовать более длинный (а значит, и более медленный) код.

Существует верхняя граница количества кодовых слов для кода длины n, введённая Р. Хэммингом и названная его именем.

Лемма 3 (Граница Хэмминга). *Число кодовых слов $(n, *, d)$-кода не превосходит*

$$\frac{2^n}{\sum_{j=1}^{\frac{d-1}{2}} C_n^j},$$

где C_n^j — биномиальные коэффициенты.

Доказательство этой леммы можно найти в книге [5].

Коды, достигающие этой границы называются *совершенными*. Поиск таких кодов является открытой задачей. Известными на сегодняшний день совершенными кодами являются (23,12,7)-код Голея и (7,4,3)-код Хэмминга.

Последний из них, в силу его исторической значимости и при этом актуальности [23], будет рассмотрен подробнее.

Коды Хэмминга

Коды Хэмминга — это целый класс кодов сходной структуры. Тем не менее, в своей работе [11] он описывает только $(7, 4, 3)$-код, за которым и закрепилось название *код Хэмминга*.

Здесь мы рассмотрим более подробно $(7, 4, 3)$-код и затем укажем общий способ построения более длинных кодов.

4.0.1 $(7, 4, 3)$-код Хэмминга

Код Хэмминга задаётся следующей матрицей

$$G_{4\times 7} = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right).$$

Составляя всевозможные линейные комбинации базисных векторов порождающей матрицы, приведём таблицу всех кодовых слов. В таблице 4.1 $i_j, j = \overline{0,3}$ обозначают информационные разряды, $s_j, j = \overline{0,2}$ — служебные.

Прежде всего, покажем, что код Хэмминга является совершенным.

Из таблицы 4.1 видно, что количество кодовых векторов равно 16, что с

№	i_0	i_1	i_2	i_3	s_0	s_2	s_3
0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1
2	0	0	1	0	1	1	0
3	0	0	1	1	1	0	1
4	0	1	0	0	1	1	1
5	0	1	0	1	1	0	0
6	0	1	1	0	0	0	1
7	0	1	1	1	0	1	0
8	1	0	0	0	1	0	1
9	1	0	0	1	1	1	0
10	1	0	1	0	0	1	1
11	1	0	1	1	0	0	0
12	1	1	0	0	0	1	0
13	1	1	0	1	0	0	1
14	1	1	1	0	1	0	0
15	1	1	1	1	1	1	1

Таблица 4.1: Кодовые слова $(7, 4, 3)$-кода Хэмминга

равенством удовлетворяет введённой ранее в лемме 3 границе Хэмминга:

$$16 = \frac{2^{n[=7]}}{C^0_{n[=7]} + C^1_{n[=7]}} = \frac{2^7}{1+7} = \frac{2^7}{2^3} = 2^4 = 16.$$

Здесь суммирование биномиальных коэффициентов оборвано на C^1_7 потому, что в соответствии с леммой 2 $(7, 4, 3)$-код способен исправить не более $\lfloor \frac{3-1}{2} \rfloor = 1$ ошибки.

По правой части матрицы G видно, что служебные компоненты задаются следующими соотношениями

$$s_0 = i_0 + i_1 + i_2,$$
$$s_1 = i_1 + i_2 + i_3,$$
$$s_2 = i_0 + i_1 + i_3.$$

Из (2.7) можно найти проверочную матрицу

$$H_{3\times 7} = \left(\begin{array}{cccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right).$$

4.0.2 Общий случай

Для построения более длинных кодов Хэмминга заметим следующий факт: правый блок матрицы G является транспонированным левым блоком матрицы H.

Таким образом, введя обозначения

$$H = \left[\begin{array}{c|c} P & E_M \end{array}\right], \quad G = \left[\begin{array}{c|c} E_{N-M} & P^T \end{array}\right], \quad E_k = \left[\begin{array}{cccc} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & 1 \end{array}\right]_{k\times k}, \qquad (4.1)$$

можно получить прямое соответствие между порождающей и проверочной матрицами кода.

Использованные параметры N и M выбирают исходя из условий

$$N = 2^M - 1, \quad n = 2^M - 1, \quad k = 2^M - M - 1. \qquad (4.2)$$

Подробно этот момент рассмотрен в [34]. Мы же не будем останавливаться на данном вопросе, так как работа не претендует на новшества в конкретной области.

Итак, возвращаясь к построению длинных кодов Хэмминга, можно видеть, что задача сводится к выбору подходящей матрицы P и / или одной из матриц G и H.

Приведём формальный алгоритм, строящий коды Хэмминга.

- Зафиксировать значения параметров $N, M : N > M$ в соответствии с (4.2).
- Построить невырожденную матрицу $P_{M\times(N-M)}$, не содержащую строк единичной матрицы соответствующей размерности.
- Достроить P^T и P до порождающей и проверочной матриц, исходя из

условий (4.1)

Найденные таким образом матрицы G и H будут задавать базисы пространства кодовых слов и ортогонального дополнения к нему соответственно.

Коды Рида–Соломона

Коды Рида–Соломона основаны на аппарате полиномов полей Галуа. Совершим небольшое введение в предметную область, необходимое для построения данного класса кодов.

4.0.3 Арифметика полиномов в конечных полях

Как и ранее, будем придерживаться бинарной арифметики, а потому будем рассматривать поля вида $GF(2^m)$.

Введём необходимые определения.

Определение 8 (Примитивный элемент)**.** *Примитивным элементом* $\mathfrak{a}$ *поля* $GF(2^m)$ *называется первообразный корень из единицы степени* $2^m - 1$*.*

Заметим, что существование единицы (равно как и нулевого элемента) гарантируется свойствами поля.

Существенным свойством примитивного элемента является то, что при возведении в степень он пробегает все ненулевые элементы поля. Таким образом, все элементы конечного поля возможно занумеровать, поставив каждому из них в соответствие некоторый номер степени примитивного элемента.

Определение 9 (Порядок элемента)**.** *Порядком элемента* $\mathfrak{b} \in GF(2^m)$ *называется такое число* j*, что* $\mathfrak{a}^j = \mathfrak{b}$*, где* $\mathfrak{a}$ *— примитивный элемент поля.*

Определение 10 (Неприводимый полином)**.** *Полином* $g(x)$ *называется* неприводимым *над полем, если он не может быть представлен в виде произведения двух других полиномов над тем же полем.*

Как показано в [27], конечное поле может быть задано при помощи неприводимого полинома. Операции сложения и умножения в таком случае задаются аналогично сложению и умножению в бесконечных полях, с той лишь разницей, что результат берётся по модулю выбранного неприводимого полинома.

Определение 11 (Генерирующий полином)**.** *Неприводимый полином $g(x)$, по модулю которого производится операция умножения в поле, называется* генерирующим *полиномом поля.*

Такой формализм позволяет нам сопоставить каждому полиному вектор его коэффициентов и наоборот. То есть, информационные вектора теперь есть также и информационные полиномы.

Пример 3 ($GF(2^4)$)**.** *Приведём пример такого соответствия для поля, генерируемого полиномом $g(x) = x^4 + x + 1$.*

$$x^3 + x + 1 \leftrightarrow (1011)$$
$$x^2 + x \leftrightarrow (0110)$$

4.0.4 Построение кодов Рида–Соломона

Как и в случае кода Хэмминга, ограничимся схематичным описанием процесса построения кода, достаточным для указания его свойств и последующего анализа.

Основная идея данного класса кодов заключается в использовании целых информационных блоков вместо единичных компонент. То есть, информационный вектор отправителя будет состоять не из скаляров, а из полиномов, принадлежащих некоторому полю $GF(2^m)$.

Таким образом, повреждение сразу нескольких бит внутри одного полинома будет считаться единичной ошибкой, и может быть исправлено декодером за один проход.

Кодирование. Коды Рида–Соломона обладают очень значимым свойством лёгкости получения кодового вектора из информационного. При всей своей кажущейся сложности, процесс кодирования сводится к умножению информационного полинома $f(x)$ на генерирующий $g(x)$. Полученное произведение и есть кодовый вектор.

Однако мы до сих пор не указали критериев выбора генерирующего полинома. Остановимся на этом подробнее.

Поскольку генерирующий полином определяет процессы кодирования и декодирования, то он выбирается таким образом, чтобы облегчить детектиро-

вание ошибок. А именно,

$$g(x) = (x - \mathfrak{a})(x - \mathfrak{a}^2)(x - \mathfrak{a}^3)\dots(x - \mathfrak{a}^{2\nu}). \tag{4.3}$$

Количество корней 2ν выбрано для удобства дальнейшего изложения.

При таком выборе, кодовый полином

$$G(x) \equiv f(x)g(x) \tag{4.4}$$

также будет иметь корнями $\mathfrak{a}, \mathfrak{a}^2, \mathfrak{a}^3, \dots, \mathfrak{a}^{2\nu}$, в силу (4.3).

Декодирование. Процедура декодирования кодов Рида–Соломона делится на две части: восстановление (расшумление) принятого кодового полинома и нахождение исходного информационного вектора.

Начнём описание со второй части, как более простой.

Принимая во внимание (4.4), логично проводить декодирование следующим образом:

$$f(x) = G(x)/g(x).$$

Однако, вообще говоря, такая запись не верна. Правильно будет искать информационный полином в виде

$$f(x) = G(x) \cdot g^{-1}(x).$$

Так как генерирующий полином $g(x)$ известен на этапах кодирования и декодирования и не меняется, есть потенциальная возможность предвычислить его обратный полином $g^{-1}(x)$ и использовать эту информацию для ускорения декодирования.

Такой подход является перспективным и актуальным. На эту тему автором данной работы было проведено исследование и написана статья [31], раскрывающая затронутый вопрос в деталях.

Перейдём к случаю расшумления принятого полинома.

Пусть на выходе канала связи оказался полином

$$\widetilde{G}(x) \neq G(x).$$

Тогда, согласно (4.3), $\widetilde{G}(x)$ будет принимать ненулевые значения в корнях генерирующего полинома $g(x)$, то есть

$$\begin{aligned} S_0 &= \widetilde{G}(\mathfrak{a}), \\ S_1 &= \widetilde{G}(\mathfrak{a}^2), \\ &\dots \\ S_{2\nu-1} &= \widetilde{G}(\mathfrak{a}^{2\nu}). \end{aligned}$$

Величины S_j называются *синдромами* полинома $\widetilde{G}(x)$. Они пока не дают сведений о позиции и величине ошибки передачи. Тем не менее, синдромы могут быть использованы для построения *полинома локаторов ошибок*.

Корнями такого полинома являются локаторы ошибок $\mathfrak{e}$, которые задаются в степенном виде $\mathfrak{e} = \mathfrak{a}^q$, в соответствии с определением 9.

Зная локатор (то есть, местоположение) ошибки и найдя её величину, мы восстанавливаем переданный полином $G(x)$ из полученного $\widetilde{G}(x)$.

Задача построения полинома локаторов ошибок формулируется аналогично задаче интерполяции в конечных полях. Другими словами, её можно задать при помощи таблицы.

x	$\mathfrak{a}^1$	$\mathfrak{a}^2$	$\dots$	$\mathfrak{a}^{2\nu}$
y	S_0	S_1	$\dots$	$S_{2\nu-1}$

Таблица 4.2: Интерполяционная таблица полинома локаторов ошибок

Здесь 2ν есть количество корней генерирующего полинома из (4.3).

Эффективный алгоритм интерполяции в конечных полях, применимый в конкретном случае, был рассмотрен автором данной работы в статье [1]. Он сводится к вычислению определителя суперсимметричной матрицы, стоящей в левой части уравнения

$$\begin{pmatrix} S_0 & S_1 & \cdots & S_{\nu-1} \\ S_1 & S_2 & \cdots & S_{\nu} \\ & & \ddots & \\ S_{\nu-1} & S_{\nu} & \cdots & S_{2\nu-2} \end{pmatrix} \begin{pmatrix} \Lambda_{\nu} \\ \Lambda_{\nu-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} S_{\nu} \\ S_{\nu+1} \\ \vdots \\ S_{2\nu-1} \end{pmatrix},$$

где неизвестные $\Lambda_1 \dots \Lambda_\nu$ суть искомые коэффициенты полинома локаторов ошибок.

Глава 5. Ганкелевы определители и полиномы.

Для числовой последовательности (конечной или бесконечной)

$$\{c_j\}_{j=0}^{\infty} = \{c_0, c_1, \dots\} \tag{5.1}$$

матрица вида

$$[c_{i+j-2}]_{i,j=1}^{k} = \begin{bmatrix} c_0 & c_1 & c_2 & \dots & c_{k-1} \\ c_1 & c_2 & c_3 & \dots & c_k \\ \vdots & \vdots & & & \vdots \\ c_{k-2} & c_{k-1} & c_k & \dots & c_{2k-3} \\ c_{k-1} & c_k & c_{k+1} & \dots & c_{2k-2} \end{bmatrix}_{k\times k} \tag{5.2}$$

называется **ганкелевой матрицей** порядка k, порожденной последовательностью (5.1). Ее определитель будем обозначать $H_k(\{c\})$, или просто H_k, если это не вызывает путаницы.

Если заменим последнюю строку ганкелевой матрицы порядка $k+1$ строкой, составленной из степеней x, то соответствующий определитель

$$\mathcal{H}_k(x;\{c\}) = \begin{vmatrix} c_0 & c_1 & c_2 & \dots & c_k \\ c_1 & c_2 & c_3 & \dots & c_{k+1} \\ \vdots & \vdots & & & \vdots \\ c_{k-1} & c_k & c_{k+1} & \dots & c_{2k-1} \\ 1 & x & x^2 & \dots & x^k \end{vmatrix}_{(k+1)\times(k+1)} \equiv \tag{5.3}$$

$$\equiv (-1)^k \begin{vmatrix} c_1 - c_0 x & c_2 - c_1 x & \dots & c_k - c_{k-1}x \\ c_2 - c_1 x & c_3 - c_2 x & \dots & c_{k+1} - c_k x \\ \vdots & & \ddots & \vdots \\ c_k - c_{k-1}x & c_{k+1} - c_k x & \dots & c_{2k-1} - c_{2k-2}x \end{vmatrix}_{k\times k},$$

или просто $\mathcal{H}_k(x)$, называется k-м **ганкелевым полиномом** [13], порожденным последовательностью (5.1). Разложение определителя (5.3) по его последней строке дает

$$\mathcal{H}_k(x) \equiv h_{k0}x^k + h_{k1}x^{k-1} + h_{k2}x^{k-2} + \dots \quad \text{при} \quad h_{k0} = H_k\,.$$

Таким образом, $\deg \mathcal{H}_k(x) = k$ тогда и только тогда, когда $H_k \neq 0$.

Теорема 1 (Якоби, Йоахимшталь)**.** *Любые три последовательных ганкелевых полинома $\mathcal{H}_{k-2}(x), \mathcal{H}_{k-1}(x), \mathcal{H}_k(x)$ связаны соотношением:*

$$H_k^2\mathcal{H}_{k-2}(x) + (H_k h_{k-1,1} - H_{k-1}h_{k1} - H_k H_{k-1}x)\,\mathcal{H}_{k-1}(x) + H_{k-1}^2\mathcal{H}_k(x) \equiv 0\,. \quad (5.4)$$

Доказательство.[1]. Рассмотрим сначала случай, когда порождающая последовательность (5.1) задается в виде

$$\left\{ c_j = \sum_{\ell=1}^{m} \lambda_\ell^j \right\}_{j=0}^{2k-1} \quad (5.5)$$

при $m > k$ и произвольных различных $\lambda_1, \dots, \lambda_m$. Докажем справедливость следующих соотношений:

$$\sum_{\ell=1}^{m} \lambda_\ell^j \mathcal{H}_k(\lambda_\ell) = \begin{cases} 0, & \text{если } j \in \{0, \dots, k-1\}, \\ H_{k+1}, & \text{если } j = k\,. \end{cases} \quad (5.6)$$

Действительно,

$$\sum_{\ell=1}^{m} \lambda_\ell^j \mathcal{H}_k(\lambda_\ell) = \sum_{\ell=1}^{m} \lambda_\ell^j \begin{vmatrix} c_0 & c_1 & \dots & c_k \\ c_1 & c_2 & \dots & c_{k+1} \\ \vdots & \vdots & & \vdots \\ c_{k-1} & c_k & \dots & c_{2k-1} \\ 1 & \lambda_\ell & \dots & \lambda_\ell^k \end{vmatrix} = \begin{vmatrix} c_0 & c_1 & \dots & c_k \\ c_1 & c_2 & \dots & c_{k+1} \\ \vdots & \vdots & & \vdots \\ c_{k-1} & c_k & \dots & c_{2k-1} \\ c_j & c_{j+1} & \dots & c_{j+k} \end{vmatrix}.$$

При $j < k$ последний определитель обращается в нуль, поскольку содержит две одинаковые строки. При $j = k$ он совпадает с H_{k+1}.

[1]Приводимое доказательство представляет собой переписанное в современных обозначениях оригинальное доказательство из [19].

Пусть $H_{k-1} \neq 0$ (т. е. $\deg \mathcal{H}_{k-1}(x) = k-1$). Разделим $\mathcal{H}_k(x)$ на $\mathcal{H}_{k-1}(x)$:

$$\mathcal{H}_k(x) \equiv Q(x)\mathcal{H}_{k-1}(x) + R(x)\,. \tag{5.7}$$

Здесь коэффициенты частного выражаются через коэффициенты $\mathcal{H}_k(x)$ и $\mathcal{H}_{k-1}(x)$:

$$Q(x) = Q_0 + Q_1 x \quad \text{при} \;\; Q_1 = \frac{H_k}{H_{k-1}}\,,\; Q_0 = \frac{H_{k-1}h_{k1} - H_k h_{k-1,1}}{H_{k-1}^2}\,. \tag{5.8}$$

Для нахождения коэффициентов остатка $R(x) = R_0 + R_1 x + \cdots + R_{k-2}x^{k-2}$ подставим $x = \lambda_1, \ldots, x = \lambda_m$ в (5.7):

$$\left\{\mathcal{H}_k(\lambda_\ell) = (Q_0 + Q_1\lambda_\ell)\,\mathcal{H}_{k-1}(\lambda_\ell) + \left(R_0 + R_1\lambda_\ell + \cdots + R_{k-2}\lambda_\ell^{k-2}\right)\right\}_{\ell=1}^{m}\,. \tag{5.9}$$

Суммирование этих равенств дает

$$\sum_{\ell=1}^{m}\mathcal{H}_k(\lambda_\ell) = Q_0\sum_{\ell=1}^{m}\mathcal{H}_{k-1}(\lambda_\ell) + Q_1\sum_{\ell=1}^{m}\lambda_\ell\mathcal{H}_{k-1}(\lambda_\ell) + (c_0R_0 + c_1R_1 + \cdots + c_{k-2}R_{k-2})\,.$$

Согласно (5.6), получим

$$0 = c_0R_0 + c_1R_1 + \cdots + c_{k-2}R_{k-2}\,.$$

Далее умножим каждое равенство (5.9) на соответствующие λ_ℓ^j и просуммируем полученные равенства по ℓ. При $j \in \{1, \ldots, k-3\}$ приходим к равенствам

$$0 = c_jR_0 + c_{j+1}R_1 + \cdots + c_{j+k-2}R_{k-2}\,.$$

При $j = k-2$ имеем равенство слегка иного вида:

$$0 = H_kQ_1 + c_{k-2}R_0 + c_{k-1}R_1 + \cdots + c_{2k-4}R_{k-2}\,.$$

Объединяя полученные равенства в систему, рассматриваем ее как линейную относительно $R_0, \ldots, R_{k-2}$ и разрешаем по формулам Крамера. Результат может быть записан как тождество

$$R(x)H_{k-1} + H_kQ_1\mathcal{H}_{k-2}(x) \equiv 0\,,$$

которое вместе с выражением (5.8) для Q_1 подтверждает истинность (5.4) для частного случая порождающей последовательности, заданной (5.5).

Рассмотрим теперь случай произвольной порождающей последовательности (5.1). Для любой заданной последовательности комплексных чисел $c_1, \ldots, c_{2k-1}$ возможно отыскать комплексные числа $\lambda_1, \ldots, \lambda_\ell$ такие, что при $\ell > 2k-1$ уравнения (5.5) будут совместными. Эти числа могут быть найдены как корни полинома степени ℓ, первые $2\,k-1$ сумм Ньютона [9] которого совпадают с $\{c_j\}_{j=1}^{2k-1}$.

Для завершения доказательства следует заполнить пробел в аргументации предыдущего абзаца. В то время как числа $c_1, \ldots, c_{2k-1}$ могут быть выбраны произвольными, число c_0 оказывается целым положительным, а именно равным ℓ. Таким образом, истинность (5.4) доказана лишь для $c_0 \in \mathbb{N}$. Однако доказываемое равенство является алгебраическим относительно $\{c_j\}_{j=0}^{2k-1}$ и, будучи выполненным для бесконечного набора целых чисел, должно выполняться для любого $c_0 \in \mathbb{C}$. □

Тождество (5.4) позволяет организовать рекурсивную процедуру вычисления ганкелевых полиномов. В самом деле, предположим, что канонические представления полиномов $\mathcal{H}_{k-2}(x)$ и $\mathcal{H}_{k-1}(x)$ уже найдены:

$$\mathcal{H}_{k-1}(x) \equiv h_{k-1,0}x^{k-1} + h_{k-1,1}x^{k-2} + \cdots + h_{k-1,k-1} \quad \text{при} \quad h_{k-1,0} = H_{k-1}\,.$$

В таком случае в (5.4) известны значения почти всех констант, за исключением H_k и h_{k1}. Для последних имеются лишь детерминантные представления:

$$H_k = \begin{vmatrix} c_0 & c_1 & \ldots & c_{k-2} & c_{k-1} \\ c_1 & c_2 & \ldots & c_{k-1} & c_k \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k-2} & c_{k-1} & \ldots & c_{2k-4} & c_{2k-3} \\ c_{k-1} & c_k & \ldots & c_{2k-3} & c_{2k-2} \end{vmatrix}, \; h_{k1} = - \begin{vmatrix} c_0 & c_1 & \ldots & c_{k-2} & c_k \\ c_1 & c_2 & \ldots & c_{k-1} & c_{k+1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k-2} & c_{k-1} & \ldots & c_{2k-4} & c_{2k-2} \\ c_{k-1} & c_k & \ldots & c_{2k-3} & c_{2k-1} \end{vmatrix}.$$

Эти определители отличаются от транспонированного детерминантного представления $\mathcal{H}_{k-1}(x)$ только последними столбцами. Разложения по элементам последних столбцов имеют одинаковые значения для соответствующих алгеб-

раических дополнений, и потому формулы

$$\begin{aligned} h_{k0} = H_k &= c_{k-1}h_{k-1,k-1} + c_k h_{k-1,k-2} + \cdots + c_{2k-2}h_{k-1,0}, \\ h_{k1} &= -(c_k h_{k-1,k-1} + c_{k+1}h_{k-1,k-2} + \cdots + c_{2k-1}h_{k-1,0}) \end{aligned} \tag{5.10}$$

позволяют выразить h_{k0} и h_{k1} посредством уже известных коэффициентов полинома $\mathcal{H}_{k-1}(x)$.

Однако предложенный алгоритм рекурсивного вычисления $\mathcal{H}_k(x)$ не работает в случае, когда $H_{k-1} = 0$. Модификация процедуры может быть осуществлена с помощью следующего результата.

Теорема 2. *Пусть* $H_{k-2} \neq 0,\ H_{k-1} = 0$. *Если* $h_{k-1,1} = 0$, *то*

$$\mathcal{H}_{k-1}(x) \equiv 0 \quad \textit{и} \quad \mathcal{H}_k(x) \equiv \frac{h_{k2}}{H_{k-2}}\mathcal{H}_{k-2}(x)\,.$$

В противном случае

$$\mathcal{H}_{k-1}(x) \equiv \frac{h_{k-1,1}}{H_{k-2}}\mathcal{H}_{k-2}(x) \tag{5.11}$$

и

$$\mathcal{H}_k(x) \equiv \frac{H_k H_{k-2} h_{k-1,1}\mathcal{H}_{k-3}(x) - \begin{vmatrix} H_{k-2} & 0 & 0 & H_k \\ h_{k-2,1} & H_{k-2} & 0 & h_{k1} \\ h_{k-2,2} & h_{k-2,1} & H_{k-2} & h_{k2} \\ x^2 & x & 1 & 0 \end{vmatrix} \mathcal{H}_{k-2}(x)}{H_{k-2}^3}\,. \tag{5.12}$$

Замечание 3. *Формулы теоремы 2 — (5.11) и (5.12) — позволяют производить рекурсивное вычисление* $\mathcal{H}_k(x)$, *когда полиномы* $\mathcal{H}_{k-2}(x)$ *и* $\mathcal{H}_{k-3}(x)$ *уже посчитаны. Участвующие в этих формулах константы, такие как* $h_{k-2,1}$, $h_{k-2,2}$, $h_{k-1,1}$ *и* $h_{k,1}$, *либо считаются известными как коэффициенты ганкелевых полиномов, либо могут быть вычислены при помощи формул (5.10). Единственным исключением является* h_{k2}. *Для вычисления этой величины предлагается использовать формулу*

$$h_{k2} = -\frac{\left(c_{2k-2}h_{k-2,0} + c_{2k-3}h_{k-2,1} + \cdots + c_k h_{k-2,k-2}\right)^2}{H_{k-2}}\,,$$

которая справедлива при $H_{k-1} = 0,\ H_{k-2} \neq 0$.

Замечание 4. *Фактически, формулу (5.4) следует воспринимать в качестве первоисточника алгоритма, известного в настоящее время как* ***алгоритм Берлекампа–Месси*** *[5]; он был предложен для декодирования кодов Боуза–Чоудхури–Хоквингема и Рида–Соломона, а также для нахождения минимального полинома линейной рекуррентной последовательности.*

Глава 6. Рациональная интерполяция.

Основной теоретический результат статьи предварим следующей леммой [9]:

Лемма 4. *Обозначим* $W(x) = \prod_{j=1}^{N}(x - x_j)$. *Справедливы следующие равенства Эйлера–Лагранжа:*

$$\sum_{j=1}^{N} \frac{x_j^k}{W'(x_j)} = \begin{cases} 0, & \text{если } k \in \{1, \ldots, N-2\}, \\ 1, & \text{если } k = N-1. \end{cases} \tag{6.1}$$

Теорема 3. *Пусть* $\{y_j \neq 0\}_{j=1}^{N}$. *Вычислим последовательности*

$$\left\{\tau_k = \sum_{j=1}^{N} y_j \frac{x_j^k}{W'(x_j)}\right\}_{k=0}^{2m} \quad \text{и} \quad \left\{\widetilde{\tau}_k = \sum_{j=1}^{N} \frac{1}{y_j} \frac{x_j^k}{W'(x_j)}\right\}_{k=0}^{2n-2} \tag{6.2}$$

и построим соответствующие им ганкелевы полиномы $\mathcal{H}_m(x; \{\tau\})$ *и* $\mathcal{H}_n(x; \{\widetilde{\tau}\})$. *Если*

$$H_n(\{\widetilde{\tau}\}) \neq 0 \tag{6.3}$$

и

$$\{\mathcal{H}_m(x_j; \{\tau\}) \neq 0\}_{j=1}^{N}, \tag{6.4}$$

то существует единственное решение задачи рациональной интерполяции при $\deg p(x) = n, \deg q(x) \leq m = N - n - 1$. *Это решение можно представить в виде*

$$p(x) = H_{m+1}(\{\tau\})\mathcal{H}_n(x; \{\widetilde{\tau}\}) = \begin{vmatrix} \tau_0 & \tau_1 & \ldots & \tau_m \\ \tau_1 & \tau_2 & \ldots & \tau_{m+1} \\ \vdots & \vdots & & \vdots \\ \tau_{m-1} & \tau_m & \ldots & \tau_{2m-1} \\ \tau_m & \tau_{m+1} & \ldots & \tau_{2m} \end{vmatrix} \cdot \begin{vmatrix} \widetilde{\tau}_0 & \widetilde{\tau}_1 & \ldots & \widetilde{\tau}_n \\ \widetilde{\tau}_1 & \widetilde{\tau}_2 & \ldots & \widetilde{\tau}_{n+1} \\ \vdots & \vdots & & \vdots \\ \widetilde{\tau}_{n-1} & \widetilde{\tau}_n & \ldots & \widetilde{\tau}_{2n-1} \\ 1 & x & \ldots & x^n \end{vmatrix}, \tag{6.5}$$

$$q(x) = H_n(\{\widetilde{\tau}\})\mathcal{H}_m(x;\{\tau\}) = \begin{vmatrix} \widetilde{\tau}_0 & \widetilde{\tau}_1 & \dots & \widetilde{\tau}_{n-1} \\ \widetilde{\tau}_1 & \widetilde{\tau}_2 & \dots & \widetilde{\tau}_n \\ \vdots & \vdots & & \vdots \\ \widetilde{\tau}_{n-1} & \widetilde{\tau}_n & \dots & \widetilde{\tau}_{2n-2} \end{vmatrix} \cdot \begin{vmatrix} \tau_0 & \tau_1 & \dots & \tau_m \\ \tau_1 & \tau_2 & \dots & \tau_{m+1} \\ \vdots & \vdots & & \vdots \\ \tau_{m-1} & \tau_m & \dots & \tau_{2m-1} \\ 1 & x & \dots & x^m \end{vmatrix}. \tag{6.6}$$

Доказательство. Если решение задачи существует, то равенства (2.11) справедливы. Домножим j-е равенство на $x_j^k/W'(x_j)$ при $k \in \{0,\dots,m-1\}$ и просуммируем полученные равенства по j. На основании равенств (6.1) приходим к системе уравнений

$$\{q_m\tau_k + q_{m-1}\tau_{k+1} + \dots + q_1\tau_{k+m-1} + q_0\tau_{k+m} = 0\}_{k=0}^{m-1}\ .$$

Таким образом, знаменатель дроби должен удовлетворять соотношению

$$Aq(x) \equiv \begin{vmatrix} \tau_0 & \tau_1 & \dots & \tau_m \\ \tau_1 & \tau_2 & \dots & \tau_{m+1} \\ \vdots & \vdots & & \vdots \\ \tau_{m-1} & \tau_m & \dots & \tau_{2m-1} \\ 1 & x & \dots & x^m \end{vmatrix}$$

при некоторой константе A.

Подобным образом, умножая равенства (2.11) на $x_j^\ell/(y_jW'(x_j))$ при $\ell \in \{0,\dots,n-1\}$ и суммируя по j, получим представление числителя в виде

$$Bp(x) \equiv \begin{vmatrix} \widetilde{\tau}_0 & \widetilde{\tau}_1 & \dots & \widetilde{\tau}_n \\ \widetilde{\tau}_1 & \widetilde{\tau}_2 & \dots & \widetilde{\tau}_{n+1} \\ \vdots & \vdots & & \vdots \\ \widetilde{\tau}_{n-1} & \widetilde{\tau}_n & \dots & \widetilde{\tau}_{2n-1} \\ 1 & x & \dots & x^n \end{vmatrix} \equiv H_n(\{\widetilde{\tau}\})x^n + \dots$$

при некоторой константе B. Согласно (6.3), имеем: $B \neq 0$ и $\deg p(x) = n$.

Для нахождения множителей A и B подставим полученные выражения в (2.10):

$$\{A\mathcal{H}_n(x_j;\{\widetilde{\tau}\}) = By_j\mathcal{H}_m(x_j;\{\tau\})\}_{j=1}^N\ . \tag{6.7}$$

Согласно (6.4), $A \neq 0$ и $\{\mathcal{H}_n(x_j;\{\widetilde{\tau}\}) \neq 0\}_{j=1}^N$. Домножим каждое из равенств (6.7) на $x_j^m/W'(x_j)$ и просуммируем получившиеся результаты. В силу свойства линейности определителя и с использованием (6.1), имеем

$$\sum_{j=1}^N \frac{\mathcal{H}_n(x_j;\{\widetilde{\tau}\})x_j^m}{W'(x_j)} = \begin{vmatrix} \widetilde{\tau}_0 & \widetilde{\tau}_1 & \dots & \widetilde{\tau}_{n-1} & \widetilde{\tau}_n \\ \widetilde{\tau}_1 & \widetilde{\tau}_2 & \dots & \widetilde{\tau}_n & \widetilde{\tau}_{n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ \widetilde{\tau}_{n-1} & \widetilde{\tau}_n & \dots & \widetilde{\tau}_{2n-2} & \widetilde{\tau}_{2n-1} \\ \displaystyle\sum_{j=1}^N \frac{x_j^m}{W'(x_j)} & \displaystyle\sum_{j=1}^N \frac{x_j^{m+1}}{W'(x_j)} & \dots & \displaystyle\sum_{j=1}^N \frac{x_j^{m+n-1}}{W'(x_j)} & \displaystyle\sum_{j=1}^N \frac{x_j^{m+n}}{W'(x_j)} \end{vmatrix} =$$

$$= \begin{vmatrix} \widetilde{\tau}_0 & \widetilde{\tau}_1 & \dots & \widetilde{\tau}_{n-1} & \widetilde{\tau}_n \\ \widetilde{\tau}_1 & \widetilde{\tau}_2 & \dots & \widetilde{\tau}_n & \widetilde{\tau}_{n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ \widetilde{\tau}_{n-1} & \widetilde{\tau}_n & \dots & \widetilde{\tau}_{2n-2} & \widetilde{\tau}_{2n-1} \\ 0 & 0 & \dots & 0 & 1 \end{vmatrix} = H_n(\{\widetilde{\tau}\})\,.$$

Аналогичным образом получаем

$$\sum_{j=1}^N \frac{\mathcal{H}_m(x_j;\{\tau\})y_j x_j^m}{W'(x_j)} = H_{m+1}(\{\tau\})\,.$$

Следовательно,

$$AH_n(\{\widetilde{\tau}\}) = BH_{m+1}(\{\tau\})\,.$$

Поскольку $A \neq 0$ и $H_n(\{\widetilde{\tau}\}) \neq 0$, то $H_{m+1}(\{\tau\}) \neq 0$, и последнее равенство завершает доказательство представимости рационального интерполянта по формулам (6.5) и (6.6).

Доказательство того факта, что эти полиномы действительно обеспечивают выполнение равенств (2.10), более громоздко и основано на равенствах

$$H_{m+1}(\{\tau\}) = (-1)^{N(N-1)/2} H_n(\{\widetilde{\tau}\}) \prod_{j=1}^N y_j,$$

$$\left\{ \mathcal{H}_m(x_k;\{\tau\}) = (-1)^{N(N-1)/2}\mathcal{H}_n(x_k;\{\widetilde{\tau}\}) \prod_{\substack{j=1\\ j\neq k}}^{N} y_j \right\}_{k=1}^{N} .$$

□

Замечание 5. *Формулировка теоремы 3 принадлежит авторам статьи. Якоби в [18] предложил разыскивать знаменатель рациональной интерполянты (потенциального кандидата) в форме ганкелева полинома* $\mathcal{H}_m(x;\{\tau\})$ *и после его вычисления свести задачу к случаю интерполяции полиномиальной. Он не рассматривал вопросов существования и единственности решения, а также не интересовался вычислительными аспектами практической реализации предложенного им подхода — включая процедуру, используемую при решении следующего примера и основанную на его же собственном результате, изложенном в п. 2.*

Пример 4. *Найти все рациональные интерполянты* $r(x) = p(x)/q(x), \deg p(x) + \deg q(x) \le 6$ *для таблицы*

x	-2	-1	0	1	2	3	4
y	$26/51$	2	$-1/2$	$1/6$	$-4/7$	$16/31$	$7/36$

.

Решение. Поскольку максимально возможная степень числителей и знаменателей искомых интерполянтов равна 6, вычислим последовательности (6.2) для значений индексов $\{0, 1, \ldots, 12\}$:

$$\tau_0 = -\frac{897683}{19123776}, \ \ldots, \tau_{12} = \frac{5257205447}{2390472}, \qquad \widetilde{\tau}_0 = -\frac{2973}{11648}, \ \ldots, \widetilde{\tau}_{12} = \frac{1294306589}{11648} .$$

Определим ганкелевы полиномы первого и второго порядков:

$$\mathcal{H}_1(x;\{\tau\}) = -\frac{897683}{19123776}x + \frac{119579}{4780944},$$

$$\mathcal{H}_2(x;\{\tau\}) = \underbrace{\frac{208609}{50996736}}_{h_{2,0}} x^2 \underbrace{-\frac{321193}{50996736}}_{h_{2,1}} x \underbrace{-\frac{7649}{1416576}}_{h_{2,2}} .$$

Расчет $\mathcal{H}_3(x;\{\tau\})$ произведем с применением тождества (5.4):

$$\mathcal{H}_3(x;\{\tau\}) \equiv -\left(\frac{h_{3,0}}{h_{2,0}}\right)^2 \mathcal{H}_1(x;\{\tau\}) + \frac{h_{3,0}}{h_{2,0}}\left(x - \frac{h_{2,1}}{h_{2,0}} + \frac{h_{3,1}}{h_{3,0}}\right)\mathcal{H}_2(x;\{\tau\}),$$

где все константы, кроме $h_{3,0} = H_3(\{\tau\})$ и $h_{3,1}$, уже известны. Для нахождения последних воспользуемся равенствами (5.10)

$$h_{3,0} = H_3(\{\tau\}) = \tau_4 h_{2,0} + \tau_3 h_{2,1} + \tau_2 h_{2,2} = -4037/16998912\,,$$

$$h_{3,1} = -(\tau_5 h_{2,0} + \tau_4 h_{2,1} + \tau_3 h_{2,2}) = 36263/50996736\,.$$

Поэтому

$$\mathcal{H}_3(x;\{\tau\}) \equiv -\frac{4037}{16998912}x^3 + \frac{36263}{50996736}x^2 - \frac{767}{12749184}x - \frac{41}{75888}\,.$$

Продолжая рекурсивное использование тождества (5.4), получим

$$\mathcal{H}_4(x;\{\tau\}) \equiv -\frac{1915}{50996736}x^4 + \frac{1915}{25498368}x^3 + \frac{9575}{152990208}x + \frac{1915}{38247552}\,,$$

$$\mathcal{H}_5(x;\{\tau\}) \equiv -\frac{1915}{21855744}x^5 + \frac{36385}{76495104}x^4 + \frac{6229}{8999424}x^3 - \frac{40711}{10927872}x^2 - \frac{359}{6374592}x + \frac{3037}{1195236}\,,$$

$$\mathcal{H}_6(x;\{\tau\}) \equiv \underbrace{\frac{991}{796824}}_{h_{6,0}}x^6 \underbrace{-\frac{8887}{1195236}}_{h_{6,1}}x^5 + \frac{3475}{2390472}x^4 + \frac{51575}{1195236}x^3 - \frac{8450}{298809}x^2 - \frac{4892}{99603}x + \underbrace{\frac{416}{42687}}_{h_{6,6}}\,.$$

Необходимо выполнить еще одну итерацию, а именно вычислить

$$H_7(\{\tau\}) = \tau_{12} h_{6,0} + \tau_{11} h_{6,1} + \cdots + \tau_6 h_{6,6} = -208/42687\,.$$

Таким образом, все возможные знаменатели интерполяционной дроби получены. Аналогичная рекурсивная процедура может быть организована для опре-

деления числителей:

$$\mathcal{H}_1(x;\{\widetilde{\tau}\}) = -\frac{2973}{11648}x+\frac{3037}{11648}, \quad \mathcal{H}_2(x;\{\widetilde{\tau}\}) = \underbrace{\frac{1915}{106496}}_{\widetilde{h}_{2,0}=H_2(\{\widetilde{\tau}\})}x^2 \underbrace{-\frac{21065}{745472}}_{\widetilde{h}_{2,1}} x+\underbrace{\frac{1915}{372736}}_{\widetilde{h}_{2,2}},$$

$$\widetilde{h}_{3,0} = \widetilde{\tau}_4\widetilde{h}_{2,0} + \widetilde{\tau}_3\widetilde{h}_{2,1} + \widetilde{\tau}_2\widetilde{h}_{2,2} = 5745/745472\,,$$

$$\widetilde{h}_{3,1} = -(\widetilde{\tau}_5\widetilde{h}_{2,0} + \widetilde{\tau}_4\widetilde{h}_{2,1} + \widetilde{\tau}_3\widetilde{h}_{2,2}) = -28725/745472\,,$$

$$\mathcal{H}_3(x;\{\widetilde{\tau}\}) \equiv -\left(\frac{\widetilde{h}_{3,0}}{\widetilde{h}_{2,0}}\right)^2 \mathcal{H}_1(x;\{\widetilde{\tau}\}) + \frac{\widetilde{h}_{3,0}}{\widetilde{h}_{2,0}}\left(x - \frac{\widetilde{h}_{2,1}}{\widetilde{h}_{2,0}} + \frac{\widetilde{h}_{3,1}}{\widetilde{h}_{3,0}}\right)\mathcal{H}_2(x;\{\widetilde{\tau}\}) \equiv$$

$$\equiv \frac{5745}{745472}x^3 - \frac{28725}{745472}x^2 + \frac{33771}{372736}x - \frac{369}{6656}\,,$$

$$\mathcal{H}_4(x;\{\widetilde{\tau}\}) \equiv \frac{36333}{745472}x^4 - \frac{72771}{372736}x^3 - \frac{11139}{28672}x^2 + \frac{1005843}{745472}x - \frac{206523}{372736}\,,$$

$$\mathcal{H}_5(x;\{\widetilde{\tau}\}) \equiv -\frac{625827}{745472}x^5+\frac{1708605}{372736}x^4-\frac{362367}{745472}x^3-\frac{2007361}{93184}x^2+\frac{3068941}{186368}x+\frac{119579}{46592}\,,$$

$$\mathcal{H}_6(x;\{\widetilde{\tau}\}) \equiv$$

$$\equiv \frac{897683}{93184}x^6-\frac{5805465}{93184}x^5+\frac{373613}{7168}x^4+\frac{24907053}{93184}x^3-\frac{9008491}{23296}x^2-\frac{392865}{23296}x+\frac{42687}{416}\,.$$

Наконец, составим рациональные интерполянты комбинациями найденных числителей и знаменателей:

$$r_{0,6}(x) = \frac{H_7(\{\tau\})}{\mathcal{H}_6(x;\{\tau\})} \equiv$$

$$\equiv -\frac{11648}{2973\,x^6 - 17774\,x^5 + 3475\,x^4 + 103150\,x^3 - 67600\,x^2 - 117408\,x + 23296},$$

$$r_{1,5}(x) = \frac{h_{6,0}\mathcal{H}_1(x;\{\widetilde{\tau}\})}{\widetilde{h}_{1,0}\mathcal{H}_5(x;\{\tau\})} \equiv$$

$$\equiv -\frac{64(2973\,x - 3037)}{13405\,x^5 - 72770\,x^4 - 105893\,x^3 + 569954\,x^2 + 8616\,x - 388736}\,,$$

$$r_{2,4}(x) = \frac{h_{5,0}\mathcal{H}_2(x;\{\widetilde{\tau}\})}{\widetilde{h}_{2,0}\mathcal{H}_4(x;\{\tau\})} \equiv \frac{7\,x^2 - 11\,x + 2}{3\,x^4 - 6\,x^3 - 5\,x - 4}\;;\cdots;$$

$$r_{6,0}(x) = \frac{h_{1,0}\mathcal{H}_6(x;\{\widetilde{\tau}\})}{\widetilde{h}_{6,0}} \equiv$$

$$\equiv -\frac{897683}{19123776}x^6+\frac{1935155}{6374592}x^5-\frac{4856969}{19123776}x^4-\frac{8302351}{6374592}x^3+\frac{9008491}{4780944}x^2+\frac{130955}{1593648}x-\frac{1}{2}\,.$$

□

Интерполяция по таблице с ошибками

Задача 2. Пусть таблица (2.8) содержит не более E "ошибочных" значений, т.е. существует полином $p(x)$ степени $n < N-1$ такой, что

$$p(x_j) = y_j \quad \text{для} \quad j \in \{1,\dots N\} \setminus \{e_1,\dots e_E\} \tag{6.8}$$

для некоторых различных $e_1,\dots e_E$ из $\{1,\dots N\}$. Точное количество ошибочных значений и их положение априори не известны. Требуется найти места ошибок и полином $p(x)$.

Существование и единственность решения поставленной задачи зависит от соотношения трёх параметров, а именно N, n и E. Заметим, что должно выполняться $N-E > n+1$, т.е. безошибочных значений должно быть долтаточно для идентификации полинома.

Пример 5. *Построить последовательность полиномов* $\{\mathcal{H}_k(x;\{\widetilde{\tau}\})\}_{k=1}^6$ *для таблицы:*

x	-2	-1	0	1	2	3	4
y	30	15	8	9	18	35	60

Решение. Имеем:

$$\mathcal{H}_1(x;\{\widetilde{\tau}\}) \equiv -\frac{89}{1814400}x+\frac{211}{226800},\ \mathcal{H}_2(x;\{\widetilde{\tau}\}) \equiv -\frac{2}{9568125}(4\,x^2-3\,x+8),$$

$$\mathcal{H}_3(x;\{\widetilde{\tau}\}) \equiv 0,\ \mathcal{H}_4(x,\{\widetilde{\tau}\}) \equiv 0,\ \mathcal{H}_5(x;\{\widetilde{\tau}\}) \equiv 0\,,$$

$$\mathcal{H}_6(x;\{\widetilde{\tau}\}) \equiv -\frac{1}{306180000}(4\,x^2-3\,x+8)\,.$$

Оказывается, что данная таблица сгенерирована полиномом $p(x) = 4\,x^2-3\,x+8$ второй степени и потому таблица избыточна. □

Следует уделить внимание тому факту, что в предыдущем примере выражение для интерполяционного полинома появилось не только на последнем шаге, но и на некоторых промежуточных в процессе построения последовательности ганкелевых полиномов.

Теорема 4. *Пусть интерполяционная таблица (2.8) сгенерирована полиномом*

$$p(x) = p_0 x^n + \cdots + p_n\,,\ p_0 \neq 0$$

степени $n < N-1$; пусть $y_j \neq 0$ для $j \in \{1,\ldots,N\}$. Тогда имеем:

$$\mathcal{H}_n(x;\{\widetilde{\tau}\}) \equiv \frac{(-1)^{Nn+n(n+1)/2} p_0^{N-n-1}}{\prod_{j=1}^{N} y_j} p(x), \quad \mathcal{H}_{N-1}(x;\{\widetilde{\tau}\}) \equiv \frac{(-1)^{N(N-1)/2}}{\prod_{j=1}^{N} y_j} p(x)\,. \tag{6.9}$$

Если $n < N-2$, то

$$\mathcal{H}_{n+1}(x;\{\widetilde{\tau}\}) \equiv 0, \ldots, \mathcal{H}_{N-2}(x;\{\widetilde{\tau}\}) \equiv 0\,. \tag{6.10}$$

Доказательство. Мы докажем теорему при дополнительном предположении, что $p(x)$ содержит только простые корни. Обозначим их $\lambda_1,\ldots,\lambda_n$. Построим новую последовательность:

$$\eta_k = \sum_{\ell=1}^{n} \frac{\lambda_\ell^k}{p'(\lambda_\ell) W(\lambda_\ell)} \quad \text{for} \quad k = 0,1,\ldots \tag{6.11}$$

Далее получим:

$$\widetilde{\tau}_k = \sum_{j=1}^{N} \frac{x_j^k}{y_j W'(x_j)} = \sum_{j=1}^{N} \frac{x_j^k}{p(x_j) W'(x_j)} = -\sum_{\ell=1}^{n} \frac{\lambda_\ell^k}{p'(\lambda_\ell) W(\lambda_\ell)} = -\eta_k \ \text{ для } k \in \{0,1,\ldots,N-n-2\} \tag{6.12}$$

в то время как

$$\widetilde{\tau}_{N+n-1} = \frac{1}{p_0} - \eta_{N+n-1}\,. \tag{6.13}$$

Используя эти отношения, представим n-й ганкелев полином в виде

$$\mathcal{H}_n(x;\{\widetilde{\tau}\}) \equiv (-1)^n \begin{vmatrix} \eta_0 & \eta_1 & \eta_2 & \ldots & \eta_{n-1} & \eta_n \\ \eta_1 & \eta_2 & \eta_3 & \ldots & \eta_n & \eta_{n+1} \\ \vdots & & & \ddots & & \vdots \\ \eta_{n-1} & \eta_n & \eta_{n+1} & \ldots & \eta_{2n-2} & \eta_{2n-1} \\ 1 & x & x^2 & \ldots & x^{n-1} & x^n \end{vmatrix} \equiv (-1)^n \mathcal{H}_n(x;\{\eta\})\,.$$

Аналогичным образом, получим

$$\mathcal{H}_n(\lambda_j;\{\eta\}) = 0 \quad \text{for} \;\; j \in \{1,\dots,n\}\,. \tag{6.14}$$

Старший коэффициент $\mathcal{H}_n(x;\{\eta\})$, т.е.

$$H_n(\{\eta\}) = \begin{vmatrix} \eta_0 & \eta_1 & \eta_2 & \dots & \eta_{n-1} \\ \eta_1 & \eta_2 & \eta_3 & \dots & \eta_n \\ \vdots & & & \ddots & \vdots \\ \eta_{n-1} & \eta_n & \eta_{n+1} & \dots & \eta_{2n-2} \end{vmatrix},$$

может быть представлен в виде произведения

$$= \begin{vmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \vdots & & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \dots & \lambda_n^{n-1} \end{vmatrix} \cdot \begin{vmatrix} \frac{1}{p'(\lambda_1)W(\lambda_1)} & 0 & \dots & 0 \\ 0 & \frac{1}{p'(\lambda_2)W(\lambda_2)} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{p'(\lambda_n)W(\lambda_n)} \end{vmatrix} \times$$

$$\times \begin{vmatrix} 1 & \lambda_1 & \dots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{n-1} \\ \vdots & & & \vdots \\ 1 & \lambda_n^{n-1} & \dots & \lambda_n^{n-1} \end{vmatrix}$$

$$= \frac{\prod\limits_{1\le k<j\le n}(\lambda_j-\lambda_k)^2}{\prod\limits_{\ell=1}^{n} p'(\lambda_\ell)\prod\limits_{\ell=1}^{n} W(\lambda_\ell)}\,.$$

Поскольку

$$\prod_{1\le k<j\le n}(\lambda_j-\lambda_k)^2 = \frac{(-1)^{n(n-1)/2}}{p_0^n}\prod_{\ell=1}^{n} p'(\lambda_\ell) \tag{6.15}$$

и

$$\prod_{\ell=1}^{n} W(\lambda_\ell) = \prod_{\ell=1}^{n}\prod_{j=1}^{N}(\lambda_\ell - x_j) = (-1)^{Nn}\prod_{j=1}^{N}\prod_{\ell=1}^{n}(x_j-\lambda_\ell) = (-1)^{Nn}\frac{\prod_{j=1}^{N} p(x_j)}{p_0^N}\,, \tag{6.16}$$

справедливость первого из равенств (6.9) установлена.

Далее рассмотрим ганкелев полином $\mathcal{H}_K(x;\{\widetilde{\tau}\})$ степени $K \in \{n+1,\ldots,\lfloor(N+n-1)/2\rfloor\}$. Воспользовавшись (6.12), данный полином можно представить как

$$\mathcal{H}_K(x;\{\widetilde{\tau}\}) \equiv (-1)^K \begin{vmatrix} \eta_0 & \eta_1 & \eta_2 & \ldots & \eta_K \\ \eta_1 & \eta_2 & \eta_3 & \ldots & \eta_{K+1} \\ \vdots & & & \ddots & \vdots \\ \eta_{K-1} & \eta_K & \eta_{K+1} & \ldots & \eta_{2K-1} \\ 1 & x & x^2 & \ldots & x^K \end{vmatrix} \equiv (-1)^K \mathcal{H}_K(x,\{\eta\})\,.$$

Коэффициенты $\mathcal{H}_K(x;\{\widetilde{\tau}\})$ совпадают (с точностью до знака) с минорами K-го порядка ганкелевой матрицы, порождённой последовательностью(6.11). Поскольку все эти миноры равны нулю, то

$$\mathcal{H}_{n+1}(x;\{\widetilde{\tau}\}) \equiv 0, \ldots, \mathcal{H}_{\lfloor(n+N-1)/2\rfloor}(x;\{\widetilde{\tau}\}) \equiv 0\,.$$

Далее рассмотрим случай, когда ганкелев полином имеет степень $K \in \{\lfloor(n+N)/2\rfloor,\ldots,N-2\}$.

$$\mathcal{H}_K(x;\{\widetilde{\tau}\}) = \begin{vmatrix} \widetilde{\tau}_0 & \widetilde{\tau}_1 & \ldots & \ldots & \ldots & \widetilde{\tau}_K \\ \widetilde{\tau}_1 & \widetilde{\tau}_2 & \ldots & \ldots & \ldots & \widetilde{\tau}_{K+1} \\ \vdots & & & & & \vdots \\ \widetilde{\tau}_{N+n-K-2} & \ldots & & & \ldots & \widetilde{\tau}_{N+n-2} \\ \widetilde{\tau}_{N+n-K-1} & & & & \widetilde{\tau}_{N+n-2} & \widetilde{\tau}_{N+n-1} \\ \vdots & & & & \vdots & \\ \widetilde{\tau}_{K-1} & \ldots & \widetilde{\tau}_{N+n-2} & \widetilde{\tau}_{N+n-1} & \ldots & \widetilde{\tau}_{2K-1} \\ 1 & x & \ldots & \ldots & & x^K \end{vmatrix}.$$

Из (6.12) элементы первых $N+n-K-1$ строк этого определителя могут быть

представлены в виде

$$\equiv (-1)^{N+n-K-1}\begin{vmatrix} \eta_0 & \eta_1 & \dots & \dots & \dots & \eta_K \\ \eta_1 & \eta_2 & \dots & \dots & \dots & \eta_{K+1} \\ \vdots & & & & & \vdots \\ \eta_{N+n-K-2} & \dots & & & \dots & \eta_{N+n-2} \\ \widetilde{\tau}_{N+n-K-1} & & & & \widetilde{\tau}_{N+n-2} & \widetilde{\tau}_{N+n-1} \\ \vdots & & & & \vdots & \\ \widetilde{\tau}_{K-1} & \dots & \widetilde{\tau}_{N+n-2} & \widetilde{\tau}_{N+n-1} & \dots & \widetilde{\tau}_{2K-1} \\ 1 & x & \dots & \dots & x^{K-1} & x^K \end{vmatrix}.$$

Эти $N+n-K-1$ строк линейно зависимы, поскольку все миноры $(N+n-K-1)$-го порядка матрицы, составленной из этих строк, равны нулю. Таким образом, $\mathcal{H}_K(x;\{\widetilde{\tau}\}) \equiv 0$ для значений K, упомянутых в начале текущего параграфа.

Пример 6. *Построить последовательность полиномов* $\{\mathcal{H}_k(x;\{\widetilde{\tau}\})\}_{k=1}^{6}$ *для таблицы*

x	-2	$\mathbf{-1}$	0	1	2	3	4
y	30	$\mathbf{12}$	8	9	18	35	60

которая отличается от таблицы из Примера 5 единственным значением в узле $x_2 = -1$.

Решение. Отбросив данное значение из рассмотрения, получим таблицу, которая всё ещё содержит достаточно данных для восстановления полинома $p(x) = 4\,x^2 - 3\,x + 8$ второй степени. Будем рассматривать $y_2 = 12$ как *ошибку* в интерполяционной таблице. Проанализируем её влияние на построение интерполяционного полинома путём рекурсивной процедуры вычисления ганкелевых полиномов. Имеем:

$$\mathcal{H}_1(x;\{\widetilde{\tau}\}) \equiv -\frac{341}{1814400}x + \frac{359}{453600},$$

$$\mathcal{H}_2(x;\{\widetilde{\tau}\}) \equiv \frac{1}{39191040000}(-19109\,x^2 + 34323\,x - 69448),$$

$$\mathcal{H}_3(x;\{\widetilde{\tau}\}) \equiv \frac{1}{2449440000}(x+1)(4\,x^2 - 3\,x + 8),$$

$$\mathcal{H}_4(x;\{\widetilde{\tau}\}) \equiv 0,$$

$$\mathcal{H}_5(x;\{\widetilde{\tau}\}) \equiv -\frac{1}{39191040000}(x+1)(4\,x^2-3\,x+8)\,,$$

$$\mathcal{H}_6(x;\{\widetilde{\tau}\}) \equiv -\frac{1}{979776000}\left(\frac{1}{40}x^6-\frac{1}{5}x^5+\frac{3}{8}x^4+\frac{1}{2}x^3+\frac{21}{10}x^2-\frac{9}{5}x+8\right)$$

с интерполяционным полиномом

$$\widetilde{p}(x) \equiv \frac{1}{40}x^6-\frac{1}{5}x^5+\frac{3}{8}x^4+\frac{1}{2}x^3+\frac{21}{10}x^2-\frac{9}{5}x+8$$

который может рассматриваться в качестве возмущения полинома $p(x)=4\,x^2-3\,x+8$:

$$\equiv p(x)+(y_2-p(x_2))\frac{W_2(x)}{W'(x_2)}\,.$$

□

Глава 7. Заключение

В настоящей статье развит основанный на идеях К. Якоби подход к решению задачи рациональной интерполяции, заключающийся в представлении решения посредством подходящих ганкелевых полиномов. Предложена процедура эффективного вычисления этих полиномов, позволяющая построить не только одиночную интерполянту, но и целое семейство интерполянт при всех возможных комбинациях степеней числителя и знаменателя. Этот аспект обеспечивает возможность выбора решения, не только формально удовлетворяющего интерполяционной таблице, но и обладающего дополнительными свойствами, существенными для задач аппроксимации (как, к примеру, ограниченность на определенном интервале вещественной оси).

Следует отметить, что появление матриц ганкелевой структуры в решениях задач помехоустойчивого кодирования и аппроксимации не должно считаться абсолютно неожиданным: достаточно вспомнить вид системы нормальных уравнений при построении полиномиальной аппроксимации интерполяционной таблицы по методу наименьших квадратов [9]. В книгах [13, 17] можно найти и другие задачи, в которых такие матрицы возникают естественным образом — например, задачу интерполяции таблицы (2.8) комбинацией экспонент: $f(x) = \sum_{j=1}^{m} a_j e^{k_j x}$.

Направления дальнейших исследований

Развитие предложенного подхода видится в направлении интерполяции многомерной. Кроме того, предполагается произвести сравнение вычислительной эффективности алгоритма с альтернативными подходами, в частности с представлением рационального интерполянта в барицентрической форме [4].

Литература

[1] Baravy I. Efficient Interpolation over Infinite and Finite Fields via Hankel Polynomials // Mathematical Modelling and Analysis, Abstracts. 2014. P. 9.

[2] Berlekamp E.R. Algebraic Coding Theory. Aegean Park Press, 1984.

[3] Berrou C. Error-correction coding method with at least two systematic convolutional codings in parallel. US Patent 5,446,747, 1995.

[4] Berrut J.-P., Baltensperger R., Mittelmann H.D. Recent developments in barycentric rational interpolation. International Series of Numerical Mathematics. 2005. Basel, Birkhäuser, Vol. 151, P. 27-51.

[5] Blahut R. Theory and Practice of Error Control Codes. Addison-Wesley, 1983.

[6] Blahut R. Fast Algorithms for Digital Signal Processing. Addison-Wesley, 1985.

[7] Bose R.C., Ray-Chaudhuri D.K. On A Class of Error Correcting Binary Group Codes // Information and Control, 1960. Vol. 3, No. 1. P. 68–79.

[8] Cauchy A.-L. Cours d'Analyse de l'École Royale Polytechnique: Part I: Analyse Algébrique. Paris, France: L'Imprimerie Royale, 1821, pt. 1. Annotated English Translation:

[9] Faddeev D. K. Lectures in Algebra. Moscow. Nauka Publ., 1984. 416 p.

[10] Hagenauer J., Offer E., Papke L. Iterative Decoding of Binary Block and Convolutional Codes // IEEE Transactions on Information Theory, 1996. Vol. 42, No. 2. P. 429-445.

[11] Hamming R.W. Error-Detecting and Error-Correcting Codes // Bell Systems Technical Journal, 1950. Vol. 29. P. 147-160.

[12] Hamming R.W. Numerical Methods for Scientists and Engineers. Dover Publications, 1987.

[13] Henrici P. Applied and computational complex analysis, Vol. 1. Power series, integration, conformal mapping, location of zeros. New York, London, Wiley and Sons Inc. Publ., 1974, 700 p.

[14] Hocquenghem A. Codes correcteurs d'erreurs. 1959.

[15] ISO/IEC 18004:2006. Information technology. Automatic identification and data capture techniques. QR Code. Geneva: ISO/IEC, 2000.

[16] ISO 2108:2005. Information and documentation – International standard book number (ISBN). ISO, 2013.

[17] Iohvidov I. S. Hankel and Toeplitz matrices and forms: algebraic theory. Boston, Birkhäuser, 1972, 260 p.

[18] Jacobi C.G.J. Über die Darstellung einer Reihe gegebner Werthe durch eine gebrochne rationale Function // J. reine angew. math, 1846. Vol. 30. P. 127-156.

[19] Joachimsthal F. Bemerkungen über den Sturm'schen Satz // J. reine angew. math, 1854. Vol. 48. P. 386-416.

[20] Kramer H.K., Lane R.N. Decomposition of a function into a weighted sum of shifted replicas of another function // Journal of Mathematical Analysis and Applications, 1974. Vol. 46, No 3. P. 395–608.

[21] Kronecker L. Über einreihige Determinanten. Nachrichten den Königlichen Gesellschaft der Wissenschaften zu Göttingen, 1881. Vol. 9. P. 271-279.

[22] Kronecker L. Zur Theorie der Elimination einer Variabeln aus zwei algebraischen Gleichungen. Monatsberichte der Königlichen Preussische Akademie des Wissenschaften zu Berlin, 1881, Juni, P. 535-600.

[23] Micron Technology. Error Correction Code (ECC) in Micron® Single-Level Cell (SLC) NAND. Technical Note TN-29-63, 2011.

[24] Netto E. Zur Cauchy'schen Interpolationsaufgabe // Math. Ann. 1893. Vol. 42 (3). P. 453-456.

[25] Reed I.S., Solomon G. Polynomial Codes over Certain Finite Fields // Journal of the Society for Industrial and Applied Mathematics (SIAM), 1960. Vol. 8, No. 2. P. 300–304.

[26] Rong Q.J. Linear independence of translates of a box spline // Journal of Approximatin Theory, 1984. Vol. 40, No 2. P. 158–160.

[27] Rudra A. Error Correcting Codes: Combinatorics, Algorithms and Applications. http://www.cse.buffalo.edu/~atri/courses/

[28] Shannon C.E. A Mathematical Theory of Communication // Bell System Technical Journal, 1948. Vol. 27, No 3. P. 379–423.

[29] STMicroelectronics. Error Correction Code in NAND Flash Memories. Application Note AN1823, 2004.

[30] Sudan M. Decoding of Reed Solomon codes beyond the error-correction bound // J. Complexity, 1997. Vol. 13. P. 180–193.

[31] Uteshev A.Yu., Baravy I. Inversion in finite fields with the aid of Hankel polynomials // International Conference on Computer Science and Information Technologies, 2013. P. 1-6.

[32] Wang Y., Zhu X. A fast algorithm for the Fourier transform over finite fields and its VLSI implementation // IEEE Journal on Selected Areas in Communications, 1988. Vol. 6, No. 3. P. 572–577.

[33] Утешев А.Ю., Боровой И.И. Решение задачи рациональной интерполяции с использованием ганкелевых полиномов // Вестник Санкт-Петербургского университета. Сер. 10. Прикладная математика. Информатика. Процессы управления, 2016. Вып. 4. С. 31–43.

[34] Записная книжка профессора Утешева. http://pmpu.ru