НАЦИОНАЛЬНЫЙ АВИАЦИОННЫЙ УНИВЕРСИТЕТ.
ИНСТИТУТ АЭРОКОСМИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ
КАФЕДРА ЭЛЕКТРОТЕХНИКИ И СВЕТОТЕХНИКИ



Воронов С.И., Воронова О.С.

Аппроксимация таблично заданных функций в инженерных задачах

(Методы и алгоритмы).

УЧЕБНОЕ ПОСОБИЕ
(Редакция от 2011г.)

ОГЛАВЛЕНИЕ

Ведение ... 3
Постановка задачи ... 3
Задача аппроксимации в дискретной форме ... 4
Построение условий для решения задачи аппроксимации в дискретной форме. ... 4
Шаг 1. Дискретные отклонения и их сумма ... 5
Шаг 2. Отклонения в форме модуля ... 6
Шаг 3. Отклонения в квадратичной форме ... 6
Шаг 4. Формальное условие для решения задачи аппроксимации ... 7
Решение задачи аппроксимации в дискретной форме ... 7
Шаг 5. Поиск минимума оценки D. Постановка задачи. ... 7
Шаг 6. Вспомогательные преобразования оценки D ... 8
Шаг 7. Дифференцирование оценки D ... 9
Шаг 8. Продолжаем упрощения частной производной функции D ... 10
Шаг 9. Построение компактной формы частных производных ... 10
Шаг 10. Построение системы линейных уравнений ... 11
Решение задачи аппроксимации в ее дискретной форме. ... 12
Интегральная форма задачи аппроксимации ... 13
Переход к интегральной форме ... 13
Решение задачи аппроксимации в интегральной форме ... 14
Выбор базисных функций ... 15
Задача аппроксимации в ортогональных базисах ... 16
Классическое решение задачи аппроксимации в ортогональном базисе ... 17
Примеры классических ортогональных функций ... 18
Многочлены Чебышева ... 18
Функции, образующие ряды Фурье ... 19
Перенормировка интервала ортогональности ... 21
Оценка качества решения задачи аппроксимации ... 23
Задача аппроксимации, как задача теории вероятностей ... 23
Изложим кратко основные моменты теории случайных процессов ... 23
Привязка положений теории случайных процессов к задаче аппроксимации ... 25
Понижение влияния шумов на результаты аппроксимации ... 26
Эвристические приемы повышения качества аппроксимации ... 29
Укрупненные алгоритмы решения задачи аппроксимации ... 31
Дискретное решение в неортогональном базисе. ... 31
Алгоритм для вычисления коэффициентов системы линейных уравнений. ... 32
Решение в интегральном виде для неортогонального базиса. ... 34
Решение в ортогональном базисе. ... 36
Описание программы для аппроксимации табличных функций степенными рядами и рядами Фурье ... 37
Список литературы, ссылки ... 40

Ведение

Данное пособие предназначено для студентов электротехнических и светотехнических специальностей, как вспомогательный материал по дисциплинам «Вычислительная техника и алгоритмические языки»; «Аппаратно – программное обеспечение технических систем»; «Основы построения систем в светотехнике»; «Компьютерная графика и моделирование». Кроме того, пособие может использоваться в целом ряде других дисциплин, в которых рассматриваются вопросы нахождения аналитического вида характеристик разнообразных объектов (например, различных датчиков, передаточных функций и т.п.) или рассматриваются вопросы построения дискретных спектров сигналов, спектров графических изображений и т.п.

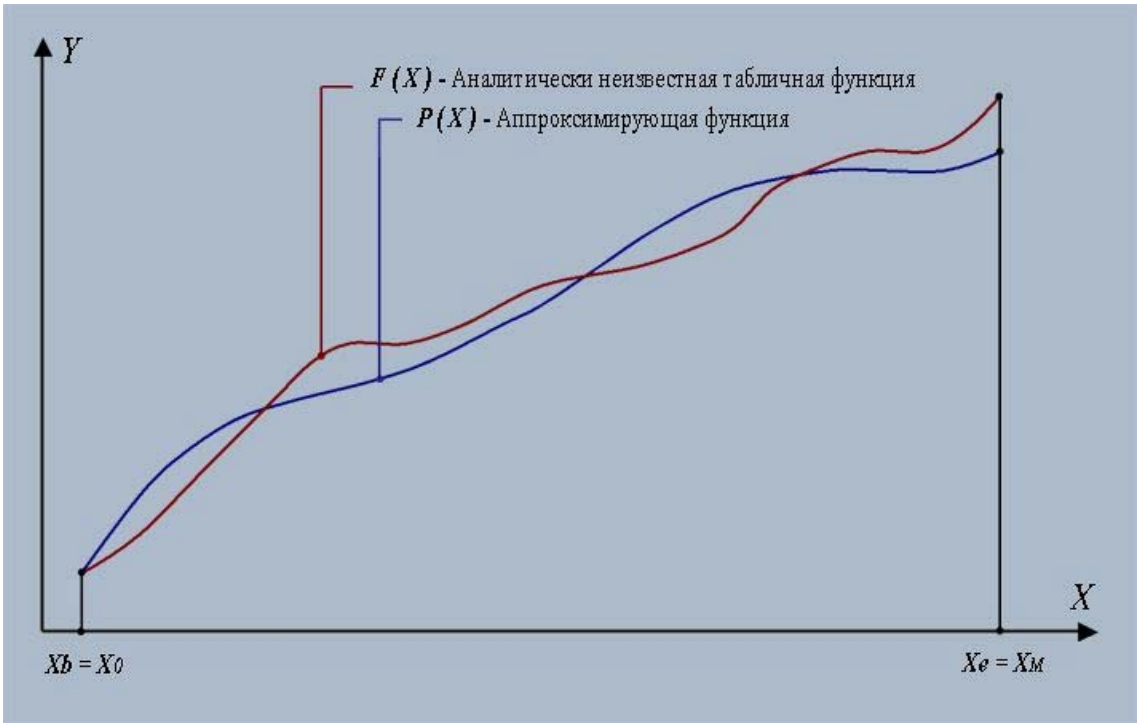
Поскольку, перечисленные дисциплины преподаются в разные годы обучения студентов, то материал пособия излагается с опорой только на начальные сведения по высшей математике, которые включают основы дифференциального и интегрального исчисления в объеме первого года обучения студентов.

Аппроксимация таблично заданной функции.

Постановка задачи

Пусть некоторую функцию ***F(x),*** представленную табличными значениями, на аргументах ***(X_0 , ... , X_m , ... , X_M)*** в области ***(X_b , X_e),*** необходимо наилучшим образом (то есть наиболее точным и, по возможности, наиболее компактным образом) представить в аналитическом виде.

Такая задача получила наименование – «задача аппроксимации таблично заданной функции».



Эта задача, как правило, решается с помощью ***аппроксимирующей*** функции ***P(x)*** путем определения ее коэффициентов ***(a_0 , ... , a_i , ... , a_n)*** таким образом, чтобы достигнуть минимального различия между исходной таблично заданной функцией ***F(x)*** и ее аналитическим эквивалентом, то есть:

$$F(x) \cong P(x) = \sum_{n=0}^{N} a_n v_n(x)$$

Где:

$v_n(x)$ - является одной из заранее выбранных аналитических функций, которые называют базисными функциями;

a_n - является искомым коэффициентом при этой функции.

Как Вы наверное уже догадались, для того, чтобы найти решение поставленной задачи, как минимум, необходимо математически определить ***условия,*** при которых достигается равенство:

$$F(x) \cong P(x)$$

Следует также подчеркнуть, что успех решения задачи аппроксимации зависит также от того, будут ли такие условия конструктивны с точки зрения достижения поставленной цели.

Задача аппроксимации в дискретной форме

Классическим примером условий, которые позволяют решить задачу аппроксимации, являются условия, сформулированные в методе наименьших квадратов. Построение таких условий мы рассмотрим ниже.

> ***Примечание для студентов пятого курса***. Формально такие условия записываются в виде функционала, который определяет некоторую оценку «расстояния между функциями», а также критерия, определяющего минимум этой оценки. Задача аппроксимации сводится к нахождению с помощью функционала, конкретной функции, удовлетворяющей критерию.

Построение условий для решения задачи аппроксимации в дискретной форме.

Для начала, исходя из предположения о существовании аналитического вида функций, графически представим функцию ***F(x)***, а также некоторую аппроксимирующую функцию ***P(x)***, которая пока еще не вполне удовлетворяет решению нашей задачи.

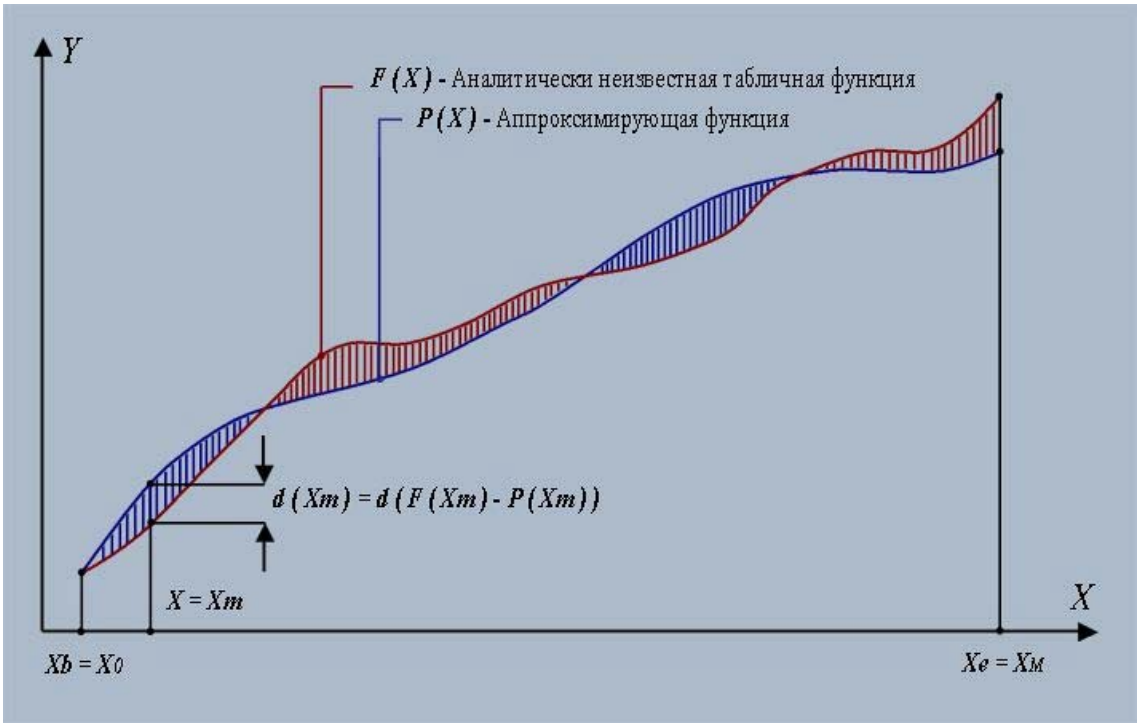


Рис.1. Отклонения между аппроксимируемой и аппроксимирующей функциями.

Шаг 1. Дискретные отклонения и их сумма

Выводы, которые непосредственно доступны из графического представления функций ***F(x)*** и ***P(x)*** можно сформулировать следующим образом:

- ***Вывод 1***. Если бы нам сразу удалось достигнуть равенства функций, то для каждого значения независимой переменной ***x***, отклонение ***d*** между соответствующими значениями этих функций было бы нулевым.

- ***Вывод 2***. Сумма всех отклонений будет тем ближе к нулю, чем ближе к нулю будет каждое из составляющих сумму отклонений. Однако, данное утверждение будет справедливым только в том случае, если все отклонения ***d*** мы будем рассматривать без учета их знака.

- ***Следствие*** – Если сумма всех отклонений (без учета их знака) стремится к нулю,

$$\sum_{m=0}^{M} d(x_m) \rightarrow 0$$

то к нулю будут стремиться все входящие в нее отклонения. Таким образом, если мы построим формальный метод, вычисляющий коэффициенты $(a_0, \ldots, a_i, \ldots, a_K)$, при этом значения этих коэффициентов позволит наилучшим образом приблизить такую сумму к нулевому значению, то это будет равнозначно решению задачи аппроксимации:

$$F(x) \cong P(x)$$

Шаг 2. Отклонения в форме модуля

Для начала, определим отклонение ***d(x)*** между функциями ***F(x)*** и ***P(x)*** как простую разность между этими функциями:

$$d(x) = F(x) - P(x)$$

К сожалению, простые разности, полученные при различных значениях ***x***, могут иметь как положительные, так и отрицательные значения. В итоге, суммируя все положительные и отрицательные отклонения между функциями ***F(x)*** и ***P(x),*** вполне возможно получить нулевой результат при очевидном неравенстве этих функций. Для этого достаточно, чтобы сумма всех положительных отклонений случайным образом оказалась равна сумме всех отрицательных отклонений.

На первый взгляд, этот недостаток легко устранить - достаточно разность между функциями записать в форме модуля, и проблема будет устранена:

$$d_{\mathrm{mod}}(x_m) = \left| F(x_m) - P(x_m) \right|$$

Однако применение модуля вносит в такое выражение особые точки, то есть точки, в которых производные, взятые слева и справа от точки не равны между собой. Появление таких особых точек (а их количество явно непредсказуемо) сразу вносит существенные сложности в методы поиска минимумов, которые нам далее придется использовать.

Шаг 3. Отклонения в квадратичной форме

Проблема особых точек, обнаруженная нами на шаге 2, ставит перед нами простой вопрос – что для нас более важно, удобная и надежная оценка равенства функций ***F(x)*** и ***P(x)*** или привязка конструируемой оценки к отклонению в виде *простой разницы* между этими функциями?

Ответ очевиден – *важно получить однозначную, всюду положительную, непрерывную функцию оценки, избавленную от особых точек*. Самый простой способ получить такую функцию оценку, это представить отклонение как квадрат разности функций ***F(x)*** и ***P(x)*** :

$$d_{sq}(x_m) = (F(x_m) - P(x_m))^2$$

В этом случае, сумма всех отклонений определенная как:

$$\sum_{m=0}^{M} d_{sq}(x_m) \to 0$$

будет располагать всем, необходимым для нас, набором свойств:

- Сумма будет гарантировано положительна и ограничена снизу нулевым значением, которое является индикатором равенства функций ***F(x)*** и ***P(x)***

- Операции, которые использовались при конструировании суммы, не будут вносить особых точек, препятствующих применению методов минимизации.

Шаг 4. Формальное условие для решения задачи аппроксимации

На основании результатов шага 3, запишем окончательное выражение для формальной оценки успешности решения задачи аппроксимации можно записать следующим образом:

$$D = \sum_{m=0}^{M} ((F(x_m) - P(x_m))^2 \rightarrow 0$$

Или:

$$D = \sum_{m=0}^{M} ((F(x_m) - \sum_{n=0}^{N} a_n v_n(x_m))^2 \rightarrow 0$$

> ***Примечание для студентов пятого курса***. Приведенное формальное условие, фактически является функционалом, который связывает множество значений ***D*** с множеством функций ***P(x)***. Кроме того, условие задает критерий решения задачи, путем поиска такой функции ***P(x)*** для которой значение ***D*** будет максимально близко к нулевому значению.

Решение задачи аппроксимации в дискретной форме

Шаг 5. Поиск минимума оценки D. Постановка задачи.

Известно, что основным способом нахождения минимумов и максимумов некоторой функции является исследование точек локальных экстремумов функций, то есть таких точек, в которых первая производная функции равна нулю.

Рассмотрим общие свойства и особенности оценки ***D***.

Оценка ***D***, если ее рассматривать как функцию, является функцией многих переменных:

$$D(F(x), (a_0, ..., a_N), (v_0(x), ..., v_N(x)))$$

Очевидно, что для нахождения экстремумов оценки ***D***, нам доступны следующие возможности:

- выбор конкретного вида базисных функций ***v(x)***,
- выбор размерности базиса, то есть, количества базисных функций;
- выбор коэффициентов ***a*** при базисных функциях.

Поскольку выбор вида и размерности базиса достаточно трудно формализовать, будем рассматривать вид и размерность базиса как некоторые параметры, а в качестве независимых переменных для поиска экстремумов выберем коэффициенты ***a*** при функциях базиса.
Как уже отмечалось, поиск экстремумов обычных функций выполняется с помощью анализа первой производной таких функций.

В нашем случае, для нахождения экстремумов функции ***D***, нам достаточно потребовать, чтобы все частные производные функции ***D***, взятые по коэффициентам ***a*** при функциях ***v(x)*** равнялись нулю.

$$\frac{\partial}{\partial a_0} D(F(x), (a_0,..., a_N), (v_0(x),..., v_N(x))) = 0$$

...

$$\frac{\partial}{\partial a_N} D(F(x), (a_0,..., a_N), (v_0(x),..., v_N(x))) = 0$$

Поскольку, сформулированные условия должны выполняться совместно, то фактически нам представлена *система уравнений*, решение которой относительно коэффициентов ***a***, будет соответствовать искомым точкам экстремумов.

Будет ли конструктивной сформулированная система условий (уравнений)? Другими словами, даст ли нам ее решение однозначный ответ или мы получим множество точек, среди которых нам еще предстоит найти точку наименьшего минимума среди всех минимумов и максимумов? Действительно, о функции ***D*** нам пока известно только то, что она всюду положительная и ограниченная снизу.

Ответ на поставленный вопрос можно получить, если обратить внимание на характер, сконструированной нами оценки:

$$D = \sum_{m=0}^{M} (d(x_m))^2$$

Приведенное выражение, позволяет заметить, что функция $d(x)^2$ имеет характер параболы, следовательно, функция ***D*** будет иметь единственную точку экстремума и, поскольку функция ***D*** ограничена только снизу, такая точка будет соответствовать единственному минимуму функции ***D***.

> ***Примечание***. В очевидности данного утверждения легко убедиться, если рассматривать ***d***, как некоторую независимую переменную, а функцию ***D(d)*** представить в интегральной форме. Если продифференцировать функцию ***D(d)*** по ее независимой переменной ***d***, то результат (с точностью до коэффициента) примет вид функции d^2, которая равняется нулю в одной единственной точке. Подробнее об интегральной форме представления функции ***D*** смотрите ниже.

Вывод: Построенная нами система уравнений, при нахождении ее корней (то есть, искомых коэффициентов ***a***), равнозначна решению задачи аппроксимации.

Шаг 6. Вспомогательные преобразования оценки D

Выполним некоторые вспомогательные преобразования выражений, которые получены нами на шаге 4.

- **Шаг 6.1.** Раскроем квадрат разности:

$$D = \sum_{m=0}^{M} ((F^2(x_m) - 2F(x_m)P(x_m) + P^2(x_m))$$

- **Шаг 6.2.** Преобразуем сумму к виду:

$$D = \sum_{m=0}^{M} F^2(x_m) - 2\sum_{m=0}^{M} F(x_m)P(x_m) + \sum_{m=0}^{M} P^2(x_m)$$

- **Шаг 6.3.** Для определения вида частных производных функции ***D***, воспользуемся правилом вычисления производной от произведения двух функций, известного также как правило Лейбница:

$$\frac{d}{dx}(f(x)\cdot g(x)) = g(x)\cdot\frac{d}{dx}f(x) + f(x)\cdot\frac{d}{dx}g(x)$$

Кроме того, для дальнейшего упрощения преобразований, покажем применение этого правила для некоторой функции, возведенной в квадрат:

$$\frac{d}{dx}(\varphi(x))^2 = \frac{d}{dx}(\varphi(x)\cdot\varphi(x)) = 2\cdot\varphi(x)\cdot\frac{d}{dx}(\varphi(x))$$

Шаг 7. Дифференцирование оценки D

Итак, запишем частную производную функции ***D*** (см. шаг 6.2), взятую по коэффициенту ***a*** с некоторым индексом ***k*** в следующем виде:

$$\frac{\partial}{\partial a_k}D = \frac{\partial}{\partial a_k}\left[\sum_{m=0}^{M} F^2(x_m) - 2\sum_{m=0}^{M} F(x_m)P(x_m) + \sum_{m=0}^{M} P^2(x_m)\right]$$

Обратим внимание, что функция ***F(x),*** заданная нам как табличная функция, а также квадрат этой функции, никоим образом не зависят от переменной, по которой определяется производная. В подобных случаях, в соответствии с правилами вычисления частных производных, такие функции можно рассматривать как константы. С учетом этого замечания, а также вспоминая, что производная от константы равна нулю, получим:

$$\frac{\partial}{\partial a_k}D = -2\sum_{m=0}^{M} F(x_m)\frac{\partial}{\partial a_k}P(x_m) + \sum_{m=0}^{M}\frac{\partial}{\partial a_k}P^2(x_m)$$

Применим правило Лейбница для дальнейшего определения вида частной производной. Как следствие получим:

$$\frac{\partial}{\partial a_k} D = -2\sum_{m=0}^{M} F(x_m)\frac{\partial}{\partial a_k} P(x_m) + 2\sum_{m=0}^{M} P(x_m)\frac{\partial}{\partial a_k} P(x_m)$$

Шаг 8. Продолжаем упрощения частной производной функции D

Для дальнейшего упрощения вида производной, вспомним, что функция ***P(x)*** была нами определена как:

$$P(x) = \sum_{n=0}^{N} a_n v_n(x)$$

Следовательно, ее частная производная, взятая по коэффициенту с некоторым индексом ***k,*** будет иметь вид:

$$\frac{\partial}{\partial a_k} P(x) = \frac{\partial}{\partial a_k}\left(\sum_{n=0}^{N} a_n \cdot v_n(x)\right)$$

Выполняя преобразования на шаге 7, мы уже акцентировали внимание, что функция ***F(x)*** не зависит от коэффициентов ***a,*** что позволило нам существенно упростить вид производной***.*** Очевидно, что функции ***v(x)*** также являются внутренне независимыми от своих коэффициентов. Следовательно. при вычислении производных взятых по коэффициентам ***a***, функции ***v(x)*** также могут рассматриваться как константы. В силу этого, можно записать:

$$\frac{\partial}{\partial a_k} P(x) = \sum_{n=0}^{N} v_n(x) \cdot \frac{\partial}{\partial a_k} a_n) = v_k(x)$$

Такое существенное упрощение возможно благодаря очевидному равенству:

$$\frac{\partial}{\partial a_k} a_n = \begin{cases} 0, & если \quad k \neq n \\ 1, & если \quad k = n \end{cases}$$

Шаг 9. Построение компактной формы частных производных

Подставим значение частной производной функции ***P(x)*** в выражение частной производной функции ***D***, которое было ранее получено на шаге 7:

$$\frac{\partial}{\partial a_k} D = -2\sum_{m=0}^{M} F(x_m)\cdot v_k(x_m) + 2\sum_{m=0}^{M} P(x_m)\cdot v_k(x_m)$$

В развернутом виде это можно записать следующим образом:

$$\frac{\partial}{\partial a_k} D = -2\sum_{m=0}^{M} F(x_m)\cdot v_k(x_m) + 2\sum_{m=0}^{M}(v_k(x_m)\cdot\sum_{n=0}^{N} a_n v_n(x_m))$$

Применяя правила перестановки операций суммирования и умножения, итоговый вид для частной производной можно представить как:

$$\frac{\partial}{\partial a_k} D = -2\sum_{m=0}^{M} F(x_m)\cdot v_k(x_m) + 2\sum_{n=0}^{N} a_n \cdot \sum_{m=0}^{M} v_k(x_m)\cdot v_n(x_m)$$

Обратим внимание на две составляющие этого выражения, которые обозначим как ***Bk*** и ***Ck,n***:

$$B_k = \sum_{m=0}^{M} F(x_m)\cdot v_k(x_m)$$
$$C_{k,n} = \sum_{m=0}^{M} v_k(x_m)\cdot v_n(x_m)$$

Легко заметить, что в эти суммы входят функции, которые, либо даны нам как табличные значения для всех своих аргументов ***x***, либо, учитывая известный аналитический вид функций ***v(x)***, могут быть вычислены как значения для такого же набора аргументов ***x***. Таким образом, рассматриваемые суммы, можно рассматривать как наборы констант, значения которых зависят только от индексов ***k*** и ***n***.

Как следствие, итоговый вид для частной производной можно представить следующим образом:

$$\frac{\partial}{\partial a_k} D = -2\cdot B_k + 2\cdot\sum_{n=0}^{N} a_n \cdot C_{k,n}$$

Шаг 10. Построение системы линейных уравнений

На шаге 5, нами было сформулировано условие нахождение таких коэффициентов ***a***, при которых функция ***D*** имеет минимальное значение. Запишем это условие в обозначениях шага 9:

$$\frac{\partial}{\partial a_0} D = 0 = -2\cdot B_0 + 2\cdot\sum_{n=0}^{N} a_n \cdot C_{0,n}$$
$$\cdots$$
$$\frac{\partial}{\partial a_N} D = 0 = -2\cdot B_N + 2\cdot\sum_{n=0}^{N} a_n \cdot C_{N,n}$$

Простейшие преобразования (сокращение множителя 2 и перенос коэффициентов ***Bk*** на противоположную сторону равенства), позволяют эти условия представить в виде системы уравнений:

$$\begin{cases} \sum_{n=0}^{N} a_n \cdot C_{0,n} = B_0 \\ \dots \\ \sum_{n=0}^{N} a_n \cdot C_{N,n} = B_N \end{cases}$$

На шаге 9, мы уже отмечали, что коэффициенты ***Bk*** и ***Ck,n***: не зависят от коэффициентов ***a*** и могут быть предварительно вычислены в виде числовых значений. Таким образом, представленная выше система уравнений является системой линейных уравнений, которые разрешаются методами линейной алгебры.

Решение задачи аппроксимации в ее дискретной форме.

Итак, с учетом условий, которые сформулированы на шаге 10, становится очевидным, что задача аппроксимации разрешима путем нахождения решения системы линейных уравнения относительно неизвестных коэффициентов ***a***:

$$\begin{cases} a_0 C_{0,0} + \dots + a_N C_{0,N} = B_0 \\ a_0 C_{1,0} + \dots + a_N C_{1,N} = B_1 \\ \dots \quad \dots \quad \dots \quad \dots \\ a_0 C_{N,0} + \dots + a_N C_{N,N} = B_N \end{cases}$$

Где:

$$B_k = \sum_{m=0}^{M} F(x_m) \cdot v_k(x_m)$$

$$C_{k,n} = \sum_{m=0}^{M} v_k(x_m) \cdot v_n(x_m)$$

А также:

- Индекс **m : (0, …, M)** – перечисляет точки аргумента функции ***F(x).***
- Индекс **n : (0, …, N)** – перечисляет базисные функции ***v(x)*** или строки системы линейных уравнений.
- Индекс **k : (0, …, N)** – указывает на конкретную строку в системе линейных уравнений.

Интегральная форма задачи аппроксимации

Переход к интегральной форме

Обращаясь к рис.2, легко заметить – что множество точек заключенных между кривыми ***F(x)*** и ***P(x)*** можно рассматривать с позиции площади (в геометрическом смысле этого слова). Иными словами, можно утверждать, что если общая площадь, образованная областями отклонений между функциями ***F(x)*** *и* ***P(x)*** стремиться к нулю, то в предельном случае исходная и аппроксимирующая функция становятся идентичными.

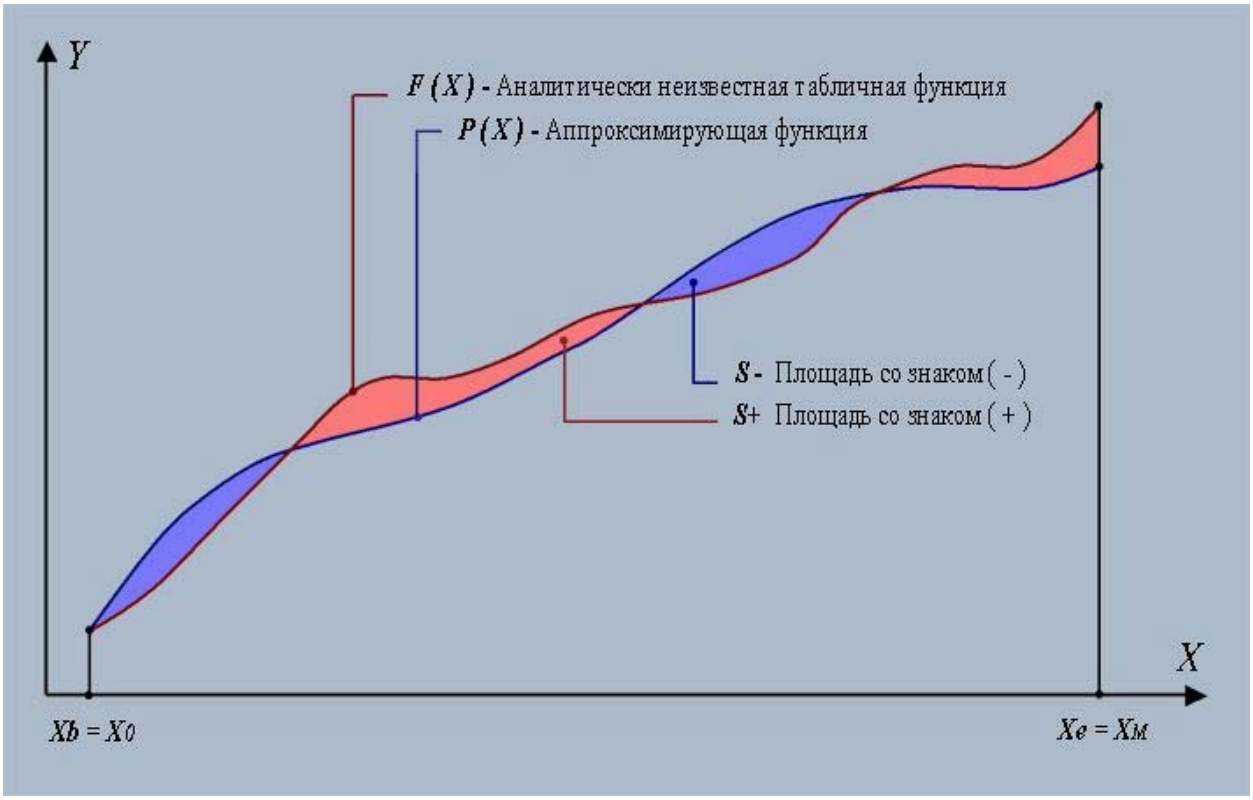


Рис.2. Интегральное отклонение между аппроксимируемой и аппроксимирующей функциями.

Это, естественным образом, приводит нас к идее обобщить в интегральной форме условие минимизации оценки ***D***.

Выполним такое обобщение путем умножения сторон равенства на значение:

$$\Delta x = x_{i+1} - x_i$$

В результате получим:

$$D\Delta x = \sum_{m=0}^{M} ((F(x_m) - \sum_{i=0}^{k} a_i v_i(x_m))^2 \Delta x$$

При устремлении ***Δx*** к нулю мы автоматически получаем нашу оценку в интегральной форме.

$$S = \lim_{\Delta x \to 0} (D\Delta x) = \int_{Xb}^{Xe} ((F(x) - \sum_{i=0}^{k} a_i v_i(x))^2 dx$$

Геометрический смысл такой оценки можно сформулировать как площадь, которая образована квадратичной разностью между функциями ***F(x)*** *и* ***P(x)***.

Таким образом, в интегральной форме, условие для поиска коэффициентов ***a,*** при котором достигается минимальное значение интегральной оценки ***S***, можно записать в виде:

$$\frac{\partial}{\partial a_0} S(F(x), (a_0, ..., a_N), (v_0(x), ..., v_N(x))) = 0$$

$$\cdots$$

$$\frac{\partial}{\partial a_N} S(F(x), (a_0, ..., a_N), (v_0(x), ..., v_N(x))) = 0$$

Учитывая, что переход к интегральной форме, сводится фактически к замещению обычной суммы по аргументу ***X*** на сумму интегральную, то такой переход можно выполнить во всех выражениях, которые рассматривались в дискретной форме решения задачи аппроксимации, путем простой замены:

$$\sum_{m=0}^{M} \varphi(x_m) \to \int_{Xb}^{Xe} \varphi(x)\, dx$$

Решение задачи аппроксимации в интегральной форме

Если опустить последовательную цепочку шагов, которые мы рассматривали при решении задачи аппроксимации в дискретной форме, то с учетом перехода от индексов дискретных значений аргумента ***x***, к границам интегрирования:

$$x_0 = x_b$$
$$x_M = x_e$$

Решение задачи аппроксимации в интегральной форме можно сразу представить в следующем виде:

Система линейных уравнений:

$$\begin{cases} a_0 C_{0,0} & +...+ & a_N C_{0,N} & = & B_0 \\ a_0 C_{1,0} & +...+ & a_N C_{1,N} & = & B_1 \\ ... & ... & ... & ... & \\ a_0 C_{N,0} & +...+ & a_N C_{N,N} & = & B_N \end{cases}$$

Константы при независимых переменных ***a*** системы линейных уравнений:

$$B_k = \int_{Xb}^{Xe} F(x) \cdot v_k(x)\, dx$$

$$C_{k,n} = \int_{Xb}^{Xe} v_k(x) \cdot v_n(x)\, dx$$

Где:

- Индекс **n : (0, …, N)** – перечисляет базисные функции ***v(x)*** или строки системы линейных уравнений.
- Индекс **k : (0, …, N)** – указывает на конкретную строку в системе линейных уравнений.

Выбор базисных функций

Выполняя решение задачи в дискретной и интегральной форме, единственное требование, которое мы накладывали на аппроксимирующую функцию ***P(X)***, было требование об известном аналитическом виде базисных функций ***v(x)***:

$$P(x) = \sum_{n=0}^{N} a_n v_n(x)$$

Такая широкая трактовка класса функций ***v(x)*** оказалась возможной в силу того, что мы не применяли никаких операций, требующих учета свойств и внутренней структуры этих функций. Строго говоря, нам было достаточно только того, что эти функции изображали некоторое преобразование над аргументом ***X***.

Таким образом, множество функций ***v(x)*** удалось сохранить достаточно широким, что безусловно является достоинством для математического анализа, но, в то же время, является явным недостатком для практического применения.

Если акцентировать наши усилия на практическом применении, то потребуется вводить (для получения практических результатов) некоторые дополнительные условия или требования, которые естественным образом ограничат множество этих функций.

Задача аппроксимации в ортогональных базисах

Одним из таких подходов, который (в числе прочего) существенно упрощает практическое решение задач аппроксимации, является подход, накладывающий на множество функций ***v(x)***, требования взаимной ортогональности. Такое требование, в контексте нашей задачи, приводит к ее существенному упрощению и может быть сформулировано, в его самой простой форме, как:

$$C_{k,n} = \int_{Xb}^{Xe} v_k(x) \cdot v_n(x)\, dx = \begin{cases} 0, & если \quad k \neq n \\ C, & если \quad k = n \end{cases}$$

При выполнении приведенного требования, система линейных уравнений, необходимых нам для вычисления коэффициентов функции ***P(X)***, приобретает вид:

$$\begin{cases} a_0 C & +0 & +...+ & +0 & = B_0 \\ 0 & +a_1 C & +...+ & +0 & = B_1 \\ 0 & +...+a_k C & +...+ & +0 & = B_k \\ 0 & +...+ & +0 & +a_N C & = B_N \end{cases}$$

Очевидно, что наша система не только упростилась, но и приобрела уже полностью решенный характер. Иными словами, любой искомый коэффициент может быть вычислен как:

$$a_k = \frac{B_k}{C} = \frac{1}{C} \int_{Xb}^{Xe} F(x) \cdot v_k(x)\, dx$$

Важное замечание. Особо следует подчеркнуть, что такой подход, приводит к тому, что коэффициенты ***a*** могут вычисляться независимо друг от друга. Важность этого замечания можно показать следующим образом:

Пусть, у нас имеется результат решенной задачи аппроксимации:

$$F(x) \cong P(x) = \sum_{n=0}^{N} a_n v_n(x)$$

Если базисные функции ***v(x)***, взаимно ортогональны, то из функции ***P(X***) можно исключить любую функцию ***v(x)*** вместе с ее коэффициентом, и это не потребует

пересчета всех остальных коэффициентов. Единственным следствием такого исключения будет повышение ошибки аппроксимации пропорционально весу исключенной функции.

В том же случае, когда базисные функции ***v(x)*** не являются взаимно ортогональными, исключение некоторой функции ***v(x)*** из функции ***P(X***) потребует полного перерасчета всей задачи. Это обусловлено тем, что изменение размерности базиса приведет к изменению размерности системы линейных уравнений, а следовательно, к необходимости ее перерасчета.

Классическое решение задачи аппроксимации в ортогональном базисе

Для понимания некоторых особенностей ортогональных наборов функций, проделаем ряд преобразований, которые повторят путь решения задачи аппроксимации в ортогональном базисе максимально компактным образом.
Итак, для исходного приближения:

$$F(x) \cong \sum_{n=0}^{N} a_n v_n(x)$$

- Умножим, обе стороны этого выражения на произведение двух функций:

(роль функции гамма, будет определена ниже по тексту):

$$F(x)\cdot\gamma(x)\cdot v_k(x) \cong \sum_{n=0}^{N} a_n \cdot \gamma(x)\cdot v_k(x)\cdot v_n(x)$$

- Проинтегрируем обе стороны полученного выражения в некоторой области определения аргумента*:*

$$\int_{Xb}^{Xe} F(x)\cdot\gamma(x)\cdot v_k(x)\,dx \cong \sum_{n=0}^{N} a_n \int_{Xb}^{Xe} \gamma(x)\cdot v_k(x)\cdot v_n(x)\,dx$$

- В этом случае, условие ортогональности примет следующий вид:

$$\int_{Xb}^{Xe} \gamma(x)\cdot v_k(x)\cdot v_n(x)\,dx = \begin{cases} 0, & если \quad k \neq n \\ C, & если \quad k = n \end{cases}$$

- В результате таких формальных преобразований, легко получить уже знакомое нам решение:

$$a_k \cong \frac{1}{C}\int_{Xb}^{Xe} F(x)\cdot\gamma(x)\cdot v_k(x)\,dx$$

Особенностью приведенных преобразований, является включение в них специальной (весовой) функции $\gamma(x)$, роль которой расширить множество функций, способных удовлетворять условию ортогональности, либо обеспечить равенство $C = 1$.

На сегодняшний день, результаты исследования функций, позволили получить множество наборов ортогональных функций. Вопросы нахождения таких наборов находятся в компетенции специальных разделов математики и выходят за рамки данной работы. По этой причине, мы кратко упомянем только незначительный ряд классических ортогональных наборов. В числе такого ряда классических наборов ортогональных функций, как правило, называют:

- Многочлены Лежандра
- Многочлены Чебышева
- Многочлены Лагерра
- Многочлены Эрмита
- Функции, образующие ряды Фурье

Примеры классических ортогональных функций

Многочлены Чебышева

Основной особенностью, которая определила включение многочленов Чебышева, в состав примеров, является возможность записать функции Чебышева как в тригонометрической, так и в степенной форме.

Аналитическая запись функции.

Тригонометрическая форма:

$$v_k(x) = \cos(k \cdot \arccos(x))$$

Степенная форма. Степенную форму функций Чебышева представим рекуррентными правилами их вычисления, которые позволяют оптимизировать алгоритмы вычисления значений этих функций:

$$v_0(x) = 1$$
$$v_1(x) = x$$
$$v_{k+1}(x) = 2x \cdot v_k(x) - v_{k-1}(x)$$

Весовая функция.

$$\gamma(x) = \frac{1}{\sqrt{1 - x^2}}$$

Выполнение условий ортогональности и интервал ортогональности.

$$\int_{Xb}^{Xe} \gamma(x)\cdot v_k(x)\cdot v_n(x)\,dx = \begin{cases} 0, & если \quad k \neq n \\ \dfrac{\pi}{2}, & если \quad k = n \neq 0 \\ \pi, & если \quad k = n = 0 \end{cases}$$

Где:

$$Xb = -1$$
$$Xe = +1$$

Функции, образующие ряды Фурье

Основной особенностью, которая определила включение в состав примеров функций, образующих ряды Фурье, является широкое использование рядов Фурье в различных технических приложениях.

Аналитическая запись функции.

$$v_k(x) = a_k \sin(kx) + b_k \cos(kx)$$

При этом функцию аппроксимации ***P(X)***, обычно представляют в форме с предварительно вычисленным для нулевого индекса значением ***v(x)***. То есть:

$$P(x) = b_0 + \sum_{k=1}^{\infty} a_k \sin(kx) + \sum_{k=1}^{\infty} b_k \cos(kx)$$

Весовая функция.

$$\gamma(x) = 1$$

Выполнение условий ортогональности и интервал ортогональности.

Интервал ортогональности для функций Фурье определяется следующим образом:

$$Xe = \alpha + 2\pi$$
$$Xb = \alpha, \quad где : \alpha - любая \quad начальная \quad фаза$$

Выполнение условий ортогональности для функций Фурье, традиционно доказывают для отдельных произведений, которые возникают при перемножении функций ***v(x)***.

$$\int_{Xb}^{Xe} \cos(kx)\cdot\cos(nx)\,dx = \begin{cases} 0, & если \quad k \neq n \\ \pi, & если \quad k = n \end{cases}$$

$$\int_{Xb}^{Xe} \sin(kx)\cdot\sin(nx)\,dx = \begin{cases} 0, & если \quad k \neq n \\ \pi, & если \quad k = n \end{cases}$$

$$\int_{Xb}^{Xe} \sin(kx)\cdot\cos(nx)\,dx = 0$$

Таким образом, решение задачи аппроксимации в базисе функций Фурье можно (пропуская очевидные промежуточные преобразования) записать в виде:

$$a_k = \frac{1}{\pi}\cdot \int_{\alpha}^{\alpha+2\pi} F(x)\sin(kx)\,dx$$

$$b_k = \frac{1}{\pi}\cdot \int_{\alpha}^{\alpha+2\pi} F(x)\cos(kx)\,dx$$

$$b_0 = \frac{1}{2\pi}\cdot \int_{\alpha}^{\alpha+2\pi} F(x)\,dx$$

Примечание.

Как уже отмечалось, базис Фурье для дискретного спектра имеет вид:

$$v_k(x) = a_k \sin(kx) + b_k \cos(kx)$$

Обратим внимание, что составляющие этого выражения достаточно просто интерпритировать как проекции векторов с длинами ***ak*** и ***bk*** соответственно спроектированные на оси координат ***Y*** и ***X***. Такая особенность позволяет представить результат аппроксимации в базисе Фурье в более компактной форме, которая, как правило, применяется в большинстве инженерных решениях.

Для получения упомянутой компактной формы, рассмотрим известные тригонометрические формулы:

$$a \cdot \sin(x) + b \cdot \cos(x) = \sqrt{a^2 + b^2} \cdot \sin(x + \varphi)$$

Где:

$$\sin(\varphi) = \frac{b}{\sqrt{a^2 + b^2}}$$

С помощью приведенных формул окончательный результат аппроксимации рядом Фурье в следующем компактном виде:

$$P(x) = b_0 + \sum_{k=1}^{\infty} L_k \cdot \sin(kx + \varphi_k)$$

Где:

$$L_k = \sqrt{a_k^2 + b_k^2}$$

Перенормировка интервала ортогональности

Рассмотренные нами примеры ортогональных функций, показывают нам вполне конкретные интервалы на которых выполняется условие ортогональности.

Если ранее, когда условия ортогональности только формулировались, это не ограничивало область определения функции ***F(X)***, поскольку сама ***F(X)*** неявным образом задавала этот интервал, то при конкретизации базиса, картина зеркально изменилась. При этом, становится очевидным, что практически для всех функции ***F(X)*** потребуется приведение области их определения к конкретному интервалу ортогональности.

Такая задача получила название перенормировки интервала ортогональности и сводится к нахождению такого масштаба, посредством которого область определения функции ***F(X)*** целиком отображается в интервал ортогональности.

Для решения задачи перенормировки определим функцию ***F***, как функцию от независимой переменной t, с областью определения (t_b, t_e).

В этом случае можно показать, что независимые переменные ***X*** и t могут быть связаны простым отношением, которое обеспечит нам необходимое масштабное преобразование:

$$x - x_b = \frac{x_e - x_b}{t_e - t_b} \cdot (t - t_b)$$

Дальнейшая подстановка данного выражения во все, рассмотренные для ортогональных функций, преобразования позволит осуществить переход, практически к произвольным границам области определения функции ***F(t)***

Примечание. Приведем пример перенормировки.

Наиболее известным из примеров перенормировки, является перенормировка ортогональных функций Фурье для решения задач анализа сигналов:

$$x = \frac{2\pi}{t_e - t_b} \cdot t = \frac{2\pi}{T} = \omega \cdot t$$

При такой перенормировке, рассмотренные ранее выражения для аппроксимации рядом Фурье примут вид:

$$F(t) \cong b_0 + \sum_{k=1}^{N} a_k \sin\ (k\,\omega\,t) + \sum_{k=1}^{N} b_k \cos\ (k\,\omega\,t)$$

Соответственно, коэффициенты или решение задачи, можно представить следующим образом:

$$a_k = \frac{2}{T} \cdot \int_{t_b}^{t_e} F(\omega t)\sin(k\omega t)\,dt$$

$$b_k = \frac{2}{T} \cdot \int_{t_b}^{t_e} F(\omega t)\cos(k\omega t)\,dt$$

$$b_0 = \frac{1}{T} \cdot \int_{t_b}^{t_e} F(\omega t)\,dt$$

Как результат, мы перешли от границ заданных в угловых единицах, к границам, которые заданы в необходимых нам границах времени. Аналогичным образом выполняется перенормировка для границ заданных расстоянием, что обычно применяется при обработке растровых изображений.

Оценка качества решения задачи аппроксимации

Рассматривая задачу аппроксимации в дискретной форме (на шаге 2) мы вводили отклонение между апроксисмируемой и апроксимирующей функцией в форме простой разности:

$$d(x) = F(x) - P(x)$$

Легко заметить, что такая разность равнозначна определению абсолютной ошибки со знаком, которое дается в метрологии. Как правило, помимо абсолютной ошибки, метрология также вводит относительные ошибки как в общей, так и в приведенной к концу шкалы форме. В рамках уже принятых нами обозначений, это можно записать в следующем виде:

$$\delta = \frac{d(x)}{F(x)} \cdot 100 \ \%$$

$$\delta_{np} = \frac{\max \ (d(x))}{F(x_e)} \cdot 100 \ \%$$

На первый взгляд, такие инструменты оценки качества можно рассматривать едва ли не как единственные при оценке решения нашей задачи. Действительно, до решения конкретной задачи аппроксимации мы сталкиваемся со значительным числом неопределенностей:

- К функции ***F(X)*** мы не предъявляли, кроме общих, каких либо особых требований или ограничений, влияющих на качество аппроксимации.
- Практически всегда, функции ***F(X)*** включают в себя всевозможные шумы и ошибки, поскольку, как правило, являются результатами тех или иных измерений.
- До получения конкретного экземпляра функции ***F(X)***, практически ничего нельзя сказать о выборе (по типу и размеру) базиса функции ***P(X)***.

Естественным образом возникает вопрос, можно ли, получив решение конкретной задачи аппроксимации, использовать те или иные оценки для последующего улучшения качества решения этой задачи?

Задача аппроксимации, как задача теории вероятностей

Рассматривая перечисленные неопределенности, достаточно просто прийти к выводу, что в первую очередь необходимо рассмотреть возможность применения к задаче аппроксимации известного инструментария теории вероятностей.

Изложим кратко основные моменты теории случайных процессов

Случайным процессом ***Z(t)*** называется процесс (функция), значение которого при любом фиксированном значении $t = t_0$ (сечение случайного процесса) является случайной величиной $Z(t_0)$.

> ***Примечание.*** Все многообразие случайных процессов, а также различных подходов к их анализу достаточно полно рассмотрено в книге [3].

Конкретный вид процесса, полученный в результате опыта с номером ***k***, называется реализацией $Z_k(t)$ случайного процесса. При проведении серии, включающей ***K*** – опытов, можно получить группу или семейство реализаций случайной функции. Такое семейство наглядно представимо с помощью следующей таблицы:

$Z_1(t)$	$Z_2(t)$	…	$Z_k(t)$	…	$Z_K(t)$
$p_1(t)$	$p_2(t)$	…	$p_k(t)$	…	$p_K(t)$

Где $p_k(t)$ описывает закон распределения вероятности для значений некоторой реализации $Z_k(t)$ в ее произвольном сечении ***t***.

Математическим ожиданием случайного процесса $Z(t)$ будем называть неслучайную функцию $M[Z(t)]$, которая при любом значении аргумента ***t*** равнна математическому ожиданию, полученному по значениям случайного процесса в сечении ***t***.

В соответствии с приведенным определением, математическое ожидание для реализаци с номером ***k*** можно представить выражением:

$$M[Z_k(t)] = \sum_t Z_k(t)\, p_k(t)$$

С помощью определения математического ожидания, можно также определить центрированный случайный процесс или центрированную реализацию случайного процесса:

$$\overset{o}{Z}_k(t) = Z_k(t) - M[Z_k(t)]$$

Кроме того, в состав классических инструментов анализа как случайного процесса, так и его реализации также входят:

1. Начальные, а также центральные моменты реализации ***n*** - порядка

$$\mu_n(t) = M[(Z_k(t))^n]$$
$$\overset{o}{\mu}_n(t) = M[(\overset{o}{Z}_k(t))^n] = M[(Z_k(t) - M[Z_k(t)])^n]$$

2. Дисперсия, как центральный момент реализации второго порядка:

$$D[Z_k(t)] = M[((Z_k(t) - M[Z_k(t)])^2]$$

Как следует из приведенного выражения, дисперсией реализации случайного процесса является неслучайная функция $D[Zk(t)]$, которая при любом значении аргумента t_0 равна дисперсии соответствующего сечения случайного процесса $Zk(t_0)$

3. Среднее квадратическое отклонение случайного процесса или его реализации, которое определяют как арифметическое значение корня квадратного из соответствующей дисперсии:

$$\tau[Z_k(t)] = \sqrt{D[(Z_k(t)]}$$

Из приведенных определений, очевидно, что наиболее полно случайные процессы описываются многомерными законами. Однако, в инженерных приложениях стараются понизить сложность и выразить многомерные законы через последовательный анализ различных одномерных сечений. Это обусловлено тем, что в инженерных приложениях, как правило, рассматриваются объекты, которые в процессе их наблюдения не изменяют случайным образом свою внутреннюю природу (внутреннюю модель). Для таких объектов, случайных характер наблюдаемых свойств (наблюдаемых как значения случайного процесса в фиксированной точке либо в виде семейста реализаций случайного процесса), определяется ошибками моделирования, ошибками измерений, либо шумами различной природы, законы распределения которых обычно очень редко связаны (не коррелируют) между сабой.

Привязка положений теории случайных процессов к задаче аппроксимации

Давайте сопоставим:

- Использованную нами оценку качества решения задачи аппроксимации

$$D = \sum_{m=1}^{M} ((F(x_m) - P(x_m))^2$$

- Дисперсию ***k*** - той реализации некоторого случайного процесса, заменив независимую переменную ***t*** на ***x***:

$$D[Z_k(x)] = M[((Z_k(x) - M[Z_k(x)])^2]$$

- А также учтем, что при такой замене независимой переменной, выражение для математического ожидания примет вид:

$$M[Z_k(x)] = \sum_x Z_k(x)\, p_k(x)$$

При таком сопоставлении, достаточно легко заметить сходство отношений, которые определяют нашу оценку ***D,*** сконструированную для решения задачи аппроксимации с дисперсией.

Следует также отметить, что сходство становится еще полнее, если принять равновероятным закон распределения $p_k(x)$ для всех J – точек некоторого, рассматриваемого сечения:

$$p_k(x) = \frac{1}{J}$$

Действительно, при таком равновероятном законе распределения математическое ожидание принимает вид обычного математического усреднения:

$$\overset{*}{M}[Z_k(x)] = \frac{1}{J}\sum_{j=1}^{J} Z_k(x_j)$$

В чем же выражается упомянутое сходство? Ответ на этот вопрос можно сформулировать следующим образом:

$$\begin{cases} Z_k(x) \sim F(x) \\ \overset{*}{M}[Z_k(a_0,..., a_N, x)] \sim P(a_0,..., a_N, x) \end{cases}$$

Очевидно, что в контексте выражения вычисляющего дисперсию, такие сопоставления позволяют сделать вывод, что решая задачу аппроксимации как задачу минимизации оценки по ее независимым параметрам ($a_0, \dots, a_N$), мы фактически минимизируем дисперсию, то есть, минимизируем как случайный характер выбора базиса для аппроксимации $P(X)$, так и случайный характер выбора его размерности.

Вывод 1. Оценкой качества результатов аппроксимации следует считать арифметическое значения корня квадратного (среднее квадратическое отклонение), взятого от вычисленного значения оценки D, после выполнения конкретной аппроксимации:

$$\tau = \sqrt{D}$$

Вывод 2. Среднее квадратичное отклонение по оценке D включает минимизированную составляющую случайных свойств выбора базиса и, к сожалению, полную составляющую случайных свойств, вносимых шумами.

Понижение влияния шумов на результаты аппроксимации

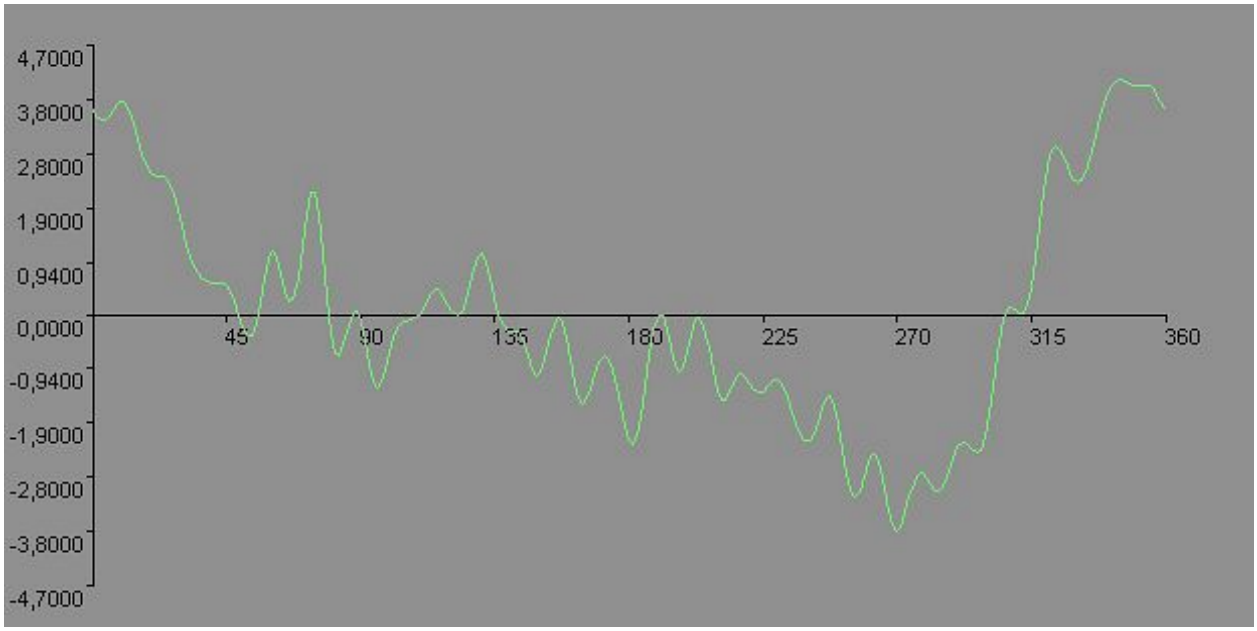
Как правило, еще до начала аппроксимации, шумовые (случайные) составляющие исходных данных стараются максимально понизить, выполняя статистическую обработку серии измерений для каждой точки функции $F(X)$.

Тем не менее, часто встречаются случаи, когда статистическая обработка каждого значения исходной функци $F(X)$ оказывается невозможной, например, в силу того, что такая функция была передана нам сторонними исследователями, либо стоимость измерений (накопления статистики) черезвычайно высока.

В такой ситуации, выполнив серию аппроксимаций и достигнув минимума дисперсии, путем подбора вида и размерности базиса для аппроксимации ***P(X)***, мы вынуждены оценивать факторы шума анализируя вид функции ошибки:

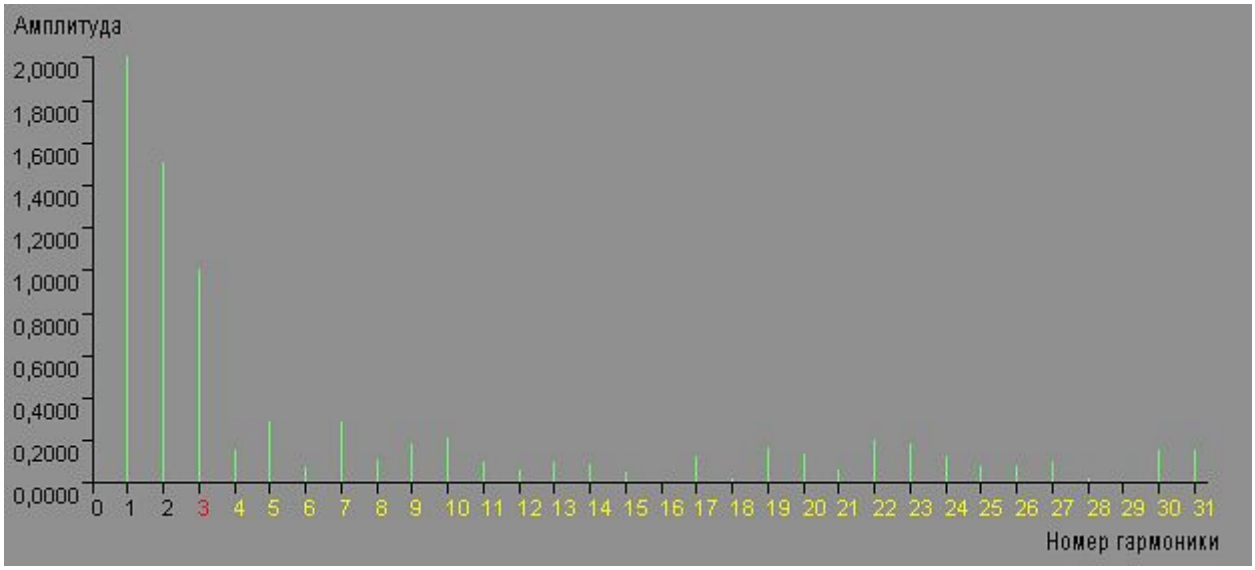
$$d(x) = F(x) - P(x)$$

Типичный вид такой функции ошибки представлен на следующем графике



Для выявления смысла тех или иных изменчивостей, которые отражены на графике ***d(x)*** аппроксимируем эту функцию рядом Фурье. При использовании компьютерных средств аппроксимации, такое предложение не является черезмерным усложнением задачи и вполне может быть рекомендовано для инженерного применения.

В случае нашего примера, соответствующее разложение функции в ряд Фурье может быть представлено в виде следующего спектра:

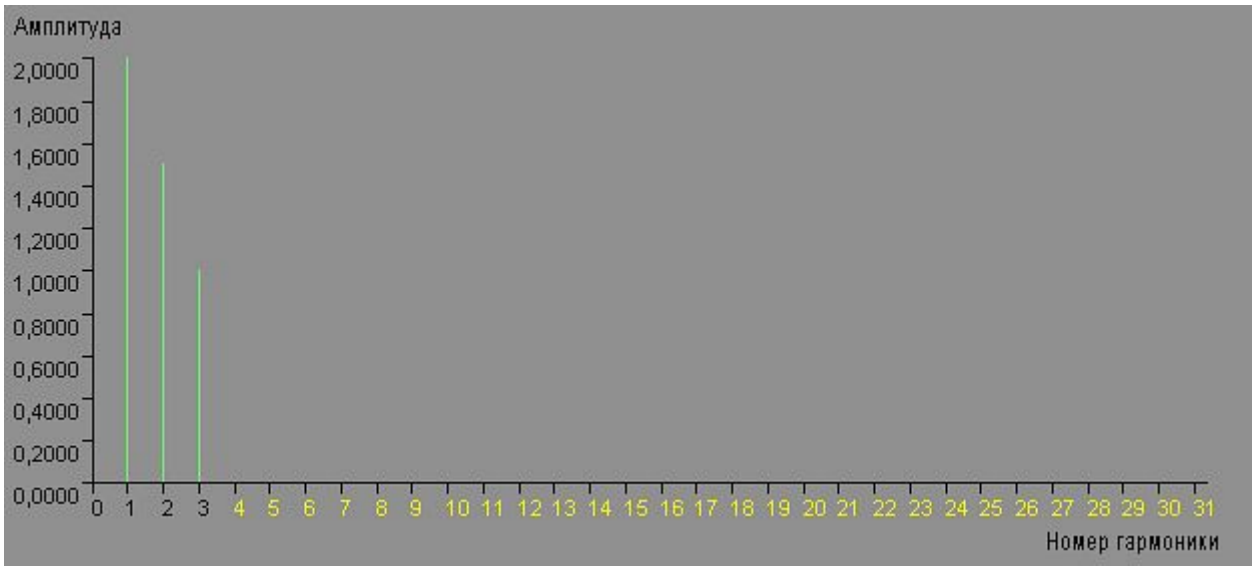


Обратим внимание, что амплитуды гармоник, начиная с четвертой, имеют приблизительно равный, однако хаотический характер, более того, имеют существенно меньший вес по отношению к амплитудам гармоник с номерами от нулевой до третьей.

Такое распределение амплитуд гармоник, позволяет сделать допущение о принадлежности малых гармоник, начиная с четвертой, к шумовому спектру. Естественно, речь идет только о допущении, которое основывается:

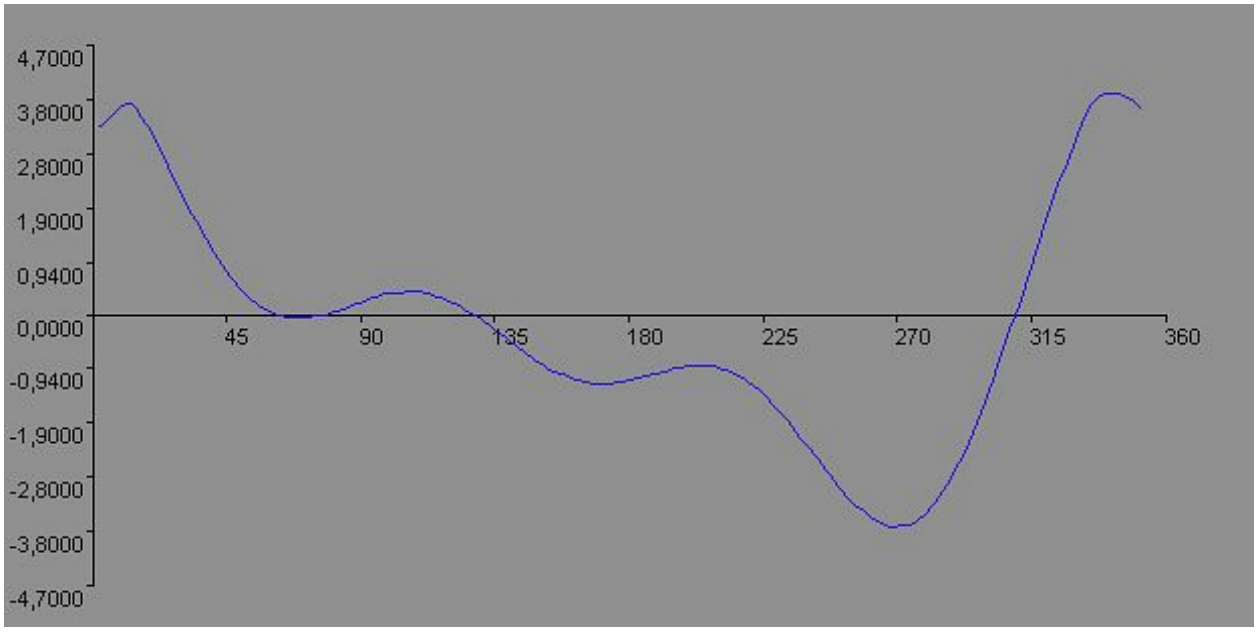
- Стационарным характером измеряемых свойств большинства инженерных объектов.
- Свойством шума обладать широким равномерным спектром .

Если подобное допущение можно обосновать характером свойств исследуемого объекта, то целесообразно выполнить фильтрацию высших гармоник в спектре функции ошибки, как это показнано на примерном графике ниже.



После выполнения фильтрации, функция ошибки приобретает более информативный вид, поскольку отображает только существенную часть изменчивостей оставшихся в составе ошибки.

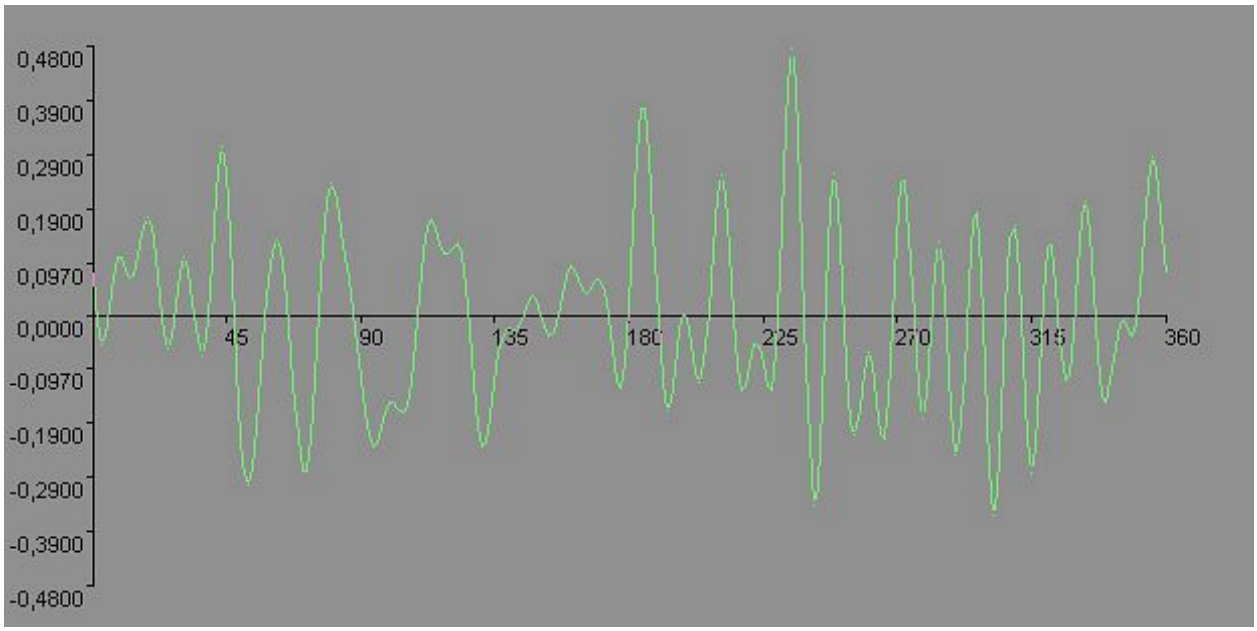
На рисунке, который предстален ниже, показан вид такой отфильтрованной функции ошибки:



Очевидно, что для отфильтрованной функции ошибки ***d *(x)***, гораздо проще подобрать соответствующую аппроксимацию, и представить окончательный результат решения задачи в виде:

$$F(x) = P(x) + d^{*}(x)$$

На следующем графике показаны остаточные шумы в окончательном результате.



Эвристические приемы повышения качества аппроксимации

Проводя параллель между оценкой ***D*** и дисперсией, мы также неявно рассмотрели и один из таких эвристических приемов, который позволяет уточнить выбор аппроксимирующего базиса и его размерности. Такая параллель позволила придать выбору базиса и его размерности случайный характер, а также сформулировать оценку успешности этого выбора. При этом, мы показали прием (фильтрация спектра функции ошибки), который в условиях шумов, позволяет уточнить как вид базиса так и его размерность. Очевидно, что показанный прием носит эвристический характер, поскольку его эффективность определяется полнотой опыта исследователя в сопоставлении внешнего вида графиков различных функций и (или) их производных.
К сожалению, задачи сопоставления различных пространственных форм, относятся к классу задач распознования образов и оказываются наиболее сложными для их решения численными методами. Как следствие, в инженерной практике такие задачи пока решаются путем проб и ошибок.

Еще одной причиной, приводящей к эвристическим решениям, является возможная неравномерность требований к точности аппроксимации в области ее определения. Это легко пояснить на примере технических систем стабилизации некоторого параметра. Например, нам необходимо с минимальными техническими затратами (то есть, ограниченный выбор базиса и его размерности) точно стабилизировать некоторый параметр в узком диапазоне его значений и при этом иметь возможность выйти на значение стабилизации из широкой области стартовых значений этого параметра. По существу, такая постановка задачи, требует минимизации значений ошибки ***d(x)*** в области стабилизации параметра.

Рассматривая аппроксимацию в ортогональных базисах, мы уже упоминали специальную весовую функцию “гамма”. Функции “гамма” можно придать несколько смыслов:

- Смысл инструмента, который обеспечивает выполнение условия ортогонализации и (или) нормировки интервала ортогональности.
- Смысл фильтра, если рассматривать выражение, вычисляющее коэффициенты при базисных функциях, как свертку.
- Смысл закона распределения вероятности, если рассматривать оценку ***D*** как дисперсию.

Это подталкивает нас к идее использовать функцию “гамма” для управления влиянием (весом) различных точек, которые входят в состав оценки ***D***, придавая этой функции смысл - вероятность появления аргумента ***x*** с некоторым конкретным значением.

Однако, в ортогональных разложениях смысл и конкретный вид такой функции уже зафиксированы выполнением условия ортогональности, а, следовательно в ортогональных разложениях данную функцию недопустимо произвольно модифицировать.

В то же время, при решении задачи аппроксимации в неортогональном базисе, мы не связаны условием, которое запрещает свободные модификации функций базиса, а, следовательно, мы можем включать необходимую (однако всюду положительную) функцию “гамма” как общий множитель оценки ***D***:

$$D = \sum_{m=1}^{M} [((F(x_m) - \sum_{n=0}^{N} a_n v_n(x_m))^2] \cdot \gamma(x_m)$$

Как следствие, выражения, которые определяют коэффициенты системы линейных уравнений, примут вид:

$$B_k = \sum_{m=1}^{M} F(x_m) \cdot \gamma(x_m) \cdot v_k(x_m)$$

$$C_{k,n} = \sum_{m=1}^{M} \gamma(x_m) \cdot v_k(x_m) \cdot v_n(x_m)$$

Таким образом, функцию “гамма” можно определить как вероятность текущего состояния системы (например, системы стабилизации некоторого параметра) и, исходя из анализа такой системы, сформулировать ее в аналитическом виде.

Понятно, что описанный подход можно достаточно строго формализовать средствами математики (например, исходя из аналитической модели объекта управления), однако реализация такой формализации, особенно в обощенном виде, будет крайне тяжеловесной для большинства инженерных или практических задач, связанных с аппроксимацией табличных функций. По этой причине, особенно в условиях незначительной стоимости получения значений для отдельных точек табличной функции, вместо функции “гамма” применяют неравномерный шаг для аргумента функции ***F(x)***, а задачу аппроксимации решают не в интегральной, а в дискретной форме.

При таком подходе, эвристически добавляя новые точки в те области, в которых необходимо получить минимальные значения отклонений, постепенно понижают значения ***d(x)*** в этих областях до приемлимых уровней.

Действительно, как при составлении оценки ***D***, так и при решении задачи аппроксимации, нам нигде не потребовалось налагать требование постоянного шага для аргумента ***x***. Это означает, что задача аппроксимации разрешима для табличных функций с произвольным шагом. Однако если в дискретной форме оценка ***D***, как сумма квадратов отклонений, чувствительна к неравномерному шагу аргумента, то интегральной форме эта эта чувствительность нивелируется переходом от суммы к площади. Это обусловлено тем, что для конкретной оценки ***D***, значение площади зависит только от границ интегрирования. Следовательно, для сохранения чувствительности к неравномерному шагу, задачу аппроксимации удобнее решать в ее дискретной форме. Следует также отметить, если для функции ***F(x)*** можно найти дополнительную функцию, которая равномерный шаг аргумента трансформирует в неравномерный и наоборот, то выбор неравномерного шага равнозначен определению в неявной форме такой дополнительной функции, эквивалентной функции “гамма”.

Укрупненные алгоритмы решения задачи аппроксимации

В процессе рассмотрения путей решения задачи аппроксисмации, нами было получено две формы ее решения, а именно – неортогональное и ортогональное решение. Рассмотрим эти формы решения последовательно.

Дискретное решение в неортогональном базисе.

Итак, дискретное решение задачи в неортогональной базисе было получено нами как решение системы линейных уравнений относительно неизвестных масштабных коэффициентов ***a***:

$$\begin{cases} a_0 C_{0,0} + \ldots + a_N C_{0,N} = B_0 \\ a_0 C_{1,0} + \ldots + a_N C_{1,N} = B_1 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \\ a_0 C_{N,0} + \ldots + a_N C_{N,N} = B_N \end{cases}$$

Где:

$$B_k = \sum_{m=0}^{M} F(x_m) \cdot v_k(x_m)$$

$$C_{k,n} = \sum_{m=0}^{M} v_k(x_m) \cdot v_n(x_m)$$

А также:

Индекс **m : (0, …, M)** – перечисляет точки аргумента функции ***F(x)***
Индекс **n : (0, …, N)** – перечисляет базисные функции ***v(x)*** или строки системы линейных уравнений.
Индекс **k : (0, …, N)** – указывает на конкретную строку в системе линейных уравнений.

Алгоритм для вычисления коэффициентов системы линейных уравнений.

Исходные данные. В качестве исходных данных алгоритма определим:

- Функцию ***F(x),*** представленную двухмерным массивом **FTAB[m,i]** размерностью **[0..M, 0..1]**, в котором индекс **m** будет перечислять точки функции ***F(x)***, а индекс **i** соответственно указывать на значение аргумента ***X*** (при **i = 0**) или значение функции ***F(x)***. (при **i = 1**)
- Функцию $v_k(x)$**,** представленную процедурной функцией **v(k,x)**.

Выходные данные. В качестве выходных данных алгоритма будем рассматривать систему линейных уравнений, представленную как двухмерный массив **ESTAB[k, j]** размерностью **[0..N, 0..N+1]**, в котором индекс **k** будет перечислять строки системы линейных уравнений, а индекс **j,** соответственно, указывать на столбцы этой системы.

Задача алгоритма. Задачей алгоритма будем считать, вычисление всех значений массива **ESTAB[k, j]**.

Условие оптимальности. Будем считать алгоритм достаточно оптимальным, если в процессе его выполнения потребуется минимальное число обращений к процедуре **v(k,x)**.

Предварительный анализ.

Шаг1. Рассматривая выражения для вычисления коэффициентов B_k и $C_{k,n}$, следует заметить, что при любой организации алгоритма, нам предстоит вычислить функцию $v_k(x)$, для всех значений ее инденкса **k**, а также для всех значений аргумента ***X***. Более того, анализ системы линейных уравнений показывает, что вычисление $v_k(x)$ будет многократно дублироваться, если алгоритмически построить вычисление коэффициентов B_k и $C_{k,n}$, в форме независимых процедур.

Вывод шага 1. Учитывая, что в современных средствах вычислительной техники память перестала быть критическим и дорогостоящим ресурсом, оптимальным решением можно считать создание и вычисление дополнительного двухмерногго массива **VTAB [m, k]** размерностью **[0..M, 0..N]**. В массиве **VTAB [m, k]** индекс **m** будет перечислять значения независимой переменной ***X***, взятой из массива **FTAB[m,0]**, а индекс **k**,соответственно обозначать, что результат вычислен функцией $v_k(x)$.

Шаг2. Вычисление массива **VTAB [m, k]** можно оптимизировать в части количества операций, если выполнять его вычисление построчно, то есть, вычислять значения всех функции $v_k(x)$.для конкретного значения независимой переменной ***X*** с помощью отдельной процедуры. Пусть такая процедура будет разработана как процедура **VS(x,k)**

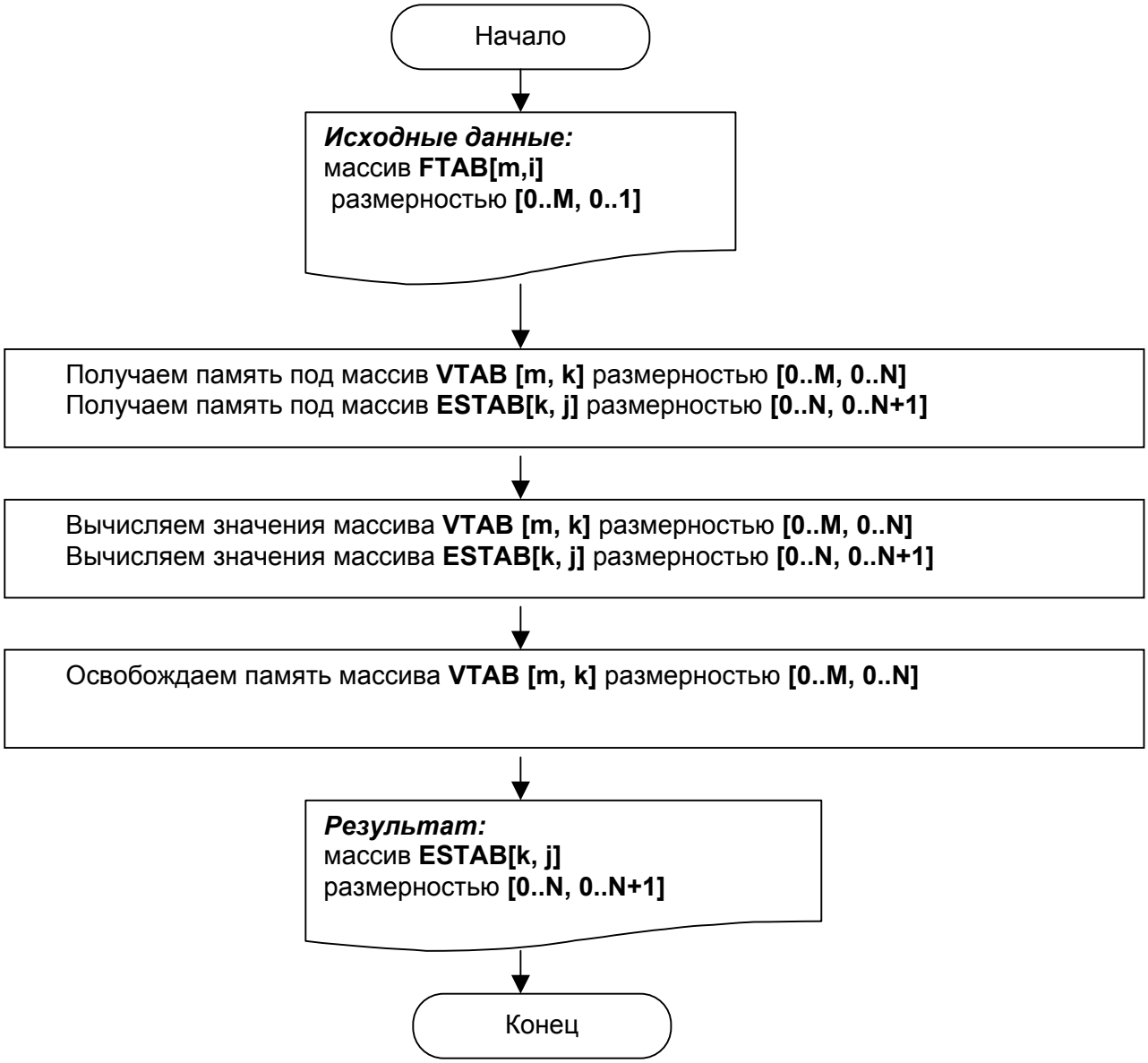
Вывод шага 2. Если базисные функции $v_k(x)$ можно записать в рекуррентной форме, то вычисление строки **VTAB [m, k]** с помощью процедуры **VS(x,k)** можно реализовать простым циклом с минимальным числом операций. Например:

Пусть функции базиса в явном виде определены как $v_k(x)=x^k$, то есть, являются элементами степенного ряда.
В этом случае такие функции можно также представить в рекурентной форме $v_{k+1}(x)=x\ v_k(x)$, где $v_0(x)=1$.
Очевидно, что для вычисления строки **VTAB [m, k]** нам потребуется только ***N-1*** раз выполнить умножение ***x*** на ***x***. Еще более оптимальный эффект получается при применении рекуррентной формы, если базис задан

функциями Чебышева или иными специальными функциями, явный вид которых, как правило, существенно превышает по сложности соответствующие рекуррентные формы.

Шаг3. Теперь мы располагаем всеми числовыми значениями для вычисления коэффициентов B_k и $C_{k,n}$, а следовательно и заполнения массива **ESTAB[k, j]**. Если рассматривать ячейки массива как накопители соответствующих сумм, то задача заполнения ячеек сводится к выполнению трех последовательно вложенных циклов. Внешний цикл определяет строку массива, первый внутренний цикл определяет столбец массива, второй внутренний цикл перечисляет индексы независимой переменной X и выполняет вычисление добавляемых элементов в суммы B_k и $C_{k,n}$.

Укрупненный алгоритм шагов 1...3 в блок – функциональной форме



Шаг4. Окончательное решение. Окончательное решение, то есть, решение сиситемы линейных уравнений (массив **ESTAB**), как првило, выполняется соответствующими библиотечными процедурами. С позиции оптимальности, или наименьшего количества выполняемых операций, следует выбирать процедуры, которые выполняют решение систем линейных уравнений по методу Гаусса.

Решение в интегральном виде для неортогонального базиса.

Решение в интегральном виде незначительно отличается от от алгоритма, который мы рассмотрели для дискретной формы.

Основное отличие заключается в процедурах для вычисления коэффициентов системы уравнений:

$$B_k = \int_{Xb}^{Xe} F(x) \cdot \gamma(x) \cdot v_k(x)\, dx$$

$$C_{k,n} = \int_{Xb}^{Xe} \gamma(x) \cdot v_k(x) \cdot v_n(x)\, dx$$

В этом случае, вместо накопления обычной суммы в ячейках массива **ESTAB[k, j]** (см. шаг 3), будет выполнять накопление суммы состоящей из частных квадратур, вычисляемых по той или иной квадратурной формуле.

В качестве квадратур, обычно рассматривают квадратуры, построенные по ***методу трапеций*** или ***методу Симпсона***. Особенностью таких квадратур является простота их вычисления, особенно в случае равномерного шага для значений аргумента подинтегральных функций.

Для начала, рассмотрим частичные квадратуры, которые вычисляются по методу трапеций, то есть, описывают элементарные площади как трапеции, образованные двумя соседними точками табличных функций:

$$s_m = \frac{y_{m+1} + y_m}{2}(x_{m+1} - x_m)$$

В этом случае, полная квадратура (или значение интеграла) ***S*** при равномерном шаге аргумента может быть представлена суммой ***S****, которая является весьма удобной для вычисления:

$$S = [\frac{y_M + y_0}{2} + \sum_{m=1}^{M-1} y_m] \cdot \Delta x = S^* \cdot \Delta x$$

Принимая равномерный шаг для аргумента, как дополнительное условие, мы можем получить еще более существенные преимущества как с точки зрения необходимого числа операций, так, и это особо важно, с точки зрения точности вычислений.

Для обоснования этого утверждения, рассмотрим некоторую строку системы линейных уравнений, описывающих решение задачи аппроксимации. Пусть эта строка имеет индекс – ***k***.

$$\sum_{n=0}^{N} a_n \cdot C_{k,n} = B_k$$

Если коэффициенты ***C*** и ***B*** вычислены по квадратурным формулам с постоянным шагом, то множитель шага можно сократить, не нарушив равенство:

$$\sum_{n=0}^{N} a_n \cdot C^*_{k,n} \cdot \Delta x = B^*_k \cdot \Delta x$$

То есть:

$$\sum_{n=0}^{N} a_n \cdot C^*_{k,n} = B^*_k$$

Как важное следствие, устраняется операция умножения сторон равенства на малое число, которое в соответствии с переходом к интегральной форме, должно устремляться к нулевому значению. Таким образом, сушественно снижаются неопределенности при разрешении системы линейных уравнений, а также устраняются лишние операции умножения.

Использование квадратур Симпсона (в обмен на более высокую точность вычисления интегралов) несколько увеличивает сложность вычислений, однако для равномерного шага аргумента, также позволяет устранить Δx из процедур, которые вычисляют коэффициенты системы уравнений.

В качестве справки, приведем выражение для вычисления частичной квадратуры по методу Симпсона для равномерного шага аргумента:

$$s_m = (y_m + 4y_{m+1} + y_{m+2}) \frac{(x_{m+1} - x_m)}{3}$$

$$s_m = (y_m + 4y_{m+1} + y_{m+2}) \frac{\Delta x}{3}$$

Рассматривая пути повышения качества аппроксимации, мы уже отмечали, что неравномерность шага аргумента дает нам некоторые эвристические возможности, которые утрачиваются при представлении оценки ***D*** в интегральной форме, то есть, в форме площади. Это позволяет сформулировать приведенный ниже вывод.

Вывод. Требование равномерности шага аргумента табличной функции, для интегральной формы задачи аппроксимации никоим образом не усиливает уже существующие ограничения и требования, однако позволяет существенным образом сократить объем вычислений.

Решение в ортогональном базисе.

Напомним кратко вид решения задачи аппроксимации в ортогональном базисе:

$$a_k \cong \frac{\int_{Xb}^{Xe} F(x)\cdot\gamma(x)\cdot v_k(x)\,dx}{\int_{Xb}^{Xe} \gamma(x)\cdot v_k(x)\cdot v_k(x)\,dx}$$

Отметим важную особенность.

Как правило, определенный интеграл в знаменателе приведенного выражения заранее вычисляется как точная константа ***C***:

$$\int_{Xb}^{Xe} \gamma(x)\cdot v_k(x)\cdot v_n(x)\,dx = \begin{cases} 0, & если \quad k \neq n \\ C, & если \quad k = n \end{cases}$$

Точность вычисления константы ***C*** обеспечивается выполнением интегрирования в аналитическом виде с последующей подстановкой границ интегрирования. Как следствие, мы получаем значение коэффициентов аппроксимации в традиционном виде:

$$a_k \cong \frac{1}{C}\int_{Xb}^{Xe} F(x)\cdot\gamma(x)\cdot v_k(x)\,dx$$

Попытки заменить этот интеграл простой суммой (то есть, сформулировать задачу в чисто дискретной форме, особенно для неравномерного шага по аргументу) требуют проверок выполнения условий ортогональности для всех соотношений индексов базисных функций. К следствие, такие попытки предъявляют особые условия на выбор множества аргументов для табличных функций и вместо ожидаемого снижения сложности вычислений мы получаем результат прямо противоположный.

Вывод. Рассмотренная особенность равнозначна требованию вычисления интеграла в числителе (интеграла с участием табличной функции ***F(x)***) только посредством квадратурых форм, поскольку константа ***C*** в знаменателе имеет размерность площади.

Поскольку вопросы вычисления интегралов с помощью квадратурных формул достаточнно хорошо рассмотрены в различной литературе, кроме того, в предшествующем разделе мы уже уделили внимание таким вычислениям, то вопрос составления алгоритмов такой задачи оставим как вопрос свободного выбора.

Описание программы для аппроксимации табличных функций степенными рядами и рядами Фурье

Программа аппроксимации табличных функций степенными рядами и рядами Фурье разработана в технологии DELPHI и ориентирована на применение в учебном процессе в следующих дисциплинах:

- Вычислительная техника и алгоритмические языки
- Аппаратно – программное обеспечение технических систем
- Основы построения систем в светотехнике
- Компьютерная графика и моделирование

Кроме того, программа также может использоваться в целом ряде других дисциплин, в которых рассматриваются вопросы нахождения аналитического вида характеристик разнообразных объектов (например, различных датчиков, передаточных функций и т.п.) или может применяться для вычисления дискретных спектров сигналов, графических изображений и т.п.

Программа очень проста в использовании, может функционировать на всех платформах Windows 98/2000/XP/2003/VISTA/2008 и не требует инсталляции.
Программа реализована всего одним файлом (на данный момент 2011г., это файл Approx205.exe, версия программы 2.05, размер файла 816К) и включает:

- Собственный редактор табличных функций;
- Средста экспорта - импорта исходных данных и результатов в текстовые файлы формата *.TXT;
- Средства графического отображения исходных данных и результатов (результатов аппроксимации и спектров);
- Позволяет сохранять все графические отображения в виде файлов формата *.JPG;
- Средства ведения и сохранения журнала всех операций и отчетов;
- Спедства отображения аппроксимирующей функции на базе вариантной фильтрации ее спектра.

В текущей версии программы используются неоптимизированные алгоритмы для вычисления коэффициентов системы линейных уравнений и коэффициентов аппроксимации. Это обусловлено ориентацией программного текста на учебный процесс.

Так например, для аппроксимации в дискретом виде с неортогональным базисом $V(X)^n$, исходный текст процедуры, составляющей матрицу системы линейных уравнений, а также инициирующий решение этой системы методом Гаусса, реализован поэлементным заполнением ячеек матрицы:

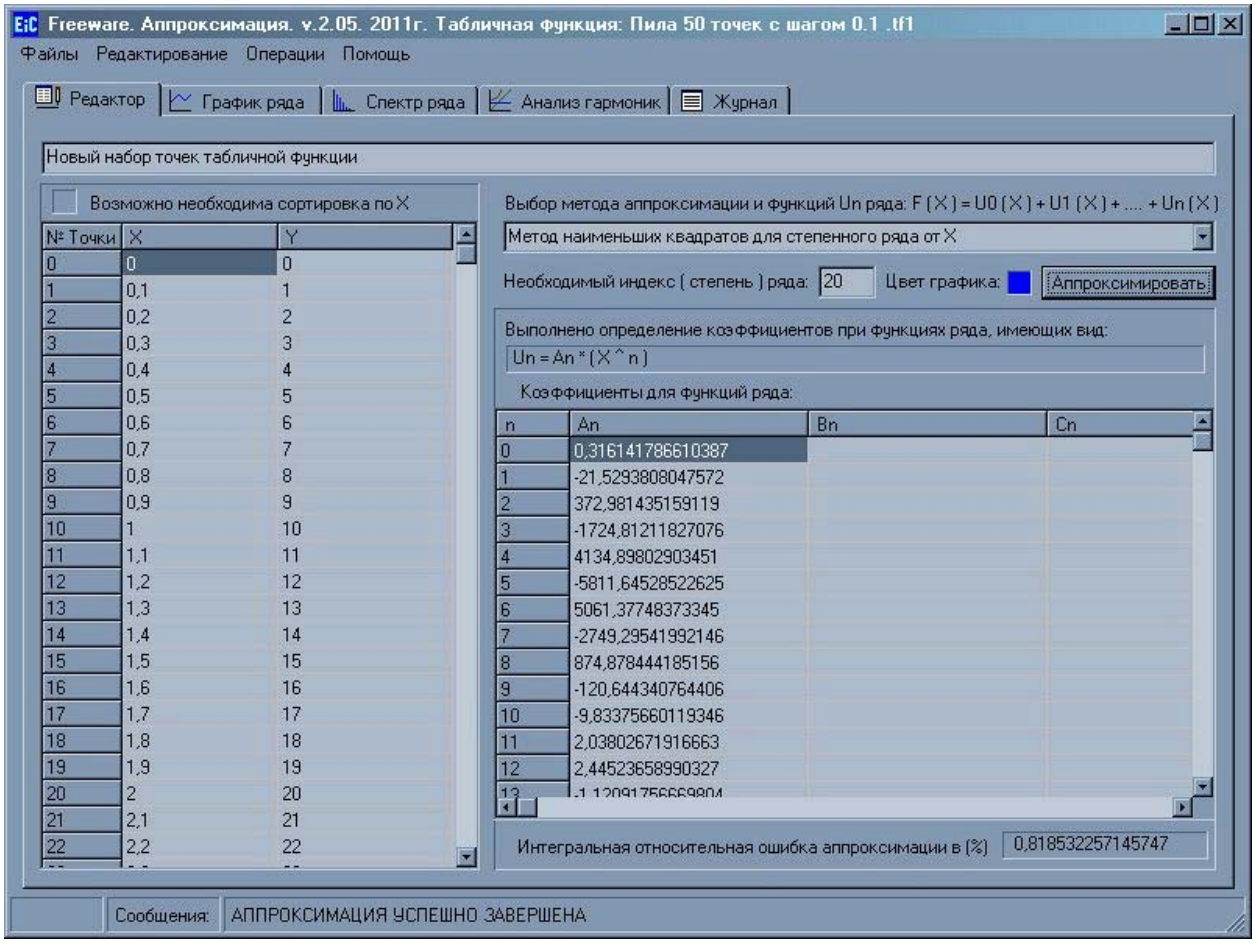
```
// ================================================================
//    ОСНОВНЫЕ ПРОЦЕДУРЫ МОДУЛЯ (Реализация для DELPHI 7)
// ================================================================
// Версия 2.03.
// 09.12.2009
// Подготовить матрицу системы линейных уравнений
// и выполнить нахождение коэффициентов аппроксимации
procedure PrepareAndExecuteGauss(RqSeries : word; var AproxCB : TAproxCB);
var  n    :  integer;  // Индекс для перебора точек табличной функции
var  i    :  integer;  // Индекс степени члена ряда в линейном уравнении
```

```
var  j    :  integer;  // Индекс степени члена свертки в линейном уравнении
var  Xn   :  double;   // Значение X в точке n
var  XnJ  :  double;   // Xn в степени свертки
var  Fn   :  double;   // Значение табличной функции в точке n
var  WRow :  integer;  // Рабочий номер строки системы линейных уравнений
var  WCol :  integer;  // Рабочий номер столбца системы линейных уравнений
var  m    : integer;   // Индекс для просмотра Matrix01.AI
begin
// ----------------------
FillChar(Matrix01, SizeOF(Matrix01), #0);
with Matrix01 do
begin
  MaxRow := AproxCB.ApExp + 1;
  MaxCol := AproxCB.ApExp + 2;
  for WRow := 1 to MaxRow do
  // Для каждой строки системы линейных уравнений
  begin
   j := WRow - 1;
   for n := Low(AproxCB.ptTabFun^) to High(AproxCB.ptTabFun^) do
   // Для каждой точки в таблично заданной функции
   begin
      // ---------------------------------
      // Выборка конкретного значения таблично заданной функции
      Xn := ReFuncArgument( RqSeries, AproxCB.ptTabFun^[n].X );
      Fn := AproxCB.ptTabFun^[n].Y;
      // ---------------------------------
      XnJ := Involve(Xn, j);  // Возвести X в степень J
      for WCol := 1 To MaxCol do
      begin
         i := WCol - 1;
         if WCol = MaxCol
         then begin
            // Формирование свободного члена WRow - линейного уравнения
            if (n = Low(AproxCB.ptTabFun^)) or (n= High(AproxCB.ptTabFun^))
            then TAB[WRow, WCol] := TAB[WRow, WCol] + (Fn * XnJ)/2
            else TAB[WRow, WCol] := TAB[WRow, WCol] +  Fn * XnJ;
         end
         else begin
            // Формирование коэфициентов при степенях
            // WRow - линейного уравнения
            if (n = Low(AproxCB.ptTabFun^)) or (n= High(AproxCB.ptTabFun^))
            then TAB[WRow, WCol] := TAB[WRow, WCol] + (Involve(Xn, i)* XnJ)/2
            else TAB[WRow, WCol] := TAB[WRow, WCol] +  Involve(Xn, i)* XnJ;
         end;
      end; // for WCol / i
   end; // for n
  end; // for WRow / j
end; // with Matrix01
// ----------------------------
// ВЫПОЛНИТЬ РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ
if ExecGaussEquationsSet (Matrix01) then
begin
   // Скопировать коэфициенты апроксимации в AproxCB
   with AproxCB do
   begin
     i := Low(ApAi);
     for m := 1 to (ApExp + 1) do
       begin
         // ----------------------------
         // Переход от Extended к Double
         // Ограничим запас на уровне 8 порядков
         if Abs(Matrix01.AI[m]) > 1.0E300
         then ApAi[i] := Sign(Matrix01.AI[m]) * 1.0E300
         else ApAi[i] := Matrix01.AI[m];
```

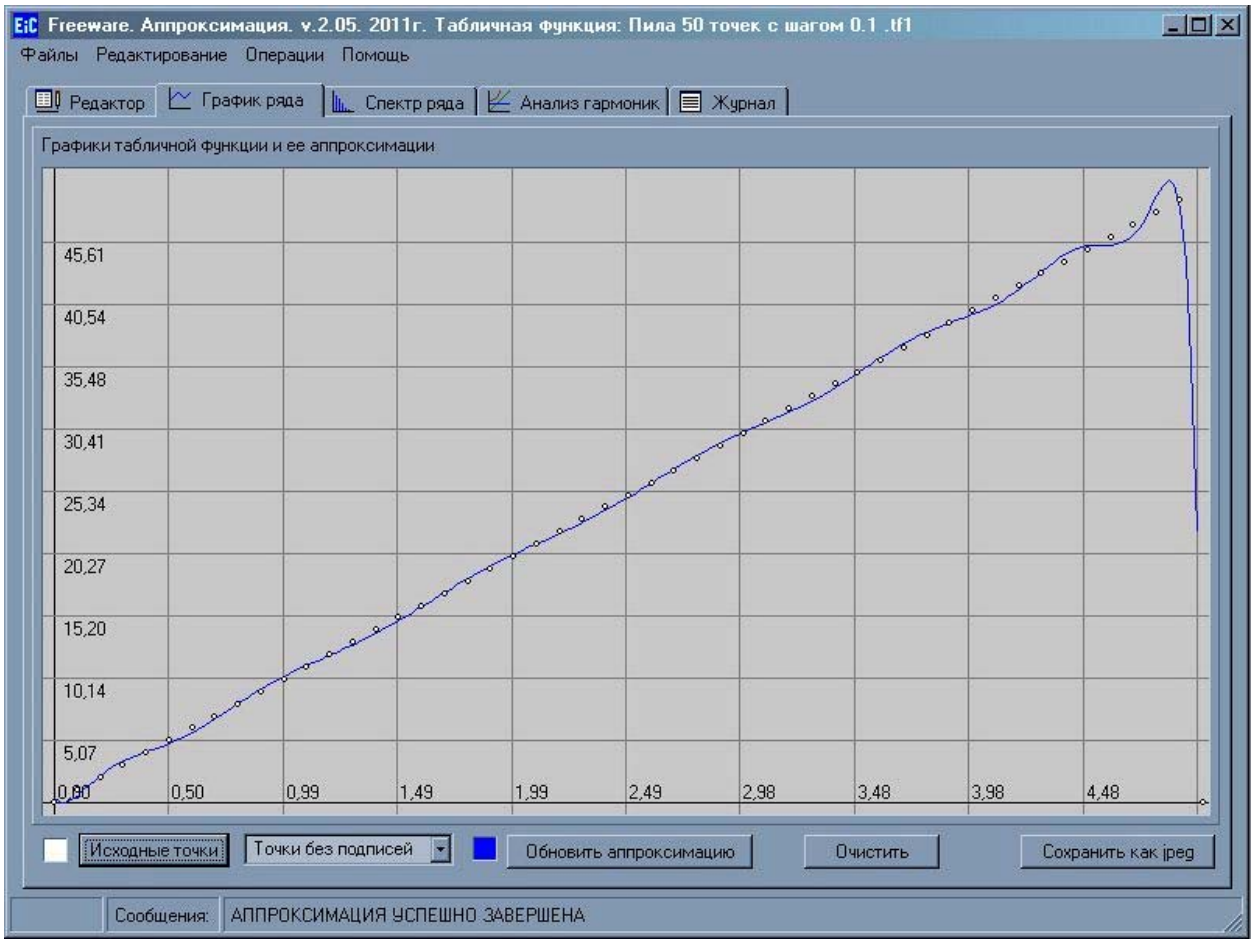
```
          //--------------------------------
          if Abs(Matrix01.AI[m]) < 1.0E-300
          then ApAi[i] := 0;
          //--------------------------------
          Inc(i);
        end;
     end; // with AproxCB
end
else begin
   with AproxCB do
   begin
     Abend := 3;
     ApCod := 130;
     ApMsg :=    ' Метод наименьших квадратов невыполним, так как'
      + #13#10 + ' характер исходных данных и вид функций аппроксимирующего'
      + #13#10 + ' ряда плохо совместимы. Выполнение апроксимации отменено.';
     SysErrorToReport (ApMsg);
   end;
end; // if
end; // procedure
```

Поскольку данная программа доступна как Open Source, причем исходные тексты детально прокомментированы, мы не будем углубляться в анализ этих текстов.

В завершение приведем несколько характерных экранных образов программы:



Панель отображения исходных точек табличной функции, а также различных аппроксимиаций исходной функции:



Список литературы, ссылки

1. Корн Г., Корн Т. Справочник по математике (для научных работников и инженеров). М.: Наука, -1978, - 832с.: ил.

«Справочник» содержит сведения по следующим разделам: высшая алгебра, аналитическая и дифференциальная геометрия, математический анализ (включая интегралы Лебега и Стилтьеса); векторный и тензорный анализ; криволинейные координаты; функции комплексного переменного; операционное исчисление; дифференциальные уравнения обыкновенные и с частными производными; вариационное исчисление; абстрактная алгебра; матрицы; линейные векторные пространства; операторы и теория представлений; интегральные уравнения; краевые задачи; теория вероятностей и математическая статистика; численные методы анализа; специальные функции.

2. Андре Анго. Математика для электро- и радиоинженеров. М.: Наука, - 1965, - 779с.: ил.

В книге рассматриваются много интересных приложений из области электро - и радиотехники будет интересна широкому кругу инженерно-технических и научных работников, имеющих дело с математикой и ее приложениями, а также студентам и аспирантам.

3. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и ее инженерные приложения. М.: Высшая школа, 2000, - 383с.: ил.

В книге дается систематическое изложение основ теории случайных процессов по специальностям: кибернетика, прикладная математика, автоматизированные системы управления и переработки информации, автоматизация технологических процессов, транспорт и т. п. Онаявляется логическим продолжением книги тех же авторов `Теория вероятностей и ее инженерные приложения`. Первое издание вышло в 1991 г. Для студентов высших технических учебных заведений. Книга может быть полезна преподавателям, инженерам и научным работникам разных профилей, которые в своей практической деятельности сталкиваются с необходимостью ставить и решать задачи, связанные с анализом случайных процессов.

4. Тиман А.Ф. Теория приближения функций действительного переменного. М.: Государственное издательство физико-математической литературы, -1960, - 624с.: ил.

В монографии излагается ряд основных разделов в современной теории приближения функций действительного переменного. Материал группируется вокруг проблемы связи наилучшего приближения функций с их структурными свойствами. Книга предназначается для аспирантов и студентов-математиков старших курсов; она представляет также интерес для научных работников в области теории функций.

5. К. Ю. Богачев. Практикум на ЭВМ. Методы приближения функций. М.: Московский гос. Университет. Учебное пособие. – 1998, - 129с.: ил.

Пособие содержит описание алгоритмов, предлагаемых к реализации на ЭВМ студентам механико-математического факультета МГУ, а также приводит необходимое теоритическое обоснование для соответствующих алгоритмов.

6. Ермистов В.В., Прозоров Д.Е. Практикум по основам метрологии: Учебное пособие.– Киров: Изд-во ВятГУ, 2004. - 114 с.

В пособии излагаются основные положения метрологии, теории погрешностей и обработки результатов измерений. Описаны методы измерений физических величин. Приведены характеристики аналоговых и цифровых средств измерений. Значительное внимание уделено цифровой обработке измерительной информации.

7. Метод Гаусса. http://ru.wikipedia.org/wiki/Метод_Гаусса

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений. Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе ступенчатого (или треугольного) вида, из которого последовательно, начиная с последних (по номеру) переменных, находятся все остальные переменные.

Киев 2011г.