

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ

Институт проблем моделирования в энергетике им.Г.Е.Пухова
Отделение гибридных моделирующих и управляющих систем в
энергетике

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

Национальный авиационный университет
Кафедра электротехники и светотехники

В.В.Васильев, Л.А.Симак

**МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ
ПРОЦЕССОВ И СИСТЕМ**

Аппроксимация сигналов с применением системы Mathematica®

Киев – 2007

УДК 681.3

Авторы: В.В.Васильев, Л.А.Симак

Рецензент: чл.-корр.НАН Украины, д.т.н., проф. С.Г.Таранов

Математическое и компьютерное моделирование процессов и систем: Аппроксимация сигналов с применением системы Mathematica®: Учебное пособие / В.В.Васильев, Л.А.Симак. – К.: НАН Украины, 2007. – 127 с.

ISBN 978-966-02-4385-9

Работа посвящена инженерным методам аналитической аппроксимации непрерывных и дискретизированных сигналов на основе метода наименьших квадратов. Иллюстративные примеры выполнены в программной среде системы Mathematica®. Для студентов и аспирантов, занимающихся математическим моделированием и его применениями в различных областях науки и техники.

The work deals with engineering methods of analytical approximation of continuous and sampled signals based on least-squares method. Artwork has been fulfilled in Mathematica® software environment. For students and post-graduate dealing with modeling and it's applications in various fields of research (science & technology).

ISBN 978-966-02-4385-9

© Отделение гибридных
моделирующих и управляющих
систем в энергетике Института
проблем моделирования в
энергетике им.Г.Е.Пухова НАН
Украины, 2007
© В.В.Васильев, Л.А.Симак, 2007

Содержание

ВВЕДЕНИЕ	4
1. АППРОКСИМАЦИЯ НЕПРЕРЫВНЫХ СИГНАЛОВ	6
2. АППРОКСИМАЦИЯ СИГНАЛОВ, ЗАДАННЫХ МАССИВАМИ (В ВИДЕ ТАБЛИЦ).....	22
3. АППРОКСИМАЦИЯ СИГНАЛОВ, ЗАДАННЫХ В ПАРАМЕТРИЧЕСКОЙ ФОРМЕ	31
4. ЛОКАЛЬНЫЕ БАЗИСНЫЕ СИСТЕМЫ	45
5. АППРОКСИМАЦИЯ ДВУМЕРНЫХ СИГНАЛОВ В ЛОКАЛЬНЫХ БАЗИСНЫХ СИСТЕМАХ СЕПАРАБЕЛЬНОГО ТИПА.....	62
6. УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	74
Приложение 1. Первоначальные сведения по системе Mathematica фирмы Wolfram Research Inc.	79
Приложение 2. Тексты программ иллюстративных примеров, рассмотренных в пособии.	85
Приложение 3. Генераторы локально-импульсных базисных функций на основе смещенных полиномов Лежандра	96
Приложение 4. Программы и решения к упражнениям.....	97
Список литературы	126

ВВЕДЕНИЕ

Проблема аналитической аппроксимации сигналов в математическом анализе и всевозможных технических приложениях возникает в следующих случаях:

- математическая модель сигнала не задана, а информация о сигнале получена с помощью аналоговых или цифровых регистраторов;
- информация о сигнале содержится в множестве отсчетов значений сигнала в некоторые моменты времени или в некоторых точках пространства (ситуация обычная для экспериментальных исследований);
- существующая математическая модель сигнала по какой-либо причине не устраивает исследователя и нужно аппроксимировать сигнал в иной форме.

В перечисленных случаях, однако, мы имеем дело всего с двумя формами представления сигналов: непрерывной (аналоговой) и дискретной (числовой массив или таблица). Наиболее часто при аппроксимации используется представление сигналов в виде обобщенных полиномов (линейной комбинации известных функций, образующих так называемую систему базисных функций). Полиномиальная аппроксимация сигналов при их последующей обработке предоставляет исследователю ряд возможностей, а именно:

- операции над неизвестными сигналами сводятся к аналогичным операциям над известными функциями, образующими базисную систему, с неизвестными коэффициентами,
- достигается сжатие информации, необходимость в которой возникает при регистрации и передаче сигналов на расстояние,
- появляется возможность низкочастотной фильтрации и сглаживания сигналов, содержащих шум, случайные помехи или высокочастотные наводки,
- аппроксимация сигналов может использоваться в системах шифрования сообщений, так как сообщение может быть восстановлено лишь в случае, когда известна система базисных функций, порождающая аппроксимирующий полином,
- при обработке экспериментальной информации с ограниченной выборкой отсчетов измеряемых сигналов появля-

ется возможность линейной и нелинейной интерполяции и экстраполяции,

- полиномиальная аппроксимация может служить основой для построения операционного исчисления, что особенно важно при исследовании динамических систем, так как любое операционное исчисление позволяет перейти от математической модели динамической системы в форме интегро-дифференциальных уравнений к эквивалентной модели в форме алгебраических уравнений.

Аппроксимация сигналов широко используется в математическом и компьютерном моделировании, построении генераторов сигналов заданной формы, идентификации параметров исследуемых систем и др.

Учитывая учебный характер данного пособия, авторы сочли возможным не перегружать его подробными выводами и доказательствами и постараться сохранить его инженерную направленность. Дополнительную информацию, связанную с теорией аппроксимации, можно получить из приведенной литературы [1, 4, 6, 8, 9 -13].

Приводимые в данном пособии примеры выполнены в среде системы Mathematica фирмы Wolfram Research Inc.[14, 15]. Для их повторения читателю потребуется первоначальное знакомство с указанной системой. Однако, рассмотренные методы аппроксимации могут быть легко реализованы и в других программных комплексах компьютерной алгебры (MATLAB, MAPPLE, MATHCAD и др.) [2, 3, 7, 16].

1. АППРОКСИМАЦИЯ НЕПРЕРЫВНЫХ СИГНАЛОВ

Пусть сигнал $x(t)$ задан на интервале $(a \leq t \leq b)$ изменения аргумента t . Предположим также, что на этом же интервале задана система линейно-независимых базисных функций $\{s_i(t)\}_{i=1}^m$,

формирующая обобщенный полином: $x_a = \sum_{i=1}^m X_i s_i(t)$. Для того,

чтобы этот полином аппроксимировал сигнал, коэффициенты полинома должны выбираться из условия минимизации нормы функции ошибки аппроксимации:

$$\varepsilon(t) = x(t) - x_a(t) = x(t) - \sum_{i=1}^m X_i s_i(t) \quad (1)$$

$$\|\varepsilon(t)\| \rightarrow \min(X_i)$$

Для среднеквадратичной нормы получим следующее условие:

$$\mu = \int_a^b \left[x(t) - \sum_{i=1}^m X_i s_i(t) \right]^2 dt \rightarrow \min(X_i). \quad (2)$$

Условие минимизации среднеквадратичной нормы функции ошибки приводит к системе линейных алгебраических уравнений относительно коэффициентов полинома:

$$\begin{aligned} \frac{\partial \mu}{\partial X_j} &= 0, \\ \sum_{i=1}^m w_{ij} X_i &= q_j, \\ j &:= 1, \dots, m, \end{aligned} \quad (3)$$

где:

$$\begin{aligned} w_{ij} &= \int_a^b s_i(t) s_j(t) dt, \\ q_j &= \int_a^b x(t) s_j(t) dt. \end{aligned} \quad (4)$$

Задача аппроксимации сигналов, заданная выражениями (3, 4) в матрично-векторной форме имеет вид:

$$W \cdot \bar{X} = \bar{Q},$$

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mm} \end{bmatrix}, \bar{Q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{bmatrix}, \bar{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix}. \quad (5)$$

В (5) приняты следующие обозначения:

W – операционная матрица аппроксимирующего полиномиального спектра,

\bar{Q} – операционный вектор спектра,

\bar{X} – аппроксимирующий полиномиальный спектр сигнала.

Решение задачи нахождения аппроксимирующего полиномиального спектра сигнала в матрично-векторной форме имеет вид:

$$\bar{X} = W^{-1} \cdot \bar{Q}. \quad (6)$$

В случае ортогональных базисных систем функций, матрица коэффициентов становится диагональной: $w_{ij, i \neq j} = 0$, а в случае ортонормированной базисной системы – единичной: $w_{ii} = 1$. Отсюда вытекает, что система линейных алгебраических уравнений (3) вырождается, и мы имеем: $X_i = q_i$.

Пример 1. Сформировать степенную систему базисных функций вида:

$$\bar{S}(t) = \left\{ \left(\frac{t}{T}\right)^0, \left(\frac{t}{T}\right)^{\frac{1}{4}}, \left(\frac{t}{T}\right)^{\frac{2}{4}}, \left(\frac{t}{T}\right)^{\frac{3}{4}}, \left(\frac{t}{T}\right)^2, \left(\frac{t}{T}\right)^3, \left(\frac{t}{T}\right)^4 \right\}$$

и определить для нее операционную матрицу аппроксимирующего полиномиального спектра W .

Ниже приведена программа и результаты ее работы в среде системы Mathematica. Примем значение константы $T=1$, а область определения системы функций $t \in [0,1)$.

Прежде всего определим вид базисной функции и сформируем из нее систему, определив ее параметры (константу $T=1$ и ее порядок $m=4$):

```
In[1]:= ss[t_, T_, i_, m_] :=  $\left(\frac{t}{T}\right)^{\text{If}[i/m < 1, \frac{i}{m}, i-m]}$ ;
```

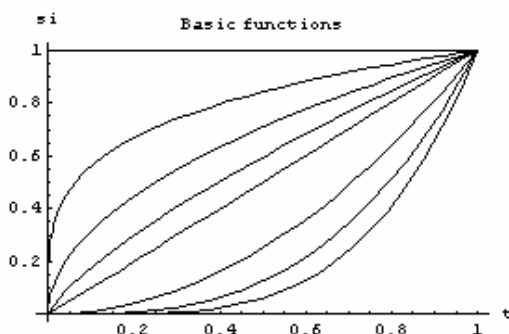
```
s = ss[t, 1, i, 4];
```

```
S = Table[s, {i, 8}]
```

```
Out[3]= {t1/4, √t, t3/4, 1, t, t2, t3, t4}
```

Отобразим на графике полученную систему базисных функций:

```
In[4]:= Plot[Evaluate[S], {t, 0.001, 1}, AxesLabel → {t, si},  
PlotLabel → Basic functions]
```



```
Out[4]= - Graphics -
```

Сформируем операционную матрицу аппроксимирующего полиномиального спектра в соответствии с выражениями (4-5):

```
In[5]:= W = ∫01 Table[S[[i]] * S[[j]], {i, 8}, {j, 8}] dt;
```

```
In[6]:= % // MatrixForm
```

```
Out[6]//MatrixForm=
```

$$\begin{pmatrix} \frac{2}{3} & \frac{4}{7} & \frac{1}{2} & \frac{4}{5} & \frac{4}{9} & \frac{4}{13} & \frac{4}{17} & \frac{4}{21} \\ \frac{4}{7} & \frac{1}{2} & \frac{4}{9} & \frac{2}{3} & \frac{2}{5} & \frac{2}{7} & \frac{2}{9} & \frac{2}{11} \\ \frac{1}{2} & \frac{4}{9} & \frac{2}{5} & \frac{4}{7} & \frac{4}{11} & \frac{4}{15} & \frac{4}{19} & \frac{4}{23} \\ \frac{4}{5} & \frac{2}{3} & \frac{4}{7} & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{4}{9} & \frac{2}{5} & \frac{4}{7} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{4}{13} & \frac{2}{7} & \frac{4}{15} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{4}{17} & \frac{2}{9} & \frac{4}{19} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{4}{21} & \frac{2}{11} & \frac{4}{23} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}$$

Пример 2. Аппроксимировать сигнал $x(t) = e^{-t}$ на интервале изменения аргумента $0 \leq t < 1$ в полученной системе степенных базисных функций. При повторении примера следует учитывать, что система базисных функций и операционная матрица спектра реализованы в соответствии с программой примера 1.

Фрагмент программы и результаты аппроксимации имеют вид (вектора Q и X в соответствии с выражениями (5-6):

```
In[7]:= Q = Integrate[Table[S[[i]] Exp[-t], {i, 1, 8}], {t, 0, 1}];
      Q1 = N[Q]
```

```
Out[8]= {0.476959, 0.378945, 0.312208, 0.632121,
      0.264241, 0.160603, 0.113929, 0.0878363}
```

```
In[9]:= X = Inverse[W].Q1
```

```
Out[9]= {-0.00344159, 0.0179984, -0.0397801, 1.00022,
      -0.967455, 0.481421, -0.142962, 0.0218797}
```

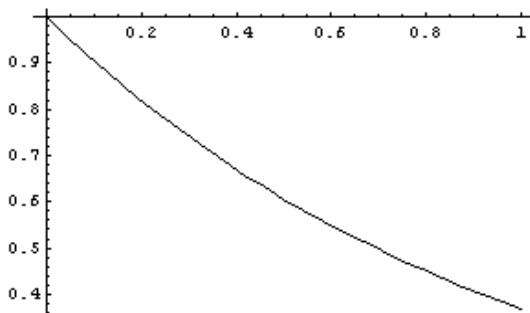
Формирование аппроксимирующего полинома:

```
In[32]:= xa = X.S
```

```
Out[32]= 1.00022 - 0.00344159 t1/4 + 0.0179984 √t - 0.0397801 t3/4 -
      0.967455 t + 0.481421 t2 - 0.142962 t3 + 0.0218797 t4
```

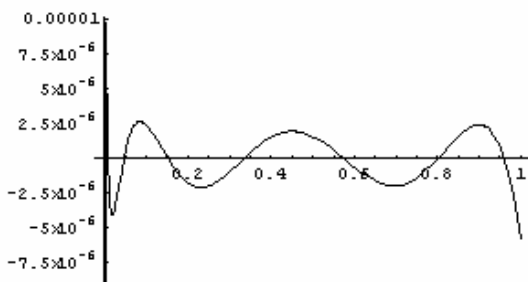
Вывод графика сигнала, совмещенного с его аппроксимацией, и графика функции ошибки аппроксимации:

```
In[11]:= Plot[{Exp[-t], xa}, {t, 0, 1}]
```



```
Out[11]= - Graphics -
```

```
In[12]:= Plot[Exp[-t] - xа, {t, 0, 1}]
```



```
Out[12]= - Graphics -
```

Пример 3. Аппроксимировать сигнал $y(t) = \text{Cos}(3.14t^2)$ на интервале изменения аргумента $0 \leq t < 1$ полиномом в той же системе степенных базисных функций.

Ниже приведены фрагменты программы в среде системы “Mathematica” и результаты ее работы. На графиках изображены сигнал, совмещенная с ним аппроксимация и функция ошибки. Команда **Chop** предусматривает обращение в нуль пренебрежимо малых составляющих в полученных результатах.

```
In[13]:= Q2 = Chop[Integrate[Table[S[[i]] Cos[3.14 t^2], {i, 1, 8}],  
{t, 0, 1}]]
```

```
Out[13]= {0.210864, 0.110369, 0.0447867, 0.374331,  
0.000253607, -0.0801576, -0.10117, -0.104289}
```

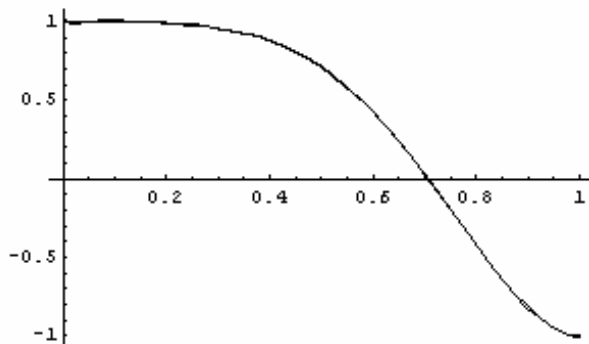
```
In[14]:= Y = Inverse[W].Q2
```

```
Out[14]= {10.5834, -54.7752, 119.576, 0.302353,  
-96.3983, 50.1747, -52.7558, 22.2821}
```

```
In[15]:= ya = Y.S
```

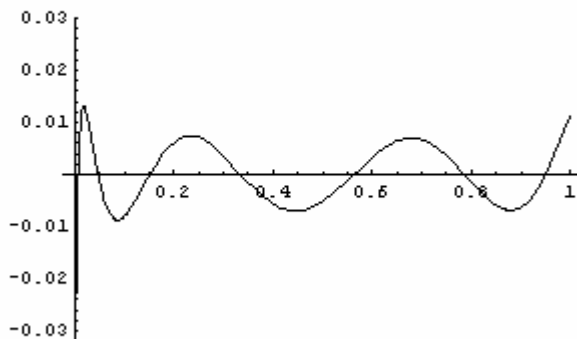
```
Out[15]= 0.302353 + 10.5834 t^(1/4) - 54.7752 sqrt[t] + 119.576 t^(3/4) -  
96.3983 t + 50.1747 t^2 - 52.7558 t^3 + 22.2821 t^4
```

```
In[16]:= Plot[{Cos[3.14 t^2], ya}, {t, 0, 1}]
```



```
Out[16]= - Graphics -
```

```
In[17]:= Plot[Cos[3.14 t^2] - ya, {t, 0, 1}]
```



```
Out[17]= - Graphics -
```

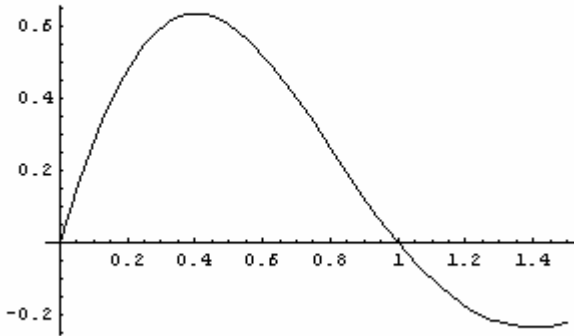
Пример 4. Аппроксимировать сигнал $e^{-t} \sin(\pi t)$ на интервале изменения аргумента $0 \leq t < \pi/2$, используя экспоненциальную систему базисных функций 5-го порядка: $\vec{S}(t) = \{1, e^{-t}, e^{-2t}, e^{-3t}, e^{-4t}\}$.

Ниже приведена программа на языке системы Mathematica и результат ее работы.

Определение сигнала и отображение его на графике:

```
In[1]:= f := Exp[-t] * Sin[Pi * t];
```

```
In[2]:= Plot[f, {t, 0, 1.5}]
```



```
Out[2]= - Graphics -
```

Определение системы базисных экспоненциальных функций, операционной матрицы спектра сигнала, нахождение аппроксимирующего полиномиального спектра и вывод вида системы базисных функций:

```
In[3]:= ss[t_, i_, m_] := Table[Exp[-t*(i-1)], {i, m}];
```

```
s1=ss[t, i, 5];
```

```
s2=ss[t, j, 5];
```

```
W=N[Table[Integrate[s1[[i]]*s2[[j]], {t, 0, Pi/2}], {i, 5}, {j, 5}]];
```

```
Q:=Chop[N[Table[Integrate[f*s1[[i]], {t, 0, Pi/2}], {i, 5}]]]
```

```
;
```

```
F:=Inverse[W].Q;
```

```
s1
```

```
Out[9]= {1, e^{-t}, e^{-2 t}, e^{-3 t}, e^{-4 t}}
```

Вывод операционной матрицы аппроксимирующего экспоненциального спектра, операционного вектора спектра, вектора коэффициентов аппроксимирующего полинома и формирование аппроксимирующего полинома:

In[11]:= **W//MatrixForm**

Out[11]//MatrixForm=

$$\begin{pmatrix} 1.5708 & 0.79212 & 0.478393 & 0.330339 & 0.249533 \\ 0.79212 & 0.478393 & 0.330339 & 0.249533 & 0.199922 \\ 0.478393 & 0.330339 & 0.249533 & 0.199922 & 0.166653 \\ 0.330339 & 0.249533 & 0.199922 & 0.166653 & 0.142855 \\ 0.249533 & 0.199922 & 0.166653 & 0.142855 & 0.125 \end{pmatrix}$$

In[12]:= **Q // MatrixForm**

Out[12]//MatrixForm=

$$\begin{pmatrix} 0.294426 \\ 0.230428 \\ 0.167553 \\ 0.121671 \\ 0.090142 \end{pmatrix}$$

In[13]:= **F // MatrixForm**

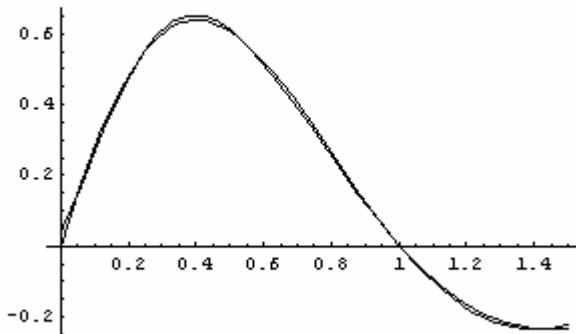
Out[13]//MatrixForm=

$$\begin{pmatrix} 1.47015 \\ -17.6018 \\ 58.6928 \\ -68.5671 \\ 26.0487 \end{pmatrix}$$

In[14]:= **fa = F.s1**

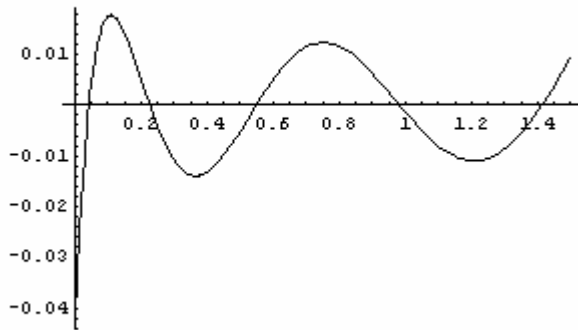
Out[14]= $1.47015 + 26.0487 e^{-4t} - 68.5671 e^{-3t} + 58.6928 e^{-2t} - 17.6018 e^{-t}$

In[15]:= **Plot[{f, fa}, {t, 0, 1.5}]**



Out[15]= - Graphics -

```
In[16]:= Plot[f - fa, {t, 0, 1.5}]
```



```
Out[16]= - Graphics -
```

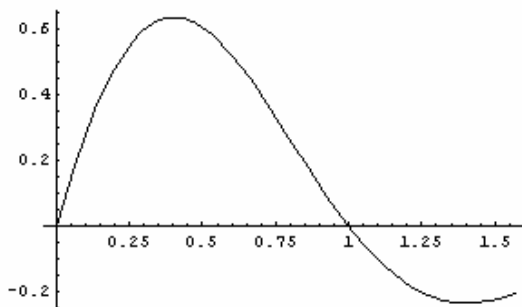
Пример 5. Аппроксимировать сигнал $e^{-t} \sin(\pi t)$ на интервале изменения аргумента $0 \leq t < \pi/2$, используя в качестве системы базисных функций тригонометрический полином:

$$\bar{S}(t) = \{1, \cos(4t), \cos(8t), \cos(12t), \cos(16t), \sin(4t), \sin(8t), \sin(12t), \sin(16t)\}$$

Ниже приведена программа на языке системы Mathematica и результаты ее работы.

Определение сигнала и отображение его на графике:

```
In[1]:= f := Exp[-t] * Sin[Pi * t];
Plot[f, {t, 0, Pi / 2}]
```



```
Out[2]= - Graphics -
```

Задание системы базисных функций:

```
ss[t_, i_, m_] :=
  Table[If[i ≤ m, Cos[4 t * (i - 1)],
    Sin[4 t * (i - m)]], {i, 2 m - 1}];
s1 = ss[t, i, 5];
s2 = ss[t, j, 5];
```

Определение операционной матрицы аппроксимирующего полиномиального спектра, нахождение операционного вектора спектра, вектора коэффициентов аппроксимирующего полинома и формирование полинома:

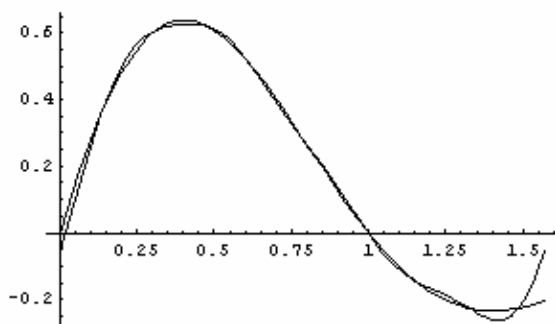
```
In[5]:= W =
  Chop[N[Table[Integrate[s1[[i]] * s2[[j]],
    {t, 0, Pi / 2}], {i, 9}, {j, 9}]]];
```

```
In[6]:= Q :=
  Chop[N[Table[Integrate[f * s1[[i]],
    {t, 0, Pi / 2}], {i, 9}]]];
F :=
  Inverse[W].Q;
```

```
In[8]:= fa = F.s1;
```

Отображение на графике сигнала и его аппроксимации:

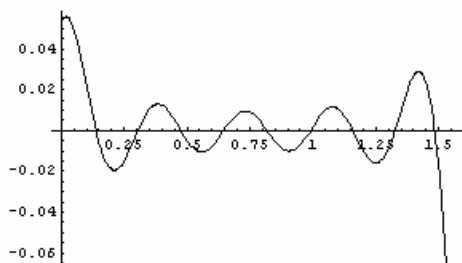
```
In[9]:= Plot[{f, fa}, {t, 0, Pi / 2}]
```



```
Out[9]= - Graphics -
```

Отображение на графике функции ошибки аппроксимации:

```
In[10]:= Plot[f - fa, {t, 0, Pi / 2}]
```



```
Out[10]= - Graphics -
```

Пример 6. Аппроксимировать сигнал $e^{\frac{(x/90)^2}{2}} - e^{-5\frac{(x/90)^2}{2}}$ на интервале изменения аргумента $-90 \leq x \leq 90$, используя в качестве системы базисных функций систему ортогональных полиномов Лежандра [1, 6, 15].

Определение типа базисной функции и формирование на ее основе базисной системы порядка 16:

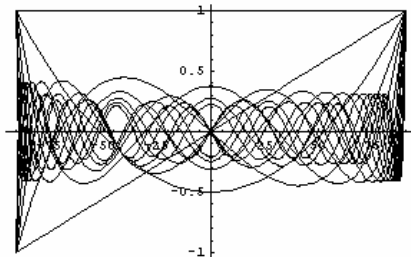
```
In[1]:= SL[x_] := Evaluate[Table[LegendreP[n, x], {n, 0, 15}]];
```

```
In[2]:= s1 = Simplify[SL[x / 90]];
```

Так как классическая система полиномов Лежандра является ортогональной на интервале изменения аргумента $(-1 < x < 1)$, то для сохранения ортогональности базисной системы функций командой *In[2]* выполнено масштабирование аргумента $x/90$.

Отображение на графике системы базисных функций:

```
In[3]:= Plot[Evaluate[s1], {x, -90, 90}]
```



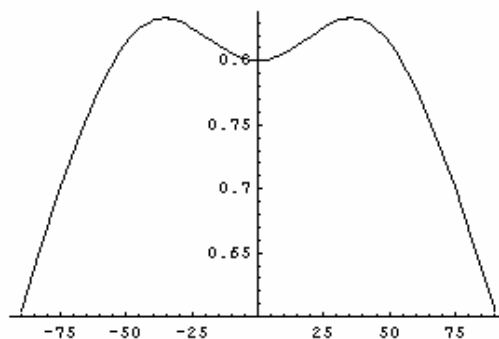
```
Out[3]= - Graphics -
```


Определение сигнала, подлежащего аппроксимации системой полиномов Лежандра:

```
In[4]:= f = Exp[-0.5 (x / 90) ^ 2] - 0.2 Exp[-5 (x / 90) ^ 2];
```

Отображение сигнала на интервале аппроксимации:

```
In[5]:= Plot[f, {x, -90, 90}, PlotRange -> All]
```



```
Out[5]= - Graphics -
```

Формирование операционной матрицы спектра:

```
In[6]:= W := Table[Integrate[s1[[i]] * s1[[j]], {x, -90, 90}],  
                  {i, 16}, {j, 16}];
```

Отображение фрагмента матрицы (с целью иллюстрации порядок матрицы уменьшен до 5):

```
In[8]:= W1
```

```
Out[8]/MatrixForm=
```

$$\begin{pmatrix} 180 & 0 & 0 & 0 & 0 \\ 0 & 60 & 0 & 0 & 0 \\ 0 & 0 & 36 & 0 & 0 \\ 0 & 0 & 0 & \frac{180}{7} & 0 \\ 0 & 0 & 0 & 0 & 20 \end{pmatrix}$$

Определение операционного вектора спектра Q:

```
In[9]:= Q := Table[Integrate[s1[[i]] * f, {x, -90, 90}], {i, 16}];
```

```
In[10]:= Q
```

```
Out[10]= {139.767, 0., -4.72853, 0., -1.36532,  
          0., 0.532145, 0., -0.136272, 0., 0.0288569,  
          0., -0.00526394, 0., 0.000838203, 0.}
```

Нахождение аппроксимирующего спектра Лежандра для заданного сигнала:

```
In[11]:= F = Inverse[W].Q;
```

```
In[12]:= F
```

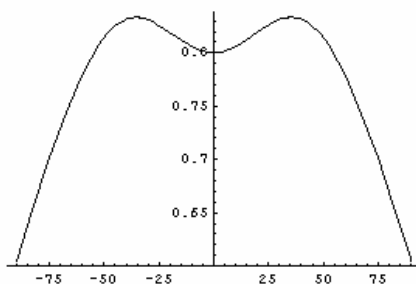
```
Out[12]= {0.776482, 0., -0.131348, 0., -0.0682659,
          0., 0.0384327, 0., -0.0128701, 0., 0.00336664,
          0., -0.000731103, 0., 0.000135044, 0.}
```

Формирование аппроксимирующего полинома:

```
In[13]:= fa = F.sl;
```

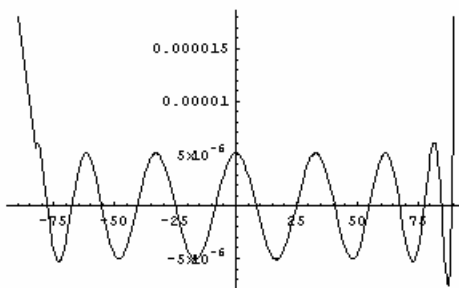
Отображение аппроксимации сигнала и ошибки аппроксимации:

```
In[14]:= Plot[fa, {x, -90, 90}, PlotRange -> All]
```



```
Out[14]= - Graphics -
```

```
In[15]:= Plot[fa - f, {x, -90, 90}, PlotRange -> All]
```



```
Out[15]= - Graphics -
```

Анализ результатов аппроксимации показывает, что в связи с тем, что аппроксимируемый сигнал является четным, в полученной

системе полиномов Лежандра отсутствуют нечетные компоненты, что позволяет уменьшить вдвое порядок базисной системы функций.

Повторим процесс аппроксимации, изменив систему базисных функций.

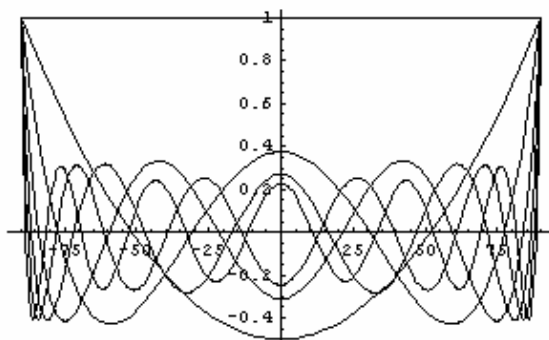
Задание системы базисных функций в виде системы полиномов Лежандра 7 порядка:

```
In[16]:= SL2[x_] := Evaluate[Table[LegendreP[n, x], {n, 0, 12, 2}]];
```

```
In[17]:= s12 = Simplify[SL2[x / 90]];
```

Отображение на графике полученной системы базисных функций:

```
In[18]:= Plot[Evaluate[s12], {x, -90, 90}]
```



```
Out[18]= - Graphics -
```

Формирование операционной матрицы спектра Лежандра:

```
In[19]:= W2 := Table[Integrate[s12[[i]] * s12[[j]], {x, -90, 90}],  
  {i, 7}, {j, 7}];
```

```
In[20]:= W2 // MatrixForm
```

```
Out[20]//MatrixForm=
```

$$\begin{pmatrix} 180 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 36 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{180}{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{180}{17} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{60}{7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{36}{5} \end{pmatrix}$$

Формирование операционного вектора спектра сигнала:

```
In[21]:= Q2 := Table[Integrate[s12[i]] * f, {x, -90, 90}], {i, 7}];
```

```
In[22]:= Q2
```

```
Out[22]:= {139.767, -4.72853, -1.36532,  
0.532145, -0.136272, 0.0288569, -0.00526394}
```

Нахождение аппроксимирующего спектра Лежандра:

```
In[23]:= F2 = Inverse[W2].Q2;
```

```
In[24]:= F2
```

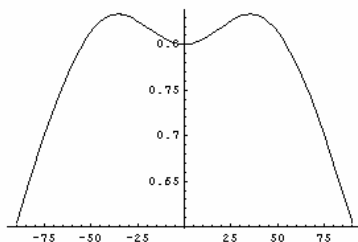
```
Out[24]:= {0.776482, -0.131348, -0.0682659,  
0.0384327, -0.0128701, 0.00336664, -0.000731103}
```

Формирование аппроксимирующего полинома для сигнала:

```
In[25]:= fa2 = F2.s12;
```

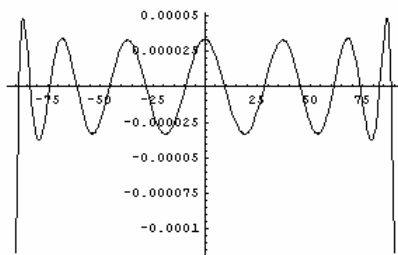
Отображение на графике аппроксимации сигнала и функции ошибки аппроксимации:

```
In[26]:= Plot[fa2, {x, -90, 90}, PlotRange -> All]
```



```
Out[26]:= - Graphics -
```

```
In[27]:= Plot[fa2 - f, {x, -90, 90}, PlotRange -> All]
```



```
Out[27]:= - Graphics -
```

Сравнение коэффициентов аппроксимирующих полиномов показывает, что ненулевые компоненты не изменились:

In[30]:= **F**

Out[30]= {0.776482, 0., -0.131348, 0., -0.0682659,
0., 0.0384327, 0., -0.0128701, 0., 0.00336664,
0., -0.000731103, 0., 0.000135044, 0.}

In[31]:= **F2**

Out[31]= {0.776482, -0.131348, -0.0682659,
0.0384327, -0.0128701, 0.00336664, -0.000731103}

2. АППРОКСИМАЦИЯ СИГНАЛОВ, ЗАДАННЫХ МАССИВАМИ (В ВИДЕ ТАБЛИЦ)

При измерениях величин, характеризующих исследуемый процесс, при обработке результатов экспериментальных исследований и в ряде других случаев часто приходится иметь дело с множеством отсчетов сигналов, относящихся к различным моментам времени или различным значениям пространственных координат. Таким образом, сигнал может быть задан в форме следующей таблицы значений:

Таблица 1

k	1	2	...	r
t_k	t_1	t_2	...	t_r
x_k	x_1	x_2	...	x_r

Поставим задачу аппроксимировать заданный сигнал обобщенным полиномом порядка m в системе базисных функций $\{s_i(t)\}_{i=1}^m$:

$$x = \sum_{i=1}^m X_i s_i(t). \quad (7)$$

Подставим в выражение аппроксимирующего полинома (7) значения отсчетов из таблицы:

$$x_k = \sum_{i=1}^m X_i s_i(t_k), \quad k := 1, \dots, r. \quad (8)$$

Последнее выражение может рассматриваться как система линейных алгебраических уравнений относительно искоемых коэффициентов аппроксимирующего полинома. Матрица коэффициентов этой системы уравнений определяется выражением:

$$w_{ki} = s_i(t_k). \quad (9)$$

Так как количество отсчетов сигнала и порядок обобщенного полинома, в общем случае, не равны, операционная матрица спектра получается прямоугольной, и решение этой системы уравнений должно получаться с помощью псевдообратной матрицы. Как известно [5], если порядок аппроксимирующего полинома меньше числа отсчетов сигнала ($m < r$), система уравнений является переопределенной, и решение, полученное с помощью

псевдообратной матрицы будет обращать в минимум сумму квадратов ошибок в уравнениях, получаемых при подстановки в них значений отсчетов. Если порядок аппроксимирующего полинома равен числу отсчетов сигнала ($m = r$), решение системы уравнений будет определять систему коэффициентов единственного аппроксимирующего полинома, проходящего через все точки отсчетов. Если порядок аппроксимирующего полинома больше числа отсчетов ($m > r$), система уравнений будет недоопределенной, однако, решение с помощью псевдообратной матрицы будет определять систему коэффициентов аппроксимирующего полинома, для которой минимальна сумма квадратов этих коэффициентов, а сам аппроксимирующий полином будет проходить через все точки отсчетов сигнала.

Рассмотрим иллюстративные примеры аппроксимации сигналов, заданных таблично.

Пример 7. Аппроксимировать сигнал, заданный значениями функции $x(t) = \cos(t^2) + 0.1\sin(7t)$ в 10 равноотстоящих точках интервала $0 \leq t < T = 1$, с помощью степенных полиномов различных порядков. Программа аппроксимации с комментариями в среде системы Mathematica и результаты ее работы приводятся ниже.

Задаем систему 10 равноотстоящих моментов времени на единичном интервале изменения аргумента:

```
In[1]:= h = 0.1; r = 10;
```

```
tt := Table[i * h, {i, 0, r - 1}]
```

```
In[3]:= tt
```

```
Out[3]= {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
```

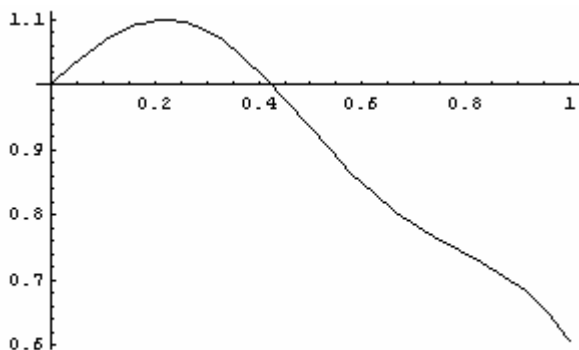
Сформируем сигнал:

```
In[4]:= T = 1;
```

```
x := Cos[(tt/T)^2] + 0.1 Sin[7 tt/T];
```

и отобразим его на графике:

```
In[6]:= p1 = Plot[x, {t, 0, T}, PlotRange -> All]
```



Out[6]= - Graphics -

Сформируем массив отсчетов сигнала в заданные выше моменты времени:

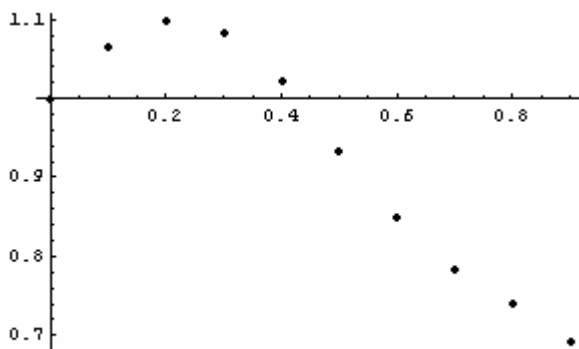
```
In[7]:= xt := Table[x /. t -> tt[[i]], {i, 1, r}]
```

```
In[8]:= xt
```

```
Out[8]= {1, 1.06437, 1.09775, 1.08227, 1.02073,
          0.933834, 0.848739, 0.784088, 0.738969, 0.69118}
```

Отообразим полученный массив отсчетов:

```
p2 = ListPlot[Table[{tt[[i]], xt[[i]]}, {i, 1, r}],
  PlotStyle -> PointSize[0.015]]
```



Out[9]= - Graphics -

Сформируем систему степенных базисных функций и выберем ее порядок равный 5:

```
In[10]:= m = 5; k = 10;

s[t_, T_, i_] := (t/T)^(i-1);

si = s[t, 1, i];

S := Table[si, {i, 1, m}]
```

```
In[14]:= S
```

```
Out[14]= {1, t, t^2, t^3, t^4}
```

Сформируем операционную матрицу аппроксимирующего степенного спектра для этой системы отсчетов:

```
In[15]:= w := Table[si /. t -> tt[[j]], {j, 1, k}, {i, 1, m}];

In[16]:= w // MatrixForm
```

```
Out[16]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0.1 & 0.01 & 0.001 & 0.0001 \\ 1 & 0.2 & 0.04 & 0.008 & 0.0016 \\ 1 & 0.3 & 0.09 & 0.027 & 0.0081 \\ 1 & 0.4 & 0.16 & 0.064 & 0.0256 \\ 1 & 0.5 & 0.25 & 0.125 & 0.0625 \\ 1 & 0.6 & 0.36 & 0.216 & 0.1296 \\ 1 & 0.7 & 0.49 & 0.343 & 0.2401 \\ 1 & 0.8 & 0.64 & 0.512 & 0.4096 \\ 1 & 0.9 & 0.81 & 0.729 & 0.6561 \end{pmatrix}$$

Определим операционный вектор спектра:

```
In[17]:= Q = xt
```

```
Out[17]= {1, 1.06437, 1.09775, 1.08227, 1.02073,
          0.933834, 0.848739, 0.784088, 0.738969, 0.69118}
```

Определим вектор коэффициентов аппроксимирующего полинома и построим аппроксимирующий полином:

```
In[18]:= X = PseudoInverse[w].Q
```

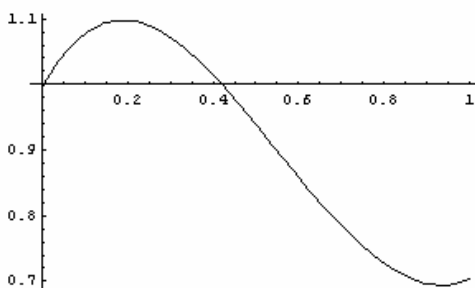
```
Out[18]:= {0.994651, 1.18001, -3.86829, 2.75786, -0.361736}
```

```
In[19]:= xa = X.S
```

```
Out[19]:= 0.994651 + 1.18001 t - 3.86829 t2 + 2.75786 t3 - 0.361736 t4
```

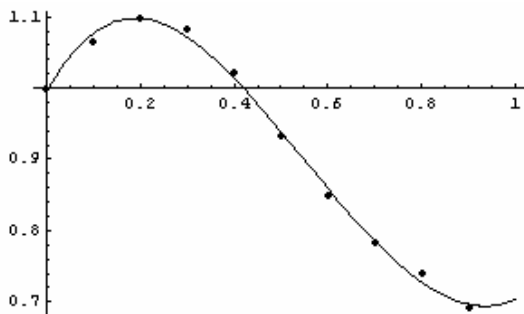
Построим график аппроксимирующего полинома:

```
In[20]:= p3 = Plot[xa, {t, 0, T}];
```



Совместим этот график с картиной точек отсчета сигнала:

```
In[21]:= Show[p2, p3]
```



```
Out[21]:= - Graphics -
```

Уменьшим порядок аппроксимирующего полинома до $m = 3$.
Результаты работы программы приводятся ниже.

```
In[22]:= S1 := Table[s[t, T, i], {i, 3}]
```

```
In[23]:= w1 := Table[si /. t -> tt[[j]], {j, k}, {i, 3}];
```

Out[24]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0.1 & 0.01 \\ 1 & 0.2 & 0.04 \\ 1 & 0.3 & 0.09 \\ 1 & 0.4 & 0.16 \\ 1 & 0.5 & 0.25 \\ 1 & 0.6 & 0.36 \\ 1 & 0.7 & 0.49 \\ 1 & 0.8 & 0.64 \\ 1 & 0.9 & 0.81 \end{pmatrix}$$

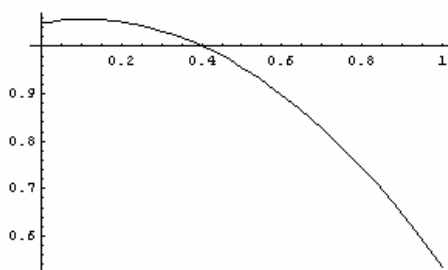
In[25]:= **X1 = PseudoInverse[w1].Q**

Out[25]= {1.0493, 0.143689, -0.658845}

In[26]:= **xa1 = X1.S1**

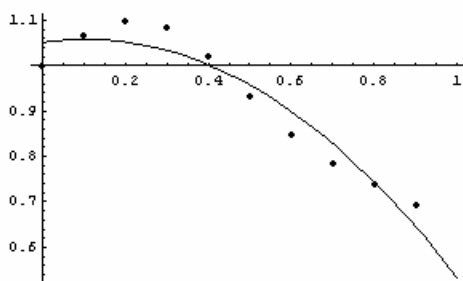
Out[26]= 1.0493 + 0.143689 t - 0.658845 t²

In[27]:= **p4 = Plot[xa1, {t, 0, T}]**



Out[27]= - Graphics -

In[28]:= **Show[p4, p2]**



Out[28]= - Graphics -

Ограничимся теперь выборкой из 5 отсчетов сигнала. Для полинома 5 степени получим следующую картину (шаг по времени увеличен вдвое на том же интервале изменения аргумента), программа и результаты ее работы приводятся без комментариев.

```
In[30]:= h1 = 0.2; r1 = 5;
```

```
tt1 := Table[i * h1, {i, 0, r1 - 1}]
```

```
In[32]:= tt1
```

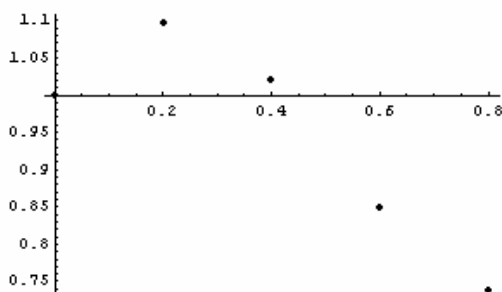
```
Out[32]:= {0, 0.2, 0.4, 0.6, 0.8}
```

```
In[33]:= xt1 := Table[x /. t -> tt1[[i]], {i, 1, r1}]
```

```
In[34]:= xt1
```

```
Out[34]:= {1, 1.09775, 1.02073, 0.848739, 0.738969}
```

```
In[35]:= p5 = ListPlot[Table[{tt1[[i]], xt1[[i]]}, {i, 1, r1}],  
PlotStyle -> PointSize[0.017]]
```



```
Out[35]:= - Graphics -
```

```
k = 5;
```

```
In[37]:= w1 := Table[si /. t -> tt1[[j]], {j, 1, k}, {i, 1, m}];
```

```
In[38]:= w1 // MatrixForm
```

```
Out[38]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0.2 & 0.04 & 0.008 & 0.0016 \\ 1 & 0.4 & 0.16 & 0.064 & 0.0256 \\ 1 & 0.6 & 0.36 & 0.216 & 0.1296 \\ 1 & 0.8 & 0.64 & 0.512 & 0.4096 \end{pmatrix}$$

```
In[39]:= Q1 = xt1
```

```
Out[39]:= {1, 1.09775, 1.02073, 0.848739, 0.738969}
```

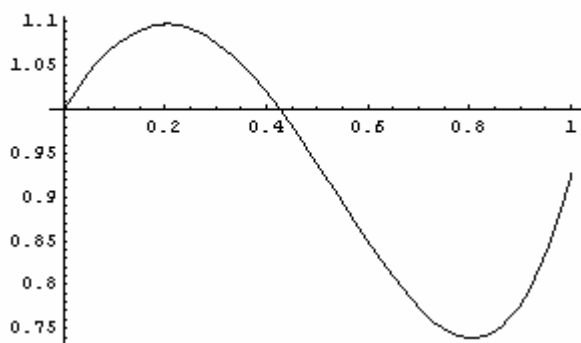
```
In[40]:= X1 = PseudoInverse[w1].Q1
```

```
Out[40]:= {1., 0.961894, -2.29526, -0.755966, 2.01532}
```

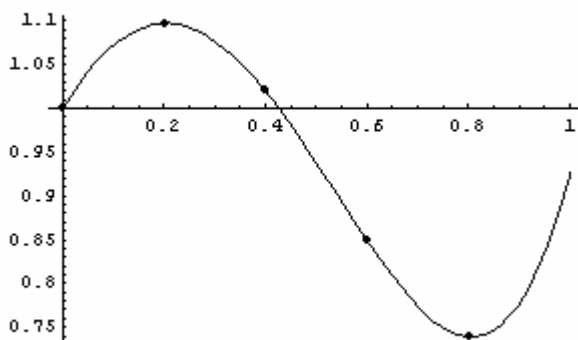
```
In[41]:= xa1 = X1.S
```

```
Out[41]:= 1. + 0.961894 t - 2.29526 t2 - 0.755966 t3 + 2.01532 t4
```

```
In[42]:= p6 = Plot[xa1, {t, 0, T}];
```



```
In[43]:= p7 = Show[p5, p6]
```



```
Out[43]= - Graphics -
```

Если на той же сокращенной выборке использовать полином 7 порядка, получим следующую картину:

```
In[44]:= m1 = 7;
```

```
In[45]:= w2 := Table[si /. t -> tt1[[j]], {j, 1, k}, {i, 1, m1}];
```

```
In[46]:= w2 // MatrixForm
```

```
Out[46]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.2 & 0.04 & 0.008 & 0.0016 & 0.00032 & 0.000064 \\ 1 & 0.4 & 0.16 & 0.064 & 0.0256 & 0.01024 & 0.004096 \\ 1 & 0.6 & 0.36 & 0.216 & 0.1296 & 0.07776 & 0.046656 \\ 1 & 0.8 & 0.64 & 0.512 & 0.4096 & 0.32768 & 0.262144 \end{pmatrix}$$

```
In[47]:= X2 = PseudoInverse[w2].Q1
```

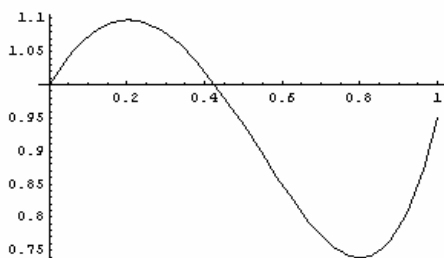
```
Out[47]:= {1., 0.984985, -2.53474, 0.0748656, 0.851311, 0.546091, 0.0276269}
```

```
In[48]:= S2 := Table[si, {i, 1, m1}]
```

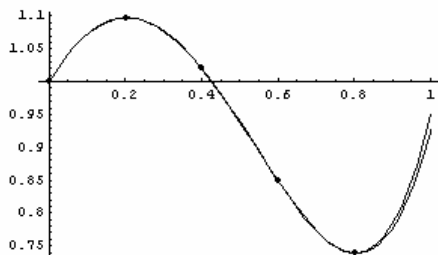
```
In[49]:= xa2 = X2.S2
```

```
Out[49]:= 1. + 0.984985 t - 2.53474 t2 +  
0.0748656 t3 + 0.851311 t4 + 0.546091 t5 + 0.0276269 t6
```

```
In[50]:= p8 = Plot[xa2, {t, 0, T}];
```



```
In[51]:= p9 = Show[p7, p8]
```



```
Out[51]= - Graphics -
```

Сравнение двух последних графиков показывает, что качество аппроксимации практически не изменилось, несмотря на увеличение порядка аппроксимирующего полинома.

3. АППРОКСИМАЦИЯ СИГНАЛОВ, ЗАДАННЫХ В ПАРАМЕТРИЧЕСКОЙ ФОРМЕ

При статистической обработке экспериментальных данных, параметрической идентификации процессов и систем и ряде других применений, связанных с цифровой обработкой сигналов, часто сигналы заданы в параметрической форме. Примерами могут служить фазовые портреты систем автоматического управления, диаграммы направленности антенных систем, диаграммы распределения силы света источников светового излучения и т.п. Основные соотношения для параметрических составляющих сигнала в случае непрерывных сигналов имеют вид (10)

$$\begin{aligned} y(x), \\ y = \varphi(t), \\ x = \psi(t), \\ t = \psi^{-1}(x), \\ y = \varphi(\psi^{-1}(x)). \end{aligned} \quad (10)$$

При дискретизации подобных сигналов в форме отсчетов возникают многомерные таблицы (Таблица 2).

Таблица 2

№ отсчета	k	1	2	...	r
Параметрический аргумент	t_k	t_1	t_2	...	t_r
Координата 1	x_k	x_1	x_2	...	x_r
Координата 2	y_k	y_1	y_2	...	y_r

Значения параметрического аргумента могут быть не заданы. Если параметрические составляющие сигнала заданы в явном виде, аппроксимация может быть выполнена описанным выше способом. В том случае, когда значения параметрического аргумента неизвестны, они приближенно могут быть оценены путем использования формул перехода от декартовой к полярным координатам:

$$t_k = \text{Arc cos} \frac{x_k}{\sqrt{x_k^2 + y_k^2}} = \text{Arc sin} \frac{y_k}{\sqrt{x_k^2 + y_k^2}} \quad (11)$$

Рассмотрим примеры аппроксимации сигналов, заданных в параметрической форме.

Пример 8. Аппроксимировать сигнал, заданный своими параметрическими составляющими:

$$y(t) = e^{-t} \sin(2\pi t),$$

$$x(t) = e^{-2t} \cos(2\pi t),$$

на интервале изменения аргумента $0 \leq t < 2$, используя степенную систему базисных функций 7 порядка.

Ниже приводится программа решения аппроксимационной задачи с комментариями и результаты ее работы.

Задание параметрических составляющих сигнала:

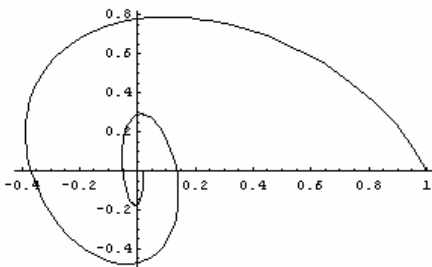
```
In[1]:= y[t_, ω_] := Exp[-t] * Sin[ω * t];
```

```
In[2]:= x[t_, ω_] := Exp[-2 t] * Cos[ω * t];
```

```
In[3]:= y1 = y[t, 2 π]; x1 = x[t, 2 π];
```

Отображение на графиках сигнала и его параметрических составляющих:

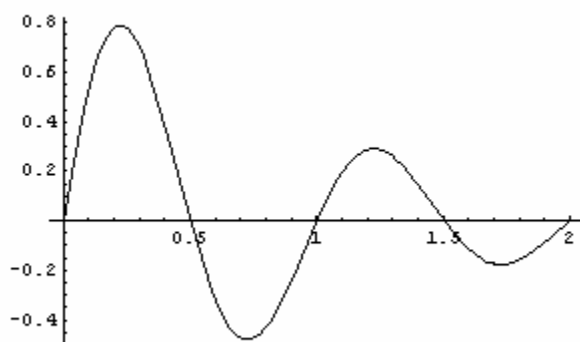
```
In[4]:= PP = ParametricPlot[{x1, y1}, {t, 0, 2}, PlotRange -> All]
```



```
Out[4]= - Graphics -
```

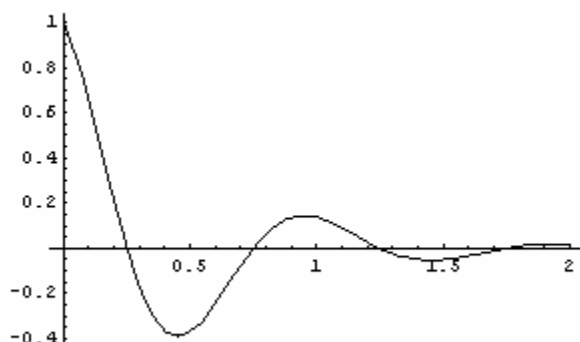


```
In[5]:= Plot[y1, {t, 0, 2}]
```



```
Out[5]= - Graphics -
```

```
In[6]:= Plot[x1, {t, 0, 2}]
```



```
Out[6]= - Graphics -
```

Формирование системы базисных функций:

```
In[7]:= m = 7; T = 2;
```

```
In[8]:= s[t_, n_, T_] := (t/T)^(n-1);
```

```
In[9]:= S := Table[s[t, n, T], {n, 1, m + 1}]
```

```
In[10]:= S
```

```
Out[10]= {1, t/2, t^2/4, t^3/8, t^4/16, t^5/32, t^6/64, t^7/128}
```

Формирование операционной матрицы аппроксимирующего полиномиального спектра:

```
In[11]:= w1 = Transpose[{S}].{S};
```

```
In[12]:= W =  $\int_0^2 w1 \, dt$ ;
```

```
In[13]:= N[%, 4] // MatrixForm
```

```
Out[13]/MatrixForm=
```

$$\begin{pmatrix} 2.000 & 1.000 & 0.6667 & 0.5000 & 0.4000 & 0.3333 & 0.2857 & 0.2500 \\ 1.000 & 0.6667 & 0.5000 & 0.4000 & 0.3333 & 0.2857 & 0.2500 & 0.2222 \\ 0.6667 & 0.5000 & 0.4000 & 0.3333 & 0.2857 & 0.2500 & 0.2222 & 0.2000 \\ 0.5000 & 0.4000 & 0.3333 & 0.2857 & 0.2500 & 0.2222 & 0.2000 & 0.1818 \\ 0.4000 & 0.3333 & 0.2857 & 0.2500 & 0.2222 & 0.2000 & 0.1818 & 0.1667 \\ 0.3333 & 0.2857 & 0.2500 & 0.2222 & 0.2000 & 0.1818 & 0.1667 & 0.1538 \\ 0.2857 & 0.2500 & 0.2222 & 0.2000 & 0.1818 & 0.1667 & 0.1538 & 0.1429 \\ 0.2500 & 0.2222 & 0.2000 & 0.1818 & 0.1667 & 0.1538 & 0.1429 & 0.1333 \end{pmatrix}$$

Нахождение операционных векторов спектров параметрических составляющих сигнала:

```
In[14]:= Qx = N[  $\int_0^T x1 * S \, dt$  ]
```

```
Out[14]= {0.0451573, -0.0100546, -0.00186559, -0.000121196,
0.000106425, 0.000249044, 0.000471636, 0.00072361}
```

```
In[15]:= Qy = N[  $\int_0^T y1 * S \, dt$  ]
```

```
Out[15]= {0.134216, -0.0176914, -0.0235391, -0.0220961,
-0.0214461, -0.0209269, -0.0201355, -0.0190608}
```

Определение векторов коэффициентов аппроксимирующих полиномов:

```
In[16]:= X = Inverse[W].Qx
```

```
Out[16]= {0.973018, -1.22185, -137.696, 952.569, -2618.98, 3551.21, -2369.1, 622.307}
```

```
In[17]:= Y = Inverse[W].Qy
```

```
Out[17]= {-0.0777317, 16.8954, -71.3631, -253.027, 1814.67, -3586.16, 3003.6, -924.678}
```

Формирование аппроксимирующих полиномов:

In[18]:= **x1a = X.S**

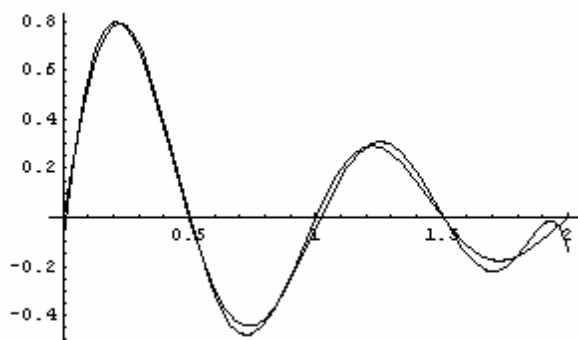
Out[18]= $0.973018 - 0.610924 t - 34.424 t^2 + 119.071 t^3 -$
 $163.686 t^4 + 110.975 t^5 - 37.0172 t^6 + 4.86178 t^7$

In[19]:= **y1a = Y.S**

Out[19]= $-0.0777317 + 8.44771 t - 17.8408 t^2 -$
 $31.6284 t^3 + 113.417 t^4 - 112.068 t^5 + 46.9313 t^6 - 7.22405 t^7$

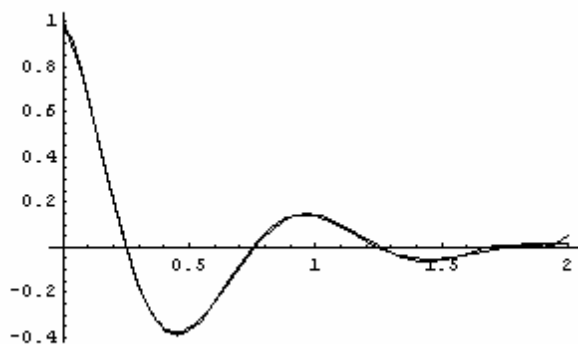
Отображение на графике аппроксимаций параметрических составляющих сигналов, совмещенных с исходными сигналами:

In[20]:= **Plot[{y1, y1a}, {t, 0, 2}]**



Out[20]= - Graphics -

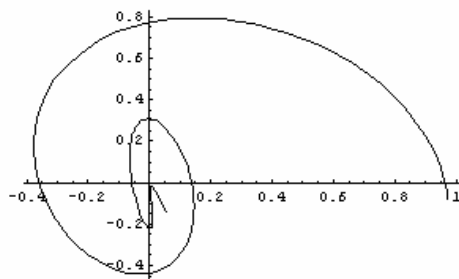
In[21]:= **Plot[{x1, x1a}, {t, 0, 2}]**



Out[21]= - Graphics -

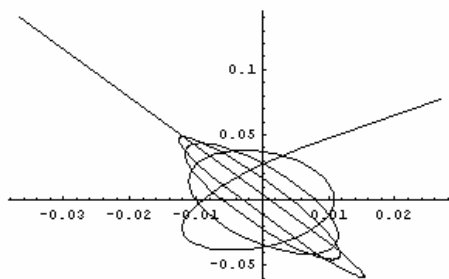
Построение параметрического графика аппроксимированного сигнала и параметрического графика функций ошибки аппроксимации:

```
In[22]:= PPa = ParametricPlot[{x1a, y1a}, {t, 0, 2}, PlotRange -> All]
```



```
Out[22]:= - Graphics -
```

```
In[23]:= PPe = ParametricPlot[{x1 - x1a, y1 - y1a}, {t, 0, 2}, PlotRange -> All]
```



```
Out[23]:= - Graphics -
```

Повторим процесс аппроксимации сигнала, заданного дискретными отсчетами параметрических составляющих сигнала.

Формирование массива отсчетов параметрических составляющих сигнала и аргумента (шаг по параметрическому аргументу t выбран равным 0.05):

```
In[24]:= h = 0.05;
```

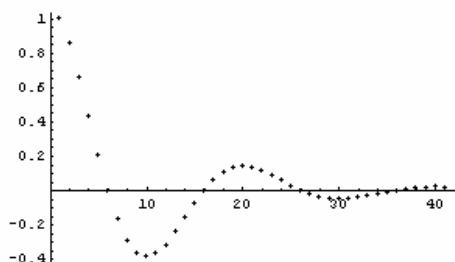
```
In[25]:= X1 = Table[x[i * h, 2 π], {i, 0, 40}];
```

```
In[26]:= Y1 = Table[y[i * h, 2 π], {i, 0, 40}];
```

```
In[27]:= Tt = Table[i * h, {i, 0, 40}];
```

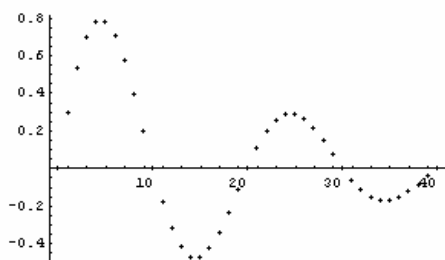
Отображение системы отсчетов параметрических составляющих сигнала на графиках:

```
In[28]:= Px = ListPlot[X1, PlotStyle → PointSize[0.015], PlotRange → All]
```



```
Out[28]= - Graphics -
```

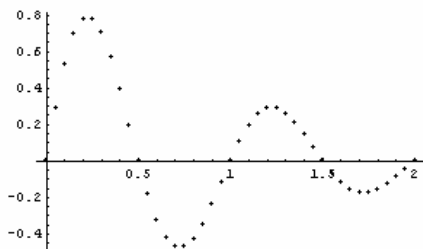
```
In[29]:= Py = ListPlot[Y1, PlotStyle → PointSize[0.015], PlotRange → All]
```



```
Out[29]= - Graphics -
```

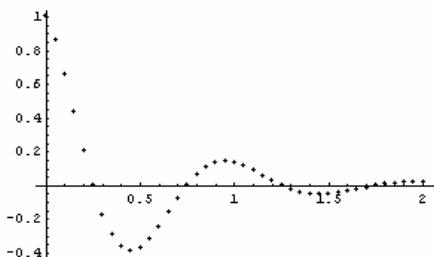
В приведенных графиках в качестве аргумента выступает номер отсчета. С целью сравнения удобно переформатировать эти графики перейдя к естественному параметрическому аргументу t :

```
In[30]:= Py2 = ListPlot[Table[{Tt[[k]], Y1[[k]]}, {k, 41}], PlotStyle → PointSize[0.015],  
PlotRange → All]
```



```
Out[30]= - Graphics -
```

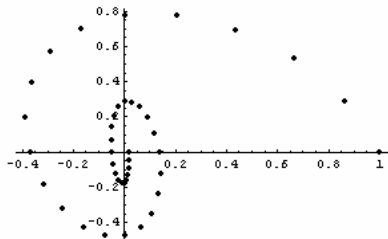
```
In[31]:= Px2 = ListPlot[Table[{Tt[[k]], X1[[k]]}, {k, 41}], PlotStyle -> PointSize[0.015],  
PlotRange -> All]
```



Out[31]= - Graphics -

Параметрическая форма дискретизированного сигнала:

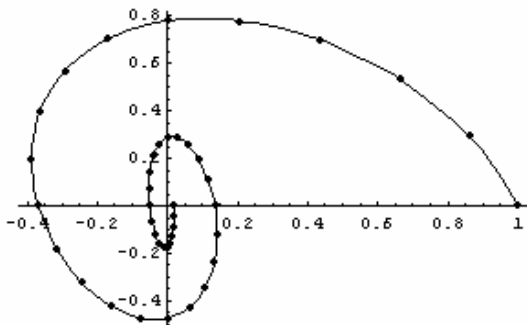
```
In[32]:= Pxy = ListPlot[Table[{X1[[k]], Y1[[k]]}, {k, 41}], PlotStyle -> PointSize[0.015],  
PlotRange -> All]
```



Out[32]= - Graphics -

Совмещение исходного и дискретизированного сигналов на одном графике в параметрической форме:

```
In[33]:= Show[{PP, Pxy}]
```



Out[33]= - Graphics -

Приведенные шаги программы иллюстрировали процесс получения массива данных дискретизированного сигнала и параметрического аргумента (X1,Y1, Tt).

Теперь переходим непосредственно к аппроксимации.

Формирование операционной матрицы спектра:

```
In[34]:= m = 7; T = 2;
```

```
In[35]:= w := Table[S[[p]] /. t -> Tt[[r]], {p, m + 1}, {r, 41}];
```

Определение векторов аппроксимирующих полиномиальных спектров:

```
In[36]:= q1 := X1;
```

```
In[37]:= q2 := Y1;
```

```
In[38]:= X = PseudoInverse[Transpose[w]].q1;
```

```
In[39]:= Y = PseudoInverse[Transpose[w]].q2;
```

```
In[40]:= X
```

```
Out[40]= {0.990612, -1.86241, -130.453, 915.584,  
          -2521.19, 3412.42, -2268.84, 593.389}
```

```
In[41]:= Y
```

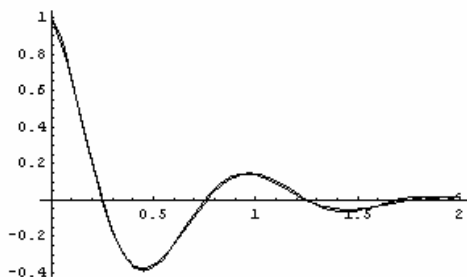
```
Out[41]= {-0.0407251, 16.0334, -65.9639, -262.709,  
          1801.47, -3521.48, 2928.8, -896.161}
```

Построение аппроксимирующих полиномов:

```
In[42]:= xa = X.S;
```

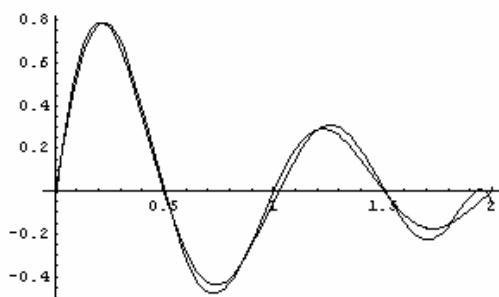
```
ya = Y.S;
```

```
In[44]:= Pxa = Plot[{x1, xa}, {t, 0, 2}]
```



```
Out[44]= - Graphics -
```

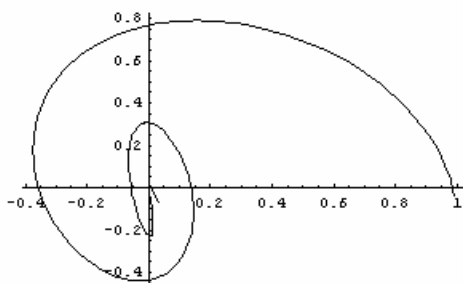
```
In[45]:= Pya = Plot[{y1, ya}, {t, 0, 2}]
```



```
Out[45]= - Graphics -
```

Отображение аппроксимации сигнала в параметрической форме:

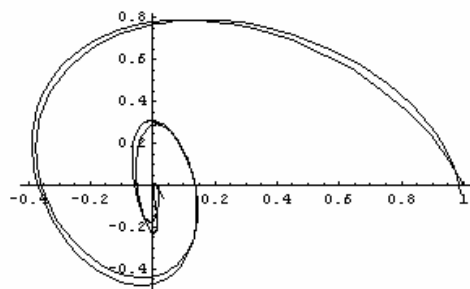
```
In[46]:= Pza = ParametricPlot[{xa, ya}, {t, 0, 2}, PlotRange → All]
```



```
Out[46]= - Graphics -
```

Сравнение в параметрической форме заданного сигнала и его аппроксимации, полученной на основе массива отсчетов:

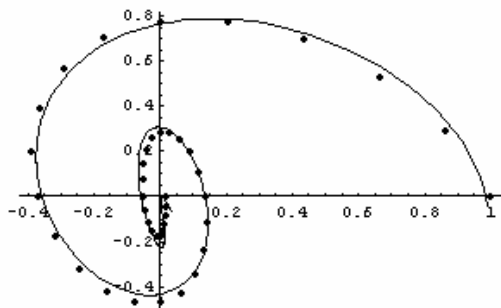
```
In[47]:= Show[{PP, Pza}]
```



```
Out[47]= - Graphics -
```


Сравнение аппроксимации сигнала с массивом отсчетов:

```
In[48]:= Show[{Pza, Pxy}]
```



```
Out[48]= - Graphics -
```

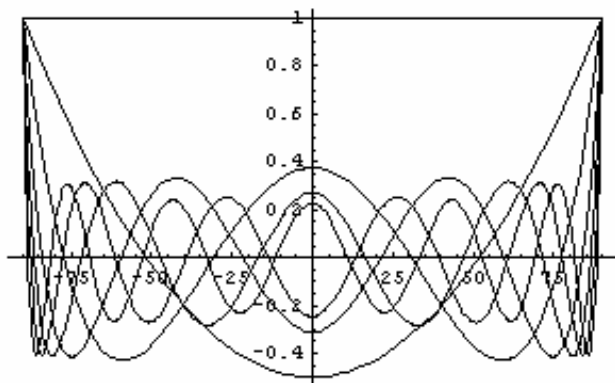
Пример 9. Аппроксимировать сигнал $e^{\frac{(x/90)^2}{2}} - e^{-5\frac{(x/90)^2}{2}}$, дискретизированный на интервале изменения аргумента $-90 \leq x \leq 90$, используя в качестве системы базисных функций систему ортогональных полиномов Лежандра. Аппроксимация непрерывной версии сигнала рассматривалась выше в примере 6.

Формирование системы базисных функций на основе четных полиномов Лежандра:

```
In[1]:= SL[x_] := Evaluate[Table[LegendreP[n, x], {n, 0, 12, 2}]];
```

```
In[2]:= s1 = Simplify[SL[x/90]];
```

```
In[3]:= Plot[Evaluate[s1], {x, -90, 90}]
```



```
Out[3]= - Graphics -
```

Задание формы сигнала и массива его дискретных значений (шаг по аргументу выбран равным 9):

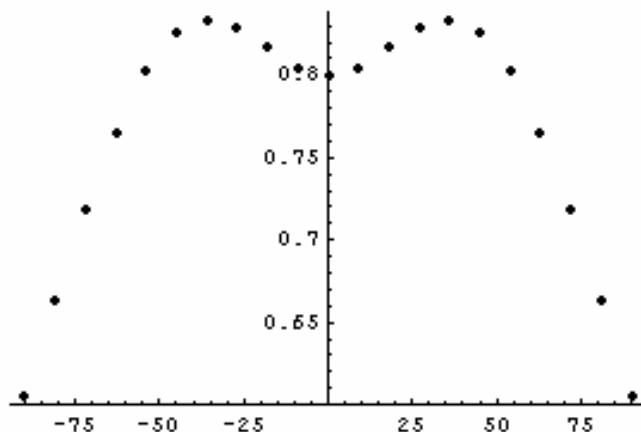
```
In[4]:= f = Exp[-0.5 (x / 90) ^ 2] - 0.2 Exp[-5 (x / 90) ^ 2];
```

```
In[5]:= h = 9;
```

```
In[6]:= fd := Table[{k * h, f /. x -> k * h}, {k, -10, 10}];
```

Отображение на графике отсчетов сигнала:

```
In[7]:= ListPlot[fd, PlotStyle -> PointSize[0.015]]
```



```
Out[7]:= - Graphics -
```

Формирование цифрового массива отсчета сигнала:

```
In[8]:= fd
```

```
Out[8]:= {{-90, 0.605183}, {-81, 0.663492}, {-72, 0.717997}, {-63, 0.765446},
{-54, 0.80221}, {-45, 0.825196}, {-36, 0.833251}, {-27, 0.828472},
{-18, 0.816453}, {-9, 0.804767}, {0, 0.8}, {9, 0.804767}, {18, 0.816453},
{27, 0.828472}, {36, 0.833251}, {45, 0.825196}, {54, 0.80221},
{63, 0.765446}, {72, 0.717997}, {81, 0.663492}, {90, 0.605183}}
```

```

In[9]:= fd // TableForm

Out[9]//TableForm=
  -90      0.605183
  -81      0.663492
  -72      0.717997
  -63      0.765446
  -54      0.80221
  -45      0.825196
  -36      0.833251
  -27      0.828472
  -18      0.816453
  -9       0.804767
  0        0.8
  9        0.804767
  18       0.816453
  27       0.828472
  36       0.833251
  45       0.825196
  54       0.80221
  63       0.765446
  72       0.717997
  81       0.663492
  90       0.605183

```

Формирование операционной матрицы и операционного вектора аппроксимирующего спектра Лежандра:

```

In[10]:= wd := Table[s1[[i]] /. x -> -90 + (k - 1) * h, {i, 7}, {k, 21}];

In[11]:= qd := Table[f /. x -> -90 + (k - 1) * h, {k, 21}];

```

Нахождение аппроксимирующего спектра Лежандра для сигнала:

```

In[13]:= Fd = PseudoInverse[Transpose[wd]].qd;

In[14]:= Fd

Out[14]= {0.776492, -0.131301, -0.0681974,
          0.0384969, -0.0128404, 0.00334062, -0.000808219}

```

Формирование аппроксимирующего полинома:

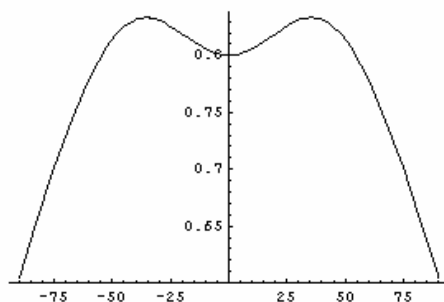
```

In[15]:= fda = Fd.s1;

```

Отображение на графике аппроксимации сигнала:

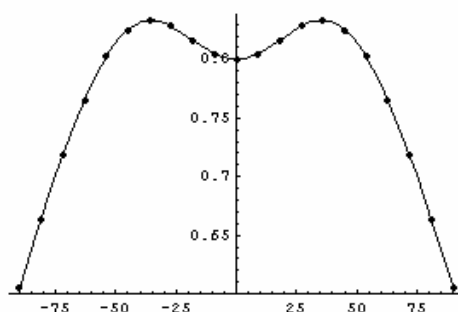
```
In[16]:= Plot[fda, {x, -90, 90}, PlotRange -> All]
```



```
Out[16]= - Graphics -
```

Совмещение на графике массива отсчетов и аппроксимации сигнала:

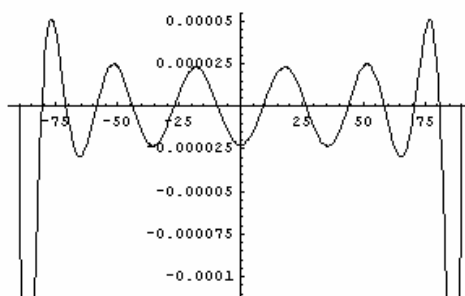
```
In[19]:= Show[{Out[7], Out[16]}, PlotRange -> All]
```



```
Out[19]= - Graphics -
```

Отображение функции ошибки аппроксимации сигнала на основе массива его значений:

```
In[18]:= Plot[f - fda, {x, -90, 90}]
```



```
Out[18]= - Graphics -
```

4. Локальные базисные системы

В рассмотренных выше примерах аппроксимации использовались системы базисных функций, которые обладают следующими двумя особенностями:

- функции системы являются непрерывными,
- энергия функций распределена на всем интервале аппроксимации.

Такие системы базисных функций принято называть глобальными. Определение коэффициентов аппроксимирующих полиномов в этих случаях выполняется путем интегрирования сигнала на всем диапазоне изменения аргумента. Это приводит к запаздыванию при аппроксимации на величину, превышающую этот диапазон. Кроме этого, при аппроксимации сигналов на достаточно больших интервалах изменения аргумента (интервалах наблюдения) для обеспечения требуемой точности аппроксимации необходимо значительно увеличивать порядок базисной системы функций. Использование методов аппроксимации с глобальными базисными системами в системах реального времени затруднительно. Избежать таких затруднений можно путем введения локальных аргументов и локальных базисных систем функций. Наибольшее распространение получили локальные импульсные базисные системы функций. Простейшей из таких систем является система блочно-импульсных функций (block-pulse functions) [10]. Эта система основана на использовании импульсных функций, имеющих вид прямоугольных импульсов, сдвинутых друг относительно друга на величину длительности импульса. Диапазон изменения аргумента сигнала (диапазон аппроксимации) разделяется на m подинтервалов и на полученной сетке вводится система упомянутых импульсных функций. Аналитическое выражение системы блочно-импульсных функций имеет вид:

$$v(t, i, m, h) = \{ \sigma(t - (i - 1)h) - \sigma(t - ih) \} \Big|_{i=1}^m, \quad (12)$$

где: $\sigma(x)$ - функция единичного скачка, равная 0 при отрицательных значениях аргумента и 1 в противном случае,

$h = T / m$ - шаг сетки аргумента (длина подинтервала).

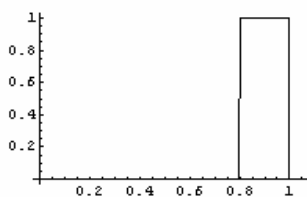
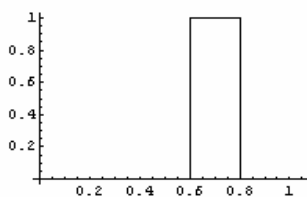
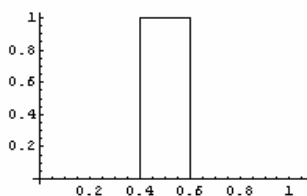
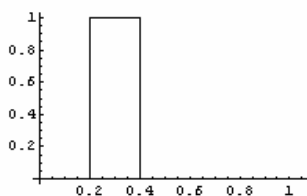
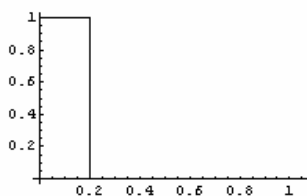
В качестве иллюстрации ниже приведена программа генерации системы блочно-импульсных функций в системе Mathematica для случая $h = T / m$ и $T = 1, m = 5$ и ее вид.

```
ln[1]:= v[t_, m_, h_] := Table[If[(i - 1) * h ≤ t < i * h, 1, 0],
    {i, m}];
```

```
ln[2]:= V = v[t, 5, 0.2];
```

```
ln[3]:= P := Table[Plot[V[[i]], {t, 0, 1.05}], {i, 5}];
```

```
ln[4]:= P
```



```
Out[4]= {-Graphics -, -Graphics -,
    -Graphics -, -Graphics -, -Graphics -}
```

Подобные базисные системы являются ортогональными, а энергия каждой базисной функции сосредоточена в ограниченной области изменения аргумента (в пределах шага сетки h). Задача нахождения системы коэффициентов аппроксимирующего полинома (блочно-импульсного спектра) упрощается и определяется следующими выражениями:

$$\begin{aligned}\bar{X} &= W^{-1} \cdot \bar{Q}, \\ W &= hE, \\ W^{-1} &= \frac{1}{h} E, \\ q_i &= \int_0^T x(t) v_i(t) dt = \int_{(i-1)h}^{ih} x(t) dt, \\ X_i &= \frac{q_i}{h} = \frac{m}{T} \int_{(i-1)h}^{ih} x(t) dt\end{aligned}\tag{13}$$

где E – единичная матрица порядка m .

Свойства функций базисной системы позволяют заменить интегрирование в пределах всего интервала изменения аргумента T интегрированием в пределах шага сетки h . Нахождение блочно-импульсного спектра сигнала осуществляется путем интегрирования сигнала в пределах h , а аппроксимация сигнала имеет вид кусочно-постоянной функции вида:

$$x_a(t) = \sum_{i=1}^m X_i v_i(t).\tag{14}$$

Повышения точности аппроксимации можно достигнуть расширением системы базисных функций, вводя подсистему сдвинутых на величину шага линейных импульсов. Выражение такой подсистемы имеет вид:

$$w(t, i, m, h) = \left\{ \left(\frac{2t}{h} - 1 - 2(i-1) \right) v(t, i, m, h) \right\} \Bigg|_{i=1}^m\tag{15}$$

Соответствующая (15) функция изменяется линейно от -1 до $+1$ в пределах i -го шага h и равна нулю за его пределами. Расширенная подобным образом система базисных функций генерируется следующей программой и имеет вид:

```
In[1]:= v[t_, m_, h_] := Table[If[(i - 1) * h ≤ t < i * h, 1, 0],
    {i, m}];
```

```
In[2]:= w[t_, m_, h_] :=
    Table[(2 t / h - 1 - 2 (i - 1)) * v[t, m, h][[i]], {i, m}];
```

```
In[3]:= V = v[t, 5, 0.2];
```

```
In[4]:= W = w[t, 5, 0.2];
```

Аппроксимация сигналов в такой расширенной системе базисных функций является кусочно-линейной и наилучшей в средне-квадратичном смысле. Систему коэффициентов аппроксимирующего полинома называют аппроксимирующим импульсным спектром сигнала (АИС) [10]. Аппроксимирующий импульсный спектр сигнала в расширенной системе базисных функций является вектором порядка $2m$:

$$\bar{X} = \begin{bmatrix} \bar{X}_0 \\ \bar{X}_1 \end{bmatrix},$$

где \bar{X}_0 - вектор порядка m , определяется в соответствии с выражениями (13), \bar{X}_1 - вектор порядка m , определяется по аналогичным выражениям (16):

$$\begin{aligned} \bar{X}_1 &= W_1^{-1} \cdot \bar{Q}_1, \\ W_1 &= \frac{h}{3} E, \\ W_1^{-1} &= \frac{3}{h} E, \\ q_{1i} &= \int_0^T x(t) w_i(t) dt = \int_{(i-1)h}^{ih} x(t) w_i dt, \\ X_{1i} &= \frac{3q_i}{h} = \frac{3m}{T} \int_{(i-1)h}^{ih} x(t) w_i dt \end{aligned} \tag{16}$$

Аппроксимация сигнала по расширенной системе базисных функций определяется выражениями:

$$x_a(t) = \sum_{i=1}^m (X_{0i}v_i(t) + X_{1i}w_i(t)) = \bar{X}_0^T \cdot \bar{V}(t) + \bar{X}_1^T \cdot \bar{W}(t) \quad (17)$$

Дальнейшее увеличение точности аппроксимации может быть достигнуто путем введения подсистемы импульсных функций, представляющих собой сдвинутые на шаг сетки импульсы параболической формы:

$$u(t, i, m, h) = \left\{ \left(6 \left(\frac{t}{h} \right)^2 - \frac{6(2i-1)t}{h} + 6i^2 - 6i + 1 \right) v(t, i, m, h) \right\} \Bigg|_{i=1}^m \quad (18)$$

Аппроксимирующий импульсный спектр сигнала становится вектором порядка $3m$ вида:

$$\bar{X} = \begin{bmatrix} \bar{X}_0 \\ \bar{X}_1 \\ \bar{X}_2 \end{bmatrix},$$

где \bar{X}_0, \bar{X}_1 определяются по выражениям, приведенным выше, а элементы вектора \bar{X}_2 вычисляются по формулам:

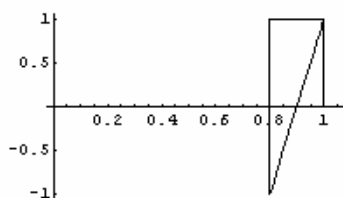
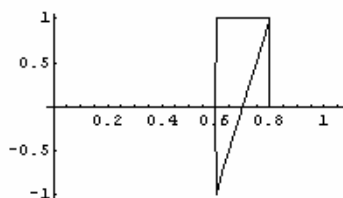
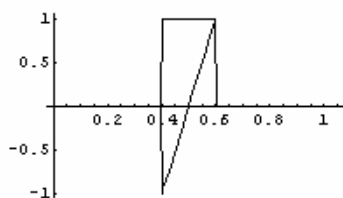
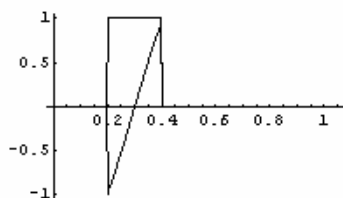
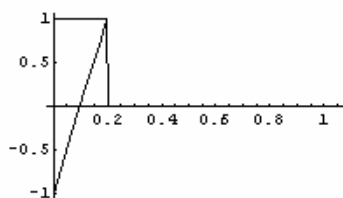
$$X_{2i} = \frac{5m}{T} \int_{(i-1)h}^{ih} x(t) u_i dt \quad (19)$$

Соответственно изменится выражение для аппроксимирующего полинома:

$$x_a(t) = \sum_{i=1}^m (X_{0i}v_i(t) + X_{1i}w_i(t) + X_{2i}u_i(t)) = \bar{X}_0^T \cdot \bar{V}(t) + \bar{X}_1^T \cdot \bar{W}(t) + \bar{X}_2^T \cdot \bar{U}(t) \quad (20)$$

```
In[5]:= P := Table[Plot[{V[[i]], W[[i]]}, {t, 0, 1.05}], {i, 5}];
```

```
In[6]:= P
```



```
Out[6]= {- Graphics -, - Graphics -,  
          - Graphics -, - Graphics -, - Graphics -}
```

Генерирующая программа для такой системы функций и графическая иллюстрация имеют вид:

```
In[1]:= v[t_, m_, h_] := Table[If[(i - 1) * h ≤ t < i * h, 1, 0],
    {i, m}];
```

```
In[2]:= w[t_, m_, h_] :=
    Table[(2 t / h - 1 - 2 (i - 1)) * v[t, m, h][[i]], {i, m}];
```

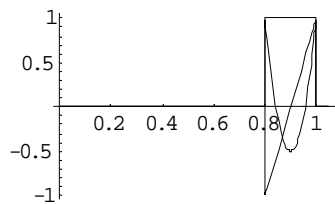
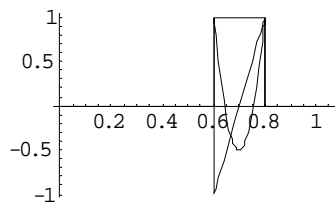
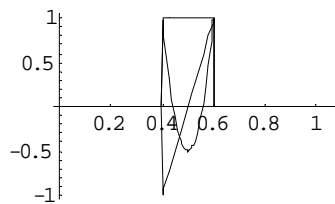
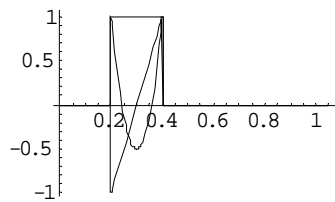
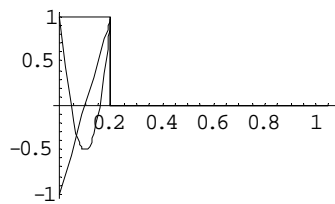
```
In[3]:= u[t_, m_, h_] :=
    Table[(6 (t / h)^2 - (6 (2 i - 1) * t) / h + 6 i^2 - 6 i + 1) *
    v[t, m, h][[i]], {i, m}];
```

```
In[4]:= V = v[t, 5, 0.2];
```

```
In[5]:= W = w[t, 5, 0.2];
    U = u[t, 5, 0.2];
```

```
In[7]:= P := Table[Plot[{V[[i]], W[[i]], U[[i]]}, {t, 0, 1.05}],
    {i, 5}];
```

```
In[8]:= P
```



Приведенные выше три локальных базисных системы являются привлекательными для их использования в анализе и моделировании по следующим причинам:

- все подсистемы и система в целом являются ортогональными системами функций, что существенно упрощает решение задачи аппроксимации сигналов,
- определение коэффициентов аппроксимирующих полиномов производится путем интегрирования сигнала, умноженного на соответствующую базисную функцию, причем интегрирование осуществляется только в пределах шага сетки h ,
- аппроксимации сигналов в зависимости от вида базисной системы являются кусочно-постоянными, кусочно-линейными или кусочно-параболическими функциями,
- при аппроксимации одного и того же сигнала последовательно расширяющимися базисными системами функций (сначала блочно-импульсной, затем на основе смещенных полиномов Лежандра первого и второго порядков) коэффициенты аппроксимирующего полинома, соответствующие блочно-импульсной подсистеме, не изменяют своих значений при расширении базисной системы до смещенных полиномов Лежандра первого порядка, и далее, коэффициенты аппроксимирующего полинома, соответствующие системе базисных функций на основе смещенных полиномов Лежандра, не изменяют своих значений при расширении базисной системы до смещенных полиномов Лежандра второго порядка,
- коэффициенты аппроксимирующих полиномов, соответствующие блочно-импульсным функциям по физическому смыслу являются средними значениями сигналов на соответствующем подинтервале изменения аргумента; коэффициенты, соответствующие второй и третьей подсистемам базисных функций по физическому смыслу являются соответственно средними значениями первой и второй производной сигнала на шаге сетки,
- если сигнал является реализацией какого-либо динамического процесса (обобщенного движения), то коэффициенты аппроксимирующего полинома на шаге сетки пропорциональны средним значениям обобщенных перемещения, скорости и ускорения.

Проиллюстрируем аппроксимацию сигналов на основе локальных импульсных базисных систем (смещенных полиномов

Лежандра нулевого, первого и второго порядков) на следующем примере:

Пример 10. Аппроксимировать сигнал $f(t) = \sin(\frac{\pi t}{2})$ на интервале

изменения аргумента $0 \leq t < 1$, используя системы базисных функций на основе смещенных полиномов Лежандра при разбиении интервала изменения аргумента на 5 одинаковых подинтервалов ($m=5$).

Фрагмент программы, генерирующей системы базисных функций:

```
In[1]:= v[t_, m_, h_] := Table[If[(i - 1) * h ≤ t < i * h, 1, 0],
    {i, m}];
```

```
In[2]:= w[t_, m_, h_] := Table[(2 t / h + 1 - 2 i) * v[t, m, h][[i]],
    {i, m}];
```

```
In[3]:= u[t_, m_, h_] :=
    Table[(6 (t/h)^2 - (6 (2 i - 1) * t) / h + 6 i^2 - 6 i + 1) *
        v[t, m, h][[i]], {i, m}];
```

```
In[4]:= V = v[t, 5, 0.2];
```

```
In[5]:= W = w[t, 5, 0.2];
```

```
In[6]:= U = u[t, 5, 0.2];
```

Подпрограмма, определяющая коэффициенты аппроксимирующих полиномов для различных подсистем базисных функций, и результаты ее работы приведены ниже.

```
In[7]:= h = 0.2; m = 5;
```

```
In[8]:= f[t_] := Sin[π * t / 2];
```

```
In[9]:= sp0 := Table[1/h * ∫(i-1)*hi*h f[t] dt, {i, m}];
```

```
In[10]:= sp1 := Table[3/h * ∫(i-1)*hi*h f[t] * (2 t / h + 1 - 2 i) dt, {i, m}];
```

```
In[11]:= sp2 :=
Table[
  
$$\frac{5}{h} * \int_{(i-1)*h}^{i*h} f[t] * \left( 6 \left( \frac{t}{h} \right)^2 - \frac{6 (2 i - 1) * t}{h} + 6 i^2 - 6 i + 1 \right) dt,$$

  {i, m}];
```

```
In[12]:= sp0
```

```
Out[12]:= {0.155792, 0.452126, 0.704203, 0.887347, 0.983632}
```

```
In[13]:= sp1
```

```
Out[13]:= {0.154763, 0.139614, 0.110798, 0.0711369, 0.0245121}
```

```
In[14]:= sp2
```

```
Out[14]:= {-0.00128436, -0.00372735,
            -0.00580548, -0.00731533, -0.0081091}
```

Реконструкции сигнала на основе последовательно расширяющихся базисных систем определяются следующими выражениями:

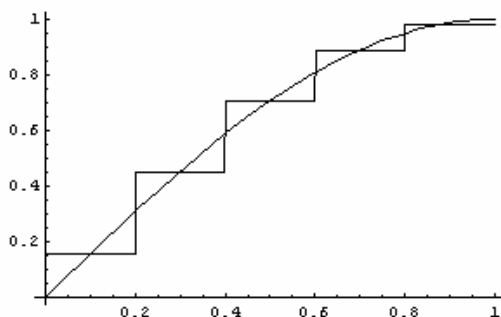
```
In[15]:= fa0 := Evaluate[sp0.V];
```

```
In[16]:= fa1 := Evaluate[sp0.V + sp1.W];
```

```
In[17]:= fa2 := Evaluate[sp0.V + sp1.W + sp2.U];
```

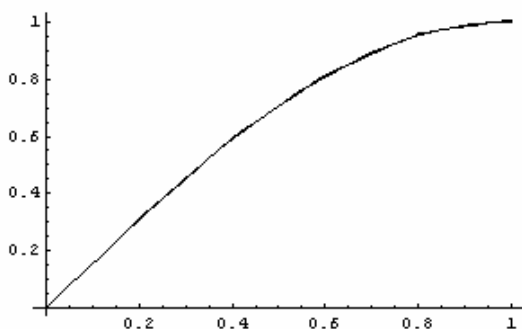
Графики аппроксимаций сигнала различными системами базисных функций, совмещенные с аппроксимируемым сигналом, имеют вид:

```
In[18]:= Plot[{f[t], fa0}, {t, 0, 1}]
```



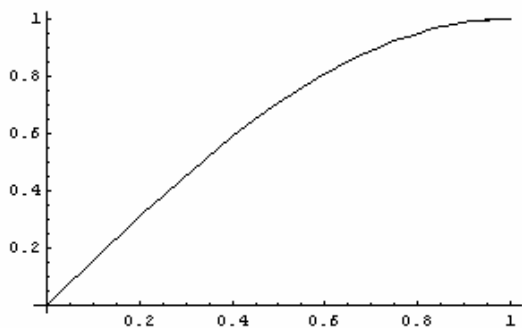
```
Out[18]:= - Graphics -
```

```
In[19]:= Plot[{f[t], fa1}, {t, 0, 1}]
```



```
Out[19]= - Graphics -
```

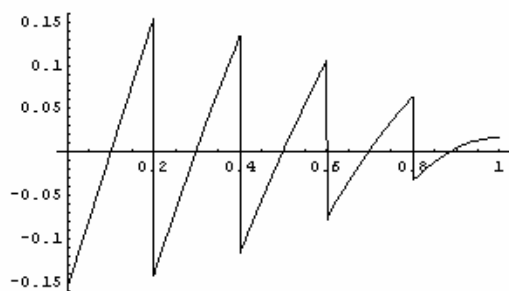
```
In[20]:= Plot[{f[t], fa2}, {t, 0, 1}]
```



```
Out[20]= - Graphics -
```

Функции ошибок аппроксимации изображены ниже.

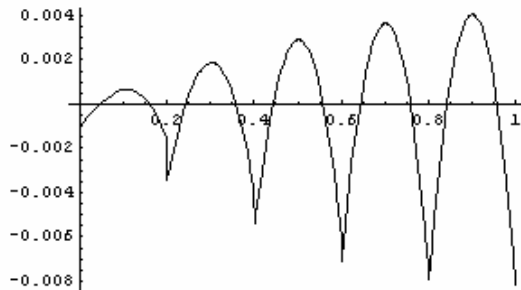
```
In[21]:= Plot[f[t] - fa0, {t, 0, 1}]
```



```
Out[21]= - Graphics -
```



```
In[22]:= Plot[f[t] - fa1, {t, 0, 1}]
```



```
Out[22]= - Graphics -
```

```
In[23]:= Plot[f[t] - fa2, {t, 0, 1}]
```



```
Out[23]= - Graphics -
```

Пример 11. Аппроксимировать сигнал $f(t) = 5 \cos(\pi t) + 6e^{-15(t-1)^2}$ на интервале изменения аргумента $0 \leq t < 2$, используя локальные базисные системы на основе смещенных полиномов Лежандра нулевого, первого и второго порядков.
Задание базисных систем функций:

```
In[1]:= v[t_, m_, h_] := Table[If[(i - 1) * h ≤ t < i * h, 1, 0],
    {i, m}];
```

```
In[2]:= w[t_, m_, h_] := Table[(2 t / h + 1 - 2 i) * v[t, m, h][[i]],
    {i, m}];
```

```
In[3]:= u[t_, m_, h_] :=
Table[ $\left(6 \left(\frac{t}{h}\right)^2 - \frac{6 (2 i - 1) * t}{h} + 6 i^2 - 6 i + 1\right) * v[t, m, h][[i]], \{i, m\}];$ 
```

```
In[4]:= V = v[t, 20, 0.1];
```

```
In[5]:= W = w[t, 20, 0.1];
```

```
In[6]:= U = u[t, 20, 0.1];
```

Задание шага по аргументу, порядка базисной системы функций и сигнала, подлежащего аппроксимации:

```
In[7]:= h = 0.1; m = 20;
```

```
In[8]:= f[t_] := 5 Cos[ $\pi * t$ ] + 6 e-15 (t-1)2;
```

Определение локального импульсного спектра Лежандра нулевого порядка (блочно-импульсного спектра):

```
In[9]:= sp0 := Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} f[t] dt, \{i, m\}];$ 
```

Определение локального импульсного спектра Лежандра первого порядка (аппроксимирующего импульсного спектра):

```
In[10]:= sp1 := Table[ $\frac{3}{h} * \int_{(i-1)*h}^{i*h} f[t] * (2 t / h + 1 - 2 i) dt, \{i, m\}];$ 
```

Определение локального импульсного спектра Лежандра второго порядка:

```
In[11]:= sp2 :=
Table[
 $\frac{5}{h} * \int_{(i-1)*h}^{i*h} f[t] * \left(6 \left(\frac{t}{h}\right)^2 - \frac{6 (2 i - 1) * t}{h} + 6 i^2 - 6 i + 1\right) dt,$ 
 $\{i, m\}];$ 
```

Численное значение спектра нулевого порядка:

```
In[12]:= sp0
```

```
Out[12]= {4.91817, 4.43688, 3.52258, 2.27283, 0.849733,
-0.472934, -1.27365, -1.1462, -0.172932, 0.794873,
0.794873, -0.172932, -1.1462, -1.27365, -0.472934,
0.849733, 2.27283, 3.52258, 4.43688, 4.91817}
```

Численное значение спектра первого порядка:

```
In[13]:= sp1
```

```
Out[13]= {-0.122547, -0.355512, -0.552373, -0.686962, -0.718431,  
          -0.575156, -0.194086, 0.319683, 0.590981, 0.301491,  
          -0.301491, -0.590981, -0.319683, 0.194086, 0.575156,  
          0.718431, 0.686962, 0.552373, 0.355512, 0.122547}
```

Численное значение спектра второго порядка:

```
In[14]:= sp2
```

```
Out[14]= {-0.0405397, -0.0365102, -0.0284787, -0.0154142, 0.00680277,  
          0.0430547, 0.0818118, 0.078951, 0.00146062, -0.0911384,  
          -0.0911384, 0.00146062, 0.078951, 0.0818118, 0.0430547,  
          0.00680277, -0.0154142, -0.0284787, -0.0365102, -0.0405398}
```

Формирование аппроксимаций сигнала различных порядков:

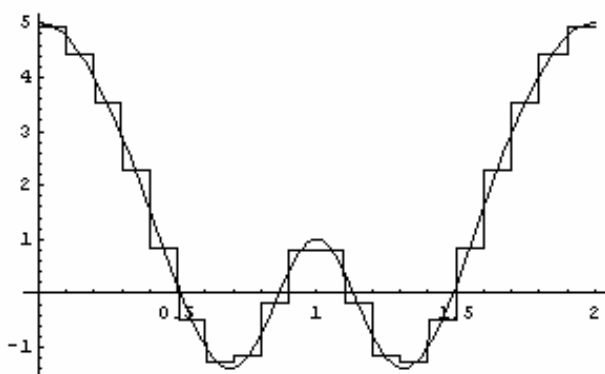
```
In[15]:= fa0 := Evaluate[sp0.V];
```

```
In[16]:= fa1 := Evaluate[sp0.V + sp1.W];
```

```
In[17]:= fa2 := Evaluate[sp0.V + sp1.W + sp2.U];
```

Аппроксимация сигнала нулевого порядка, совмещенная с сигналом:

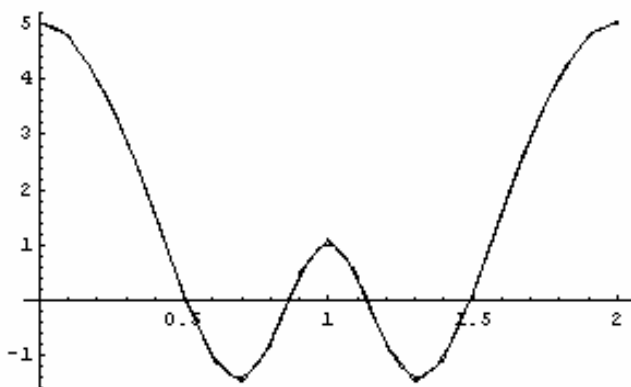
```
In[18]:= Plot[{f[t], fa0}, {t, 0, 2}]
```



```
Out[18]= - Graphics -
```

Аппроксимация сигнала первого порядка, совмещенная с сигналом:

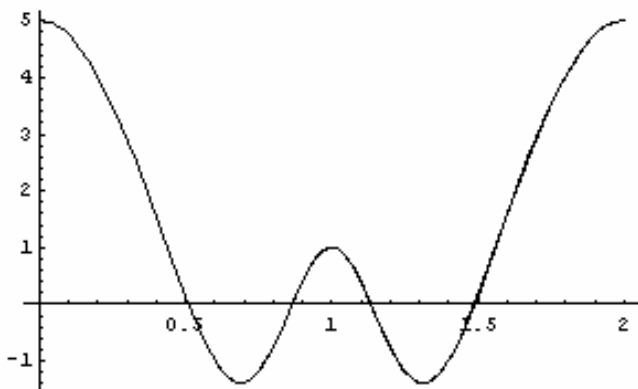
```
In[19]:= Plot[{f[t], fa1}, {t, 0, 2}]
```



```
Out[19]= - Graphics -
```

Аппроксимация сигнала первого порядка, совмещенная с сигналом:

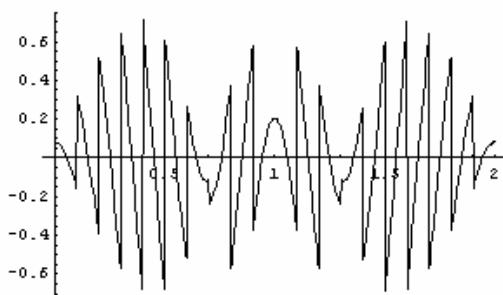
```
In[20]:= Plot[{f[t], fa2}, {t, 0, 2}]
```



```
Out[20]= - Graphics -
```

Функция ошибки аппроксимации нулевого порядка:

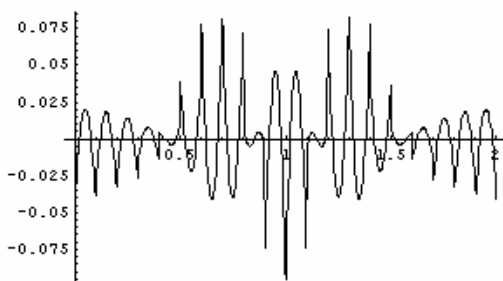
```
In[21]:= Plot[f[t] - fa0, {t, 0, 2}]
```



```
Out[21]= - Graphics -
```

Функция ошибки аппроксимации первого порядка:

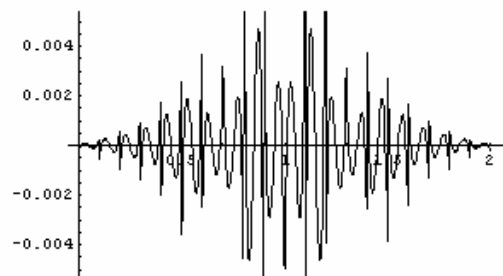
```
In[22]:= Plot[f[t] - fa1, {t, 0, 2}]
```



```
Out[22]= - Graphics -
```

Функция ошибки аппроксимации второго порядка:

```
In[23]:= Plot[f[t] - fa2, {t, 0, 2}]
```



```
Out[23]= - Graphics -
```

5. АППРОКСИМАЦИЯ ДВУМЕРНЫХ СИГНАЛОВ В ЛОКАЛЬНЫХ БАЗИСНЫХ СИСТЕМАХ СЕПАРАБЕЛЬНОГО ТИПА

Полиномиальная аппроксимация двумерных сигналов связана с использованием базисных систем функций, зависящих от двух аргументов. В качестве таких систем могут использоваться полиномы, определенные в комплексной области, степенные полиномы двух действительных переменных, обобщения на двумерный случай локальных импульсных базисных систем. Ниже рассматривается аппроксимация сигнала $f(x,y)$ полиномом вида:

$$f_a(x, y) = \sum_{i=1}^m \sum_{j=1}^n (F_{ij} \cdot v_{ij}(x, y) + F_{xij} \cdot w_{xij}(x, y) + F_{yij} \cdot w_{yij}(x, y)) \quad (21)$$

Базисная система функций строится путем обобщения систем на основе смещенных полиномов Лежандра, причем используются сепарабельные версии, получаемые произведением одномерных базисных функций различных переменных [13].

Систему базисных функций независимых переменных x и y введем следующим образом. Пусть аргументы x и y изменяются в пределах $0 \leq x < T_x$ и $0 \leq y < T_y$. Введем на полученной прямоугольной области сетку, разбивая ее на m x n ячеек, причем

$$h_x = \frac{T_x}{m}, h_y = \frac{T_y}{n}. \quad (22)$$

Используя введенные выше одномерные версии базисных подсистем функций на основе смещенных полиномов Лежандра нулевого и первого порядков, образуем двумерные базисные подсистемы в соответствии с выражениями:

$$v_{ij}(x, y) = v_i(x) \cdot v_j(y); \quad (23)$$

$$w_{xij}(x, y) = w_i(x) \cdot v_j(y); \quad (24)$$

$$w_{yij}(x, y) = w_j(y) \cdot v_i(x); \quad (25)$$

где

$$v_i(x) = \sigma(x - (i-1)h_x) - \sigma(x - ih_x), \quad (26)$$

$$v_j(y) = \sigma(y - (j-1)h_y) - \sigma(y - jh_y), \quad (27)$$

$$w_i(x) = \left(\frac{2x}{h_x} - 1 - 2(i-1)\right)v_i(x), \quad (28)$$

$$w_j(y) = \left(\frac{2y}{h_y} - 1 - 2(j-1)\right)v_j(y). \quad (29)$$

Коэффициенты аппроксимирующего полинома (F_{ij}, F_{xij}, F_{yij}) определяются из условия минимума двойного интеграла квадрата функции ошибки, взятого по области изменения аргументов сигнала:

$$J = \int_0^{T_x} \int_0^{T_y} (f(x, y) - f_a(x, y))^2 dy dx \rightarrow \min. \quad (30)$$

С учетом ортогональности системы базисных функций эти коэффициенты определяются выражениями:

$$F_{ij} = \frac{1}{h_x h_y} \int_{(i-1)h_x}^{ih_x} \int_{(j-1)h_y}^{jh_y} f(x, y) dy dx, \quad (31)$$

$$F_{xij} = \frac{3}{h_x h_y} \int_{(i-1)h_x}^{ih_x} \int_{(j-1)h_y}^{jh_y} f(x, y) \cdot w_i(x) dy dx, \quad (32)$$

$$F_{yij} = \frac{3}{h_x h_y} \int_{(i-1)h_x}^{ih_x} \int_{(j-1)h_y}^{jh_y} f(x, y) \cdot w_j(y) dy dx, \quad (33)$$

Коэффициенты (31)-(33) по физическому смыслу соответствуют: F_{ij} - среднему значению сигнала, F_{xij} - среднему значению первой частной производной по x , F_{yij} - среднему значению первой частной производной по y на элементе сеточной области ij .

В целом выражение (21) является кусочно-плоскостной аппроксимацией сигнала, наилучшей в средне-квадратичном смысле.

Рассмотрим пример формирования локальной базисной системы функций двух переменных.

Пример 12. Определить локально-импульсную базисную систему функций, определенную на единичном квадрате изменения

аргументов ($T_x = T_y = 1$) и на равномерной сетке с шагом $h_x = h_y = 0.1$. Программа-генератор такой системы функций и результаты ее работы приведены ниже. На графиках изображены три характерных базисных функции для случая $i = j = 5$.

Задание параметров системы функций:

```
In[1]:= m := 10; n := 10; Tx := 1; Ty := 1;
```

Формирование подсистемы двумерной базисной системы функций нулевого порядка:

```
In[2]:= v[x_, i_, m_, T_] := If[(i - 1) T / m ≤ x < i T / m, 1, 0];
```

```
In[3]:= vx = v[x, i, m, Tx];
```

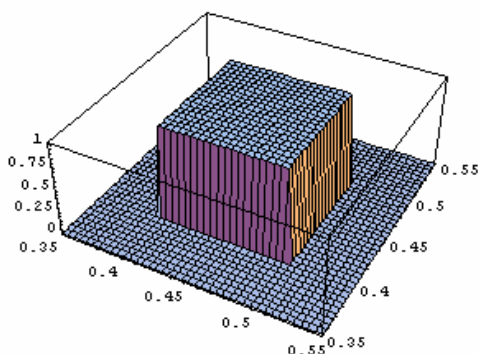
```
In[4]:= vy = v[y, j, n, Ty];
```

```
In[5]:= v = vx * vy;
```

```
In[6]:= Vxy = Table[v, {i, m}, {j, n}];
```

Отображение одной из функций сформированной подсистемы на части координатной плоскости:

```
In[7]:= Plot3D[Vxy[[5, 5]], {x, 0.35, 0.55}, {y, 0.35, 0.55},  
PlotPoints → 40]
```



```
Out[7]= - SurfaceGraphics -
```

Формирование подсистем двумерной базисной системы функций первого порядка:


```
In[8]:= w[x_, i_, m_, T_] := 1 - 2 i +  $\frac{2 m * x}{T}$  ;
```

```
In[9]:= wx = w[x, i, m, Tx] * v;
```

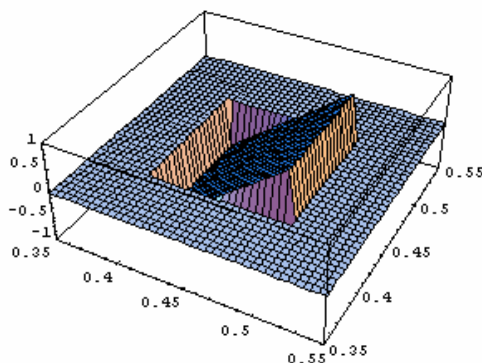
```
In[10]:= Wx = Table[wx, {i, m}, {j, n}];
```

```
In[11]:= wy = w[y, j, n, Ty] * v;
```

```
In[12]:= Wy = Table[wy, {i, m}, {j, n}];
```

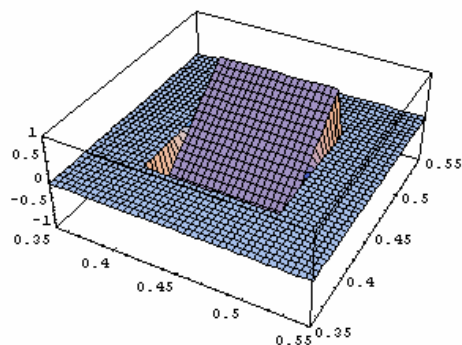
Отображение некоторых функций сформированной подсистемы на части координатной плоскости:

```
In[13]:= Plot3D[Wx[[5, 5]], {x, 0.35, 0.55}, {y, 0.35, 0.55},  
PlotPoints -> 40]
```



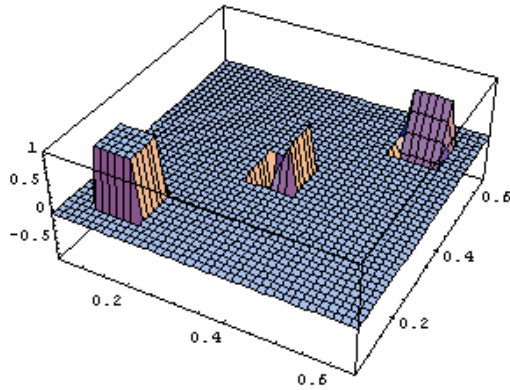
```
Out[13]= - SurfaceGraphics -
```

```
In[14]:= Plot3D[Wy[[5, 5]], {x, 0.35, 0.55}, {y, 0.35, 0.55},  
PlotPoints -> 40]
```



```
Out[14]= - SurfaceGraphics -
```

```
In[15]:= Plot3D[Vxy[[2, 2]] + Wx[[4, 4]] + Wy[[6, 6]], {x, 0.05, 0.65},
           {y, 0.05, 0.65}, PlotPoints -> 40]
```



```
Out[15]= - SurfaceGraphics -
```

Рассмотрим несколько иллюстративных примеров аппроксимации двумерных сигналов.

Пример 13. Аппроксимировать сигнал $f(x, y) = 1 - x^2 - y^2$, определенный в области $\begin{pmatrix} 0 \leq x < 1, \\ 0 \leq y < 1 \end{pmatrix}$, применив равномерную сетку с шагом $h_x = h_y = 0.1$.

Программа аппроксимации сигнала и результаты ее работы приведены ниже.

Определение двумерной базисной системы функций

```
In[1]:= v[x_, i_, m_, T_] := If[(i - 1) T / m ≤ x < i T / m, 1, 0];
```

```
In[2]:= vx = v[x, i, 10, 1];
```

```
In[3]:= vy = v[y, j, 10, 1];
```

```
In[4]:= Vxy = vx vy;
```

```
In[5]:= vxy1 := Table[Vxy, {i, 1, 10}, {j, 1, 10}];
```

```
In[6]:= w[x_, i_, m_, T_] :=
        (1 - 2 i + 2 m x / T) If[(i - 1) T / m ≤ x < i T / m, 1, 0];
```

```
In[7]:= wx = w[x, i, 10, 1] vy;
```

```
In[8]:= wy = w[y, j, 10, 1] vx;
```

```
In[9]:= zx := Table[wx, {i, 1, 10}, {j, 1, 10}];
```

```
In[10]:= zy := Table[wy, {i, 1, 10}, {j, 1, 10}];
```

Определение сигнала, подлежащего аппроксимации:

```
In[11]:= f[x_, y_] := 1 - x2 - y2;
```

```
In[12]:= f1 = f[x, y];
```

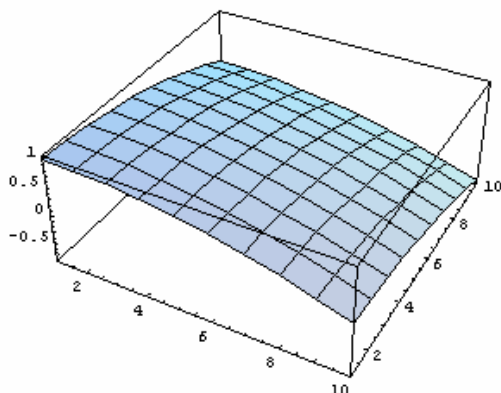
Нахождение аппроксимирующего блочно-импульсного спектра сигнала:

```
In[13]:= Fv[i_, j_] := 100  $\int_{(i-1)/10}^{i/10} \int_{(j-1)/10}^{j/10} f1 dy dx$ ;
```

```
In[14]:= FV = Table[Fv[i, j], {i, 1, 10}, {j, 1, 10}];
```

Отображение массива спектра сигнала с интерполяцией по 100 элементам спектра (внутренняя функция команды ListPlot3D):

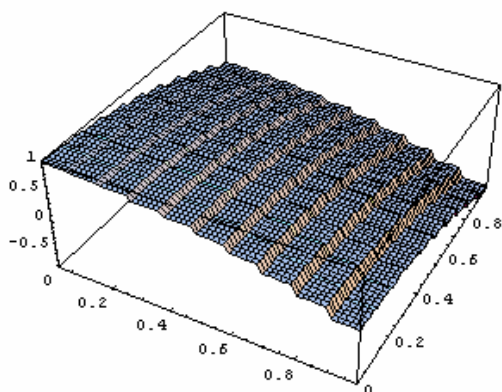
```
In[15]:= ListPlot3D[%]
```



```
Out[15]= - SurfaceGraphics -
```

Определение аппроксимации сигнала по блочно-импульсному спектру:

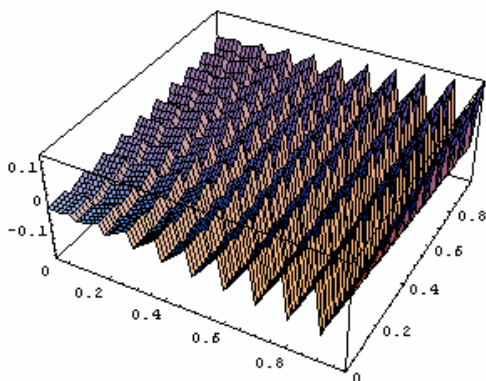
```
In[17]:= Plot3D[fa, {x, 0, 0.99}, {y, 0, 0.99}, PlotPoints -> 60]
```



```
Out[17]= - SurfaceGraphics -
```

Визуализация функции ошибки аппроксимации:

```
In[18]:= Plot3D[fa - f1, {x, 0, 0.99}, {y, 0, 0.99}, PlotPoints -> 60]
```



```
Out[18]= - SurfaceGraphics -
```

Нахождение фрагмента аппроксимирующего импульсного спектра сигнала, соответствующего кусочно-линейным базисным функциям:

```
In[19]:= Fwx[i_, j_] := 300  $\int_{(i-1)/10}^{i/10} \int_{(j-1)/10}^{j/10} f1 \, wx1 \, dy \, dx;$ 
```

```
In[20]:= wx1 = w[x, i, 10, 1];
```

```
In[21]:= wy1 = w[y, j, 10, 1];
```

```
In[22]:= Fwy[i_, j_] := 300  $\int_{(j-1)/10}^{j/10} \int_{(i-1)/10}^{i/10} f1 wy1 dx dy$ ;
```

```
In[23]:= Fwx1 = Table[Fwx[i, j], {i, 1, 10}, {j, 1, 10}];
```

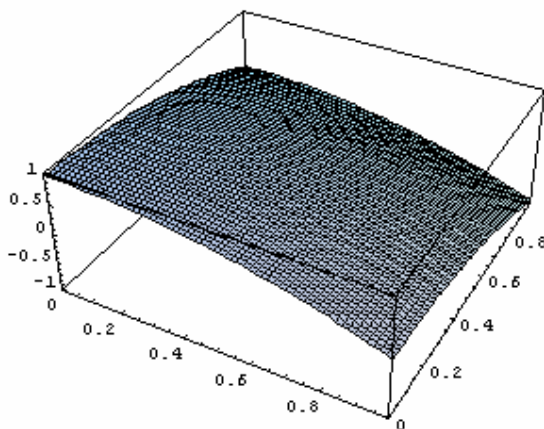
```
In[24]:= Fwy1 = Table[Fwy[i, j], {i, 1, 10}, {j, 1, 10}];
```

Определение аппроксимации сигнала по аппроксимирующему импульсному спектру:

```
In[25]:= fa1 =
```

$$\sum_{i=1}^{10} \sum_{j=1}^{10} \{FV[[i, j]] + Fwx1[[i, j]] zx[[i, j]] + Fwy1[[i, j]] zy[[i, j]]\} vxy1[[i, j]];$$

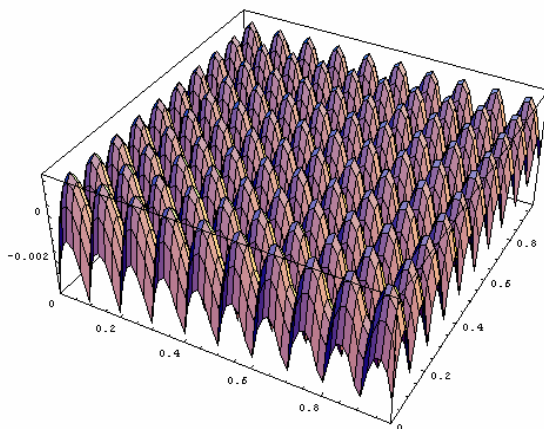
```
In[26]:= Plot3D[fa1, {x, 0, 0.99}, {y, 0, 0.99}, PlotPoints -> 60]
```



```
Out[26]:= - SurfaceGraphics -
```

Визуализация функции ошибки аппроксимации:

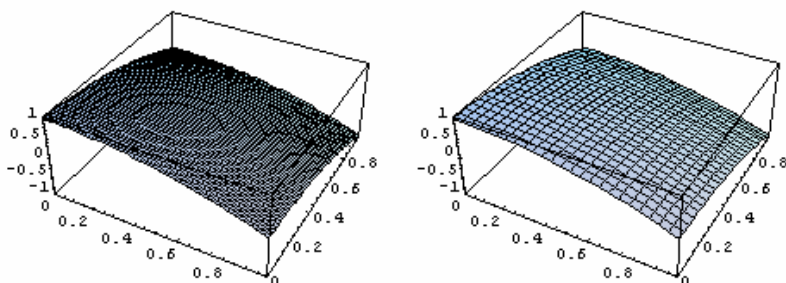
```
In[27]:= Plot3D[f1 - fa1, {x, 0, 0.99}, {y, 0, 0.99}, PlotPoints -> 60]
```



```
Out[27]:= - SurfaceGraphics -
```

Ниже приведены для сопоставления графический образ сигнала (слева) и его аппроксимация (справа).

```
In[29]:= Show[GraphicsArray[{Out[26], pf}]]
```



```
Out[29]:= - GraphicsArray -
```

Пример 14. Аппроксимировать сигнал $f(x, y) = 10 \sin\left(\frac{\pi(x+y)}{10}\right)$ в

области изменения аргументов $\begin{pmatrix} 0 \leq x < 10, \\ 0 \leq y < 10 \end{pmatrix}$, используя сетку с

шагом $h_x = h_y = 1$. Программа аппроксимации с результатами ее работы приведена ниже. Так как методика аппроксимации

аналогична рассмотренной в примере 13, комментарии не приводятся.

```
In[1]:= v[x_, i_, m_, T_] := If[(i - 1) T / m ≤ x < i T / m, 1, 0];
```

```
In[2]:= vx = v[x, i, 10, 10];
```

```
In[3]:= vy = v[y, j, 10, 10];
```

```
In[4]:= vxy = vx vy;
```

```
In[5]:= vxy1 := Table[vxy, {i, 10}, {j, 10}];
```

```
In[6]:= f[x_, y_] := 10 Sin[π * (x + y) / 10];
```

```
In[7]:= f1 = f[x, y];
```

```
h := 1; T := 10;
```

```
In[9]:= Fv[i_, j_] :=  $\frac{1}{h^2} \int_{(i-1)h}^{ih} \int_{(j-1)h}^{jh} f1 dy dx$ ;
```

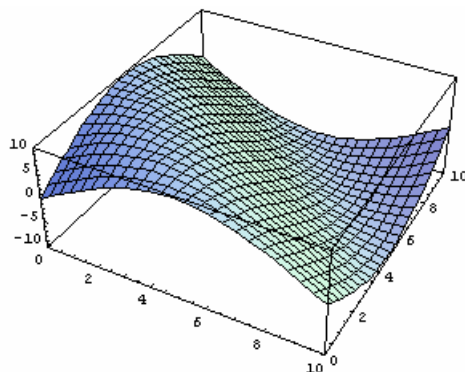
```
In[10]:= FV = Table[Fv[i, j], {i, 10}, {j, 10}];
```

```
In[11]:= N[FV, 3] // MatrixForm
```

Out[11]//MatrixForm=

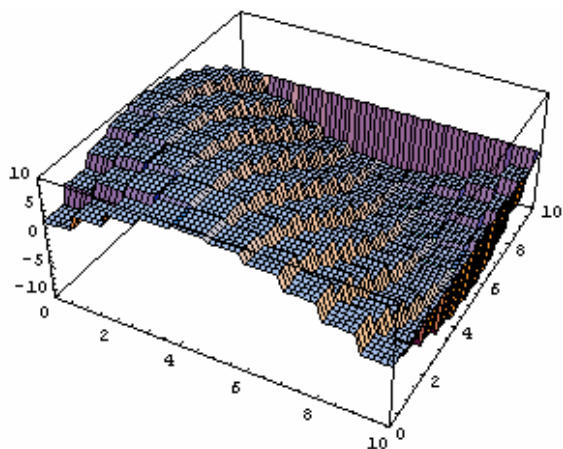
3.06	5.83	8.02	9.43	9.92	9.43	8.02	5.83	3.06	0
5.83	8.02	9.43	9.92	9.43	8.02	5.83	3.06	0	-3.06
8.02	9.43	9.92	9.43	8.02	5.83	3.06	0	-3.06	-5.83
9.43	9.92	9.43	8.02	5.83	3.06	0	-3.06	-5.83	-8.02
9.92	9.43	8.02	5.83	3.06	0	-3.06	-5.83	-8.02	-9.43
9.43	8.02	5.83	3.06	0	-3.06	-5.83	-8.02	-9.43	-9.92
8.02	5.83	3.06	0	-3.06	-5.83	-8.02	-9.43	-9.92	-9.43
5.83	3.06	0	-3.06	-5.83	-8.02	-9.43	-9.92	-9.43	-8.02
3.06	0	-3.06	-5.83	-8.02	-9.43	-9.92	-9.43	-8.02	-5.83
0	-3.06	-5.83	-8.02	-9.43	-9.92	-9.43	-8.02	-5.83	-3.06

```
In[12]:= Plot3D[f1, {x, 0, 10}, {y, 0, 10}]
```



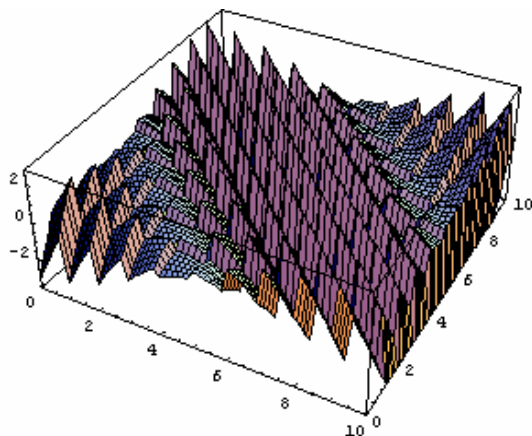
Out[12]= - SurfaceGraphics -

```
In[13]:= fa :=  $\sum_{i=1}^{10} \sum_{j=1}^{10} FV[[i, j]] \text{ vxy1}[[i, j]];$ 
```



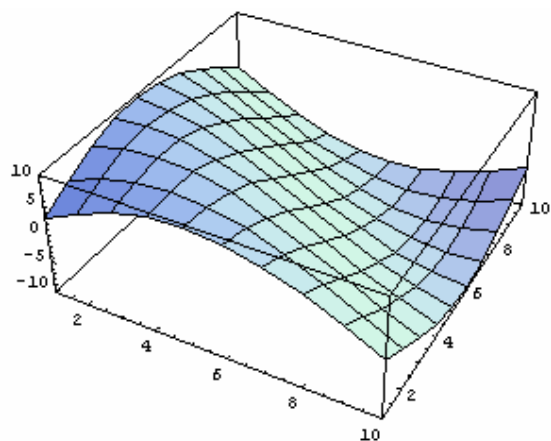
```
Out[14]= - SurfaceGraphics -
```

```
In[15]:= Plot3D[f1 - fa, {x, 0, 10}, {y, 0, 10}, PlotPoints -> 60]
```



```
Out[15]= - SurfaceGraphics -
```

```
In[16]:= ListPlot3D[FV]
```

Out[16]= - SurfaceGraphics -

6. УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

В этом разделе приведено пять упражнений по аппроксимации различных сигналов полиномами с различными системами базисных функций. Постановка задач дополняется советами и рекомендациями о порядке выполнения упражнений. Если в ходе самостоятельной работы встретятся непреодолимые трудности, в конце работы содержатся полные программы с решениями задач, которые позволят шаг за шагом выполнить все необходимые этапы аппроксимации. Авторы, однако, настоятельно рекомендуют сначала попробовать свои силы и самостоятельно достичь цели.

Упражнение 1. Аппроксимация непрерывного сигнала

Постановка задачи:

Аппроксимировать сигнал $x(t) = \frac{1}{\sqrt{1+5t^2}} \sin(2\pi t)$ на интервале изменения аргумента ($0 \leq t < 2$) степенными полиномами 5, 7 и 10 порядков

Указания:

1.1. Сформировать базисную систему функций степенного типа:

$$s(t, i, T) = \left(\frac{t}{T} \right)^{i-1} \text{ для } n=10 \text{ и } T=2.$$

1.2. Определить операционную матрицу аппроксимирующего степенного спектра W с элементами вида:

$$w_{ij} = \int_0^T s(t, i, T) \cdot s(t, j, T) dt$$

1.3. Определить вид аппроксимируемого сигнала.

1.4. Визуализировать сигнал с помощью команды Print.

1.5. Определить операционный вектор спектра Q с элементами вида:

$$q_i = \int_0^T x(t) \cdot s(t, i, T) dt .$$

1.6. Определить вектор аппроксимирующего степенного спектра сигнала по формуле: $\bar{X} = W^{-1} \cdot \bar{Q}$.

1.7. Сформировать в векторно-матричной форме аппроксимирующий полином.

- 1.8. Сформировать функцию ошибки аппроксимации.
- 1.9. Визуализировать аппроксимацию сигнала и функцию ошибки аппроксимации.
- 1.10. Повторить шаги 1.1-1.9 для порядков полиномов 5 и 7.
- 1.11. Сравнить качество аппроксимации полиномами различных порядков.

Упражнение 2. Аппроксимация непрерывного сигнала

Постановка задачи:

Аппроксимировать сигнал $x(t) = e^{-10(t-1)^2} + e^{-10(t+1)^2} - e^{-20t^2}$ на интервале $(-2 \leq t < 2)$ изменения аргумента, используя блочно-импульсную систему функций с шагом $h=0.04$.

Указания:

- 2.1. Определить сигнал и визуализировать его с помощью команды Plot.
- 2.2. Сформировать систему блочно-импульсных базисных функций, определенных на заданном диапазоне изменения аргумента.
- 2.3. Определить вектор блочно-импульсного спектра сигнала с

$$\text{элементами: } X_i = \frac{1}{h} \int_{a+(i-1)h}^{a+ih} x(t) dt.$$

- 2.4. Сформировать аппроксимирующий полином вида:

$$x_a(t) = \bar{X}^T \cdot \bar{S}(t).$$

- 2.5. Визуализировать аппроксимацию сигнала.
- 2.6. Сформировать функцию ошибки аппроксимации.
- 2.7. Визуализировать элементы вектора блочно-импульсного спектра сигнала с помощью команды ListPlot[X] в системе координат {x,i}.
- 2.8. Сформировать таблицу отсчетов $\{t_i, X_i\}$.
- 2.9. Визуализировать элементы вектора блочно-импульсного спектра сигнала в системе координат $\{x, t\}$.
- 2.10. Совместить результат п. 2.9 с графиком сигнала.
- 2.11. Повторить п.п. 2.1 – 2.10, изменив шаг по аргументу ($h=0.08$) и порядок базисной системы функций ($m=50$). Сравнить результаты аппроксимации.

Упражнение 3. Аппроксимация фазового портрета сигнала

Постановка задачи:

Аппроксимировать фазовый портрет сигнала $y(t) = 5e^{-0.5t} \cdot \sin(2\pi t)$ на интервале $(0 \leq t < 1)$ изменения аргумента, используя дискретизацию сигнала с шагом по аргументу $h=0.025$ и систему базисных функций 7 порядка на основе смещенных полиномов Лежандра.

Указания:

- 3.1. Определить сигнал.
- 3.2. Определить первую производную по времени от сигнала.
- 3.3. Построить фазовый портрет сигнала, используя команду ParametricPlot и , имея в виду, что под фазовым портретом сигнала понимают зависимость между сигналом и его первой производной, причем аргумент сигнала выступает как параметрический аргумент: $y = f(y')$, $y = \varphi(t)$, $y' = \psi(t)$.
- 3.4. Определить систему базисных функций, используя встроенную функцию системы LegendreP, преобразовав аргумент так, чтобы интервал ортогональности полиномов Лежандра совпал с интервалом аппроксимации $(0, 1)$.
- 3.5. Визуализировать систему базисных функций.
- 3.6. Дискретизировать сигналы функции и первой производной и сформировать таблицы дискретизированных значений сигналов и аргумента.
- 3.7. Визуализировать дискретизированные сигналы, используя команду ListPlot/
- 3.8. Построить таблицы значений дискретизированных сигналов.
- 3.9. Сформировать операционную матрицу спектра W .
- 3.10. Определить операционные векторы спектров сигналов.
- 3.11. Определить аппроксимирующие спектры Лежандра для сигнала и производной.
- 3.12. Построить аппроксимации сигналов.
- 3.13. Визуализировать аппроксимации сигналов, совместив их на одном графике с аппроксимируемыми сигналами.
- 3.14. Построить аппроксимацию фазового портрета и сравнить его с заданным фазовым портретом.
- 3.15. Построить параметрический график ошибок аппроксимации сигнала и его первой производной.

Упражнение 4. Аппроксимация фазового портрета сигнала

Постановка задачи:

Аппроксимировать фазовый портрет сигнала $y(t) = 5e^{-0.5t} \cdot \sin(2\pi t)$ на интервале $(0 \leq t < 1)$ изменения аргумента, используя систему блочно-импульсных базисных функций 50 порядка.

Указания:

- 4.1. Определить сигнал.
- 4.2. Определить первую производную по времени от сигнала.
- 4.3. Построить фазовый портрет сигнала, используя команду ParametricPlot и, имея в виду, что под фазовым портретом сигнала понимают зависимость между сигналом и его первой производной, причем аргумент сигнала выступает как параметрический аргумент: $y = f(y')$, $y = \varphi(t)$, $y' = \psi(t)$.
- 4.4. Определить блочно-импульсную систему базисных функций.
- 4.5. Определить векторы блочно-импульсных спектров сигнала и его первой производной.
- 4.6. Определить аппроксимации сигналов.
- 4.7. Визуализировать аппроксимации сигналов, совместив их на одном графике с аппроксимируемыми сигналами.
- 4.8. Построить аппроксимацию фазового портрета и сравнить его с заданным фазовым портретом.

Упражнение 5. Аппроксимация пространственной кривой**Постановка задачи:**

Аппроксимировать пространственную кривую, заданную параметрическими составляющими:

$$\begin{cases} fx = e^{-0.5t} \cdot \sin(4\pi t), \\ fy = e^{-t} \cdot \cos(4\pi t), \\ fz = t \end{cases}$$

на интервале $(0 \leq t < 1)$ изменения аргумента, используя систему блочно-импульсных базисных функций 100 порядка.

Указания:

- 5.1. Определить функции параметрических составляющих кривой.
- 5.2. Визуализировать кривую, используя команду ParametricPlot3D.
- 5.3. Визуализировать параметрические составляющие кривой, используя команду Plot.
- 5.4. Определить базисную систему функций и ее параметры (h, m) .

- 5.5. Определить векторы блочно-импульсных спектров параметрических составляющих кривой.
- 5.6. Сформировать аппроксимации параметрических составляющих кривой.
- 5.7. Визуализировать аппроксимации параметрических составляющих кривой и сравнить их с аппроксимируемыми путем совмещения на одном графике.
- 5.8. Построить аппроксимацию пространственной кривой и сравнить ее с исходной путем совмещения на одном графике.
- 5.9. Визуализировать функцию ошибки аппроксимации в параметрической форме.

Приложение 1. Первоначальные сведения по системе Mathematica фирмы Wolfram Research Inc.

Система Mathematica фирмы Wolfram Research Inc. является мощной универсальной системой компьютерной алгебры, ориентированной на символические вычисления. Однако, она великолепно доводит до числового результата сложнейшие вычисления. Первоначальные навыки работы в системе можно получить из электронного помощника (Help) , выбрав в меню Help строку Tutorial (версия 5.2), или обратившись к самоучителю [14] или к книге [15].

1. Некоторые особенности языка и синтаксиса системы Mathematica

Арифметические операции выполняются аналогично клавишным калькуляторам:

Операция	Символ (команда)
сложение	+
вычитание	-
скалярное умножение	* (или пробел между сомножителями)
векторно-матричное умножение	.
деление	/
возведение в степень	^
Круглые скобки используются исключительно для изменения естественного порядка выполнения операций	()
Квадратные скобки используются только для ввода аргументов функций и других выражений	[]
Фигурные скобки используются только для задания матриц, векторов, списков, множеств и других многоэлементных объектов	{ }
Команда присвоения значения b величине a:	a=b
Команда определения функции или выражения	y:=f[x]
Задание уравнения	= =
Команда заменить в выражении f величину x на w:	f /.x → w

Операция	Символ (команда)
Команда выполнения программы или ее части:	Shift+Enter
Обращение к только что полученному результату:	% далее Shift+Enter
Встроенные функции и выражения начинаются с заглавной буквы:	Solve, Integrate и т.д.
В сложных встроенных функциях и выражениях составные части выражения начинаются с заглавных букв, а части вводятся без пробела между ними:	FindRoot, ParametricPlot и т.д.
Краткая справка по встроенной функции или выражению:	?Solve или ??Solve, далее Shift+Enter
Написание числа перед буквенным выражением с пробелом или без него обозначает	умножение этого числа на выражение
Написание числа после буквенного выражения без пробела воспринимается системой как	новый символ
Точка с запятой, следующая за введенным определением функции или выражения, предотвращает вывод результата, который может быть промежуточным или громоздким:	;
После команды Shift+Enter система автоматически присваивает введенным командам и выражениям имена In[1], In[2] и т. д., а результатам вычислений - имена Out[1], Out[2] и т.д.	Shift+Enter
Выражения типа Out[n] могут быть использованы как имена соответствующих функций и выражений в последующих вычислениях	Out[n]

2. Примеры выполнения различных операций

Арифметические:

In[1]:= 5 * (2.34 + 1.45) ^ 2 / 2.3

Out[1]= 31.2263

Алгебраические:

- Раскрыть выражение $(a + b)^4$:

In[2]:= **Expand**[(a + b) ^4]

Out[2]= $a^4 + 4 a^3 b + 6 a^2 b^2 + 4 a b^3 + b^4$

- Разложить на множители выражение $a^3 - b^3$:

In[3]:= **Factor**[a ^3 - b ^3]

Out[3]= $(a - b) (a^2 + a b + b^2)$

- Решить алгебраическое уравнение $ax^2 + bx + c = 0$:

In[4]:= **Solve**[a * x ^2 + b * x + c == 0, x]

Out[4]= $\left\{ \left\{ x \rightarrow \frac{-b - \sqrt{b^2 - 4 a c}}{2 a} \right\}, \left\{ x \rightarrow \frac{-b + \sqrt{b^2 - 4 a c}}{2 a} \right\} \right\}$

- Упростить тригонометрическое выражение $\sin(x)\cos(x)$:

In[5]:= **TrigReduce**[Sin[x] * Cos[x]]

Out[5]= $\frac{1}{2} \sin[2 x]$

- Упростить алгебраическое выражение

$$\frac{a^3 + 3a^2b + 3ab^2 + b^3}{a^2 - b^2}:$$

In[6]:= **Simplify**[$\frac{a^3 + 3 a^2 * b + 3 a * b^2 + b^3}{a^2 - b^2}$]

Out[6]= $\frac{(a + b)^2}{a - b}$

- Обратить матрицу $A = \begin{bmatrix} 6 & 4 & 1 \\ 4 & 6 & 4 \\ 1 & 4 & 6 \end{bmatrix}$:

```
In[7]:= A := {{6, 4, 1}, {4, 6, 4}, {1, 4, 6}};
```

```
B = Inverse[A]
```

```
Out[8]= {{2/5, -2/5, 1/5}, {-2/5, 7/10, -2/5}, {1/5, -2/5, 2/5}}
```

- Представить полученную матрицу в привычной математической записи:

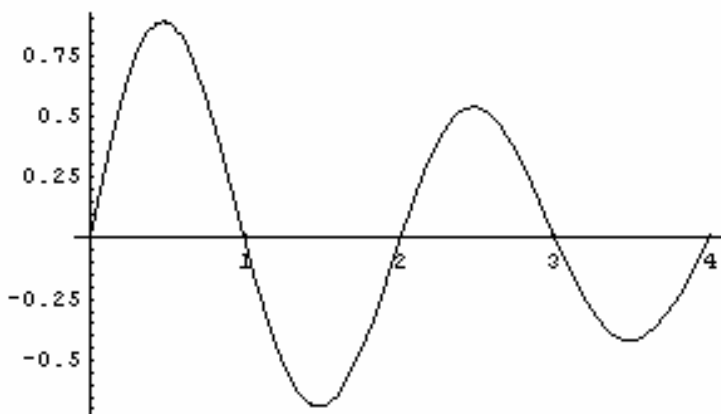
```
In[9]:= % // MatrixForm
```

```
Out[9]//MatrixForm=
```

$$\begin{pmatrix} \frac{2}{5} & -\frac{2}{5} & \frac{1}{5} \\ -\frac{2}{5} & \frac{7}{10} & -\frac{2}{5} \\ \frac{1}{5} & -\frac{2}{5} & \frac{2}{5} \end{pmatrix}$$

- Нарисовать график функции $f(x) = e^{-x/4} \sin(\pi x)$ на интервале изменения аргумента $0 \leq x < 4$:

```
In[10]:= Plot[Exp[-x/4] * Sin[Pi * x], {x, 0, 4}]
```

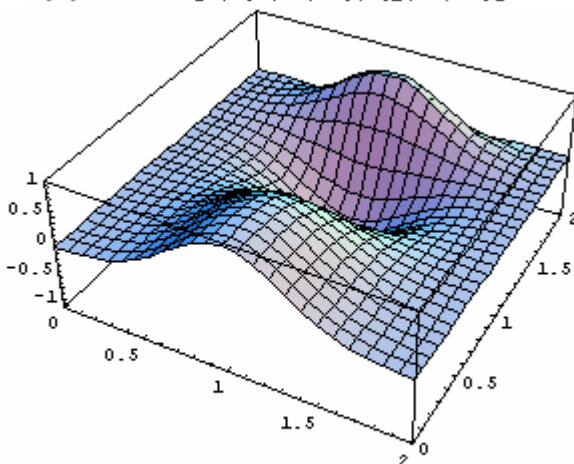


```
Out[10]= - Graphics -
```

- Отобразить поверхность, заданную функцией $z = e^{-5(x-1)^2} \cos(2y^2)$, в области, ограниченной координатами $0 \leq x < 2, 0 \leq y < 2$:

```
In[11]:= z := Exp[-5 (x - 1) ^ 2] * Cos[2 y ^ 2];
```

```
In[12]:= Plot3D[z, {x, 0, 2}, {y, 0, 2}]
```



```
Out[12]= - SurfaceGraphics -
```

- Построить таблицу значений функции $y(x) = \sin(x^3)$ при значениях аргумента, равномерно распределенных на отрезке (0, 2) с шагом 0.1:

```
In[13]:= y := Sin[x^3];
```

```
In[14]:= Y = Table[{0.1 i, y /. x -> i * 0.1}, {i, 0, 20}]
```

```
Out[14]= {{0, 0}, {0.1, 0.001}, {0.2, 0.00799991},
{0.3, 0.0269967}, {0.4, 0.0639563}, {0.5, 0.124675},
{0.6, 0.214324}, {0.7, 0.336314}, {0.8, 0.489922},
{0.9, 0.666124}, {1., 0.841471}, {1.1, 0.971386},
{1.2, 0.987669}, {1.3, 0.810258}, {1.4, 0.3872},
{1.5, -0.231294}, {1.6, -0.815971}, {1.7, -0.979945},
{1.8, -0.436033}, {1.9, 0.544518}, {2., 0.989358}}
```

- Вывести полученный массив вместе со значениями аргумента в привычной табличной форме:

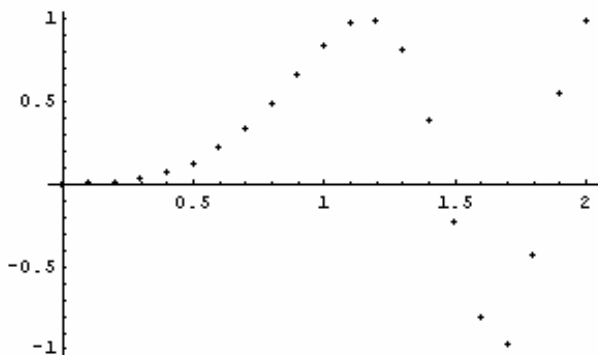
```
In[15]:= % // TableForm
```

Out[15]/TableForm=

0	0
0.1	0.001
0.2	0.00799991
0.3	0.0269967
0.4	0.0639563
0.5	0.124675
0.6	0.214324
0.7	0.336314
0.8	0.489922
0.9	0.666124
1.	0.841471
1.1	0.971386
1.2	0.987669
1.3	0.810258
1.4	0.3872
1.5	-0.231294
1.6	-0.815971
1.7	-0.979945
1.8	-0.436033
1.9	0.544518
2.	0.989358

- Отобразить полученный массив отсчетов на графике:

In[16]:= **ListPlot[Y, PlotStyle → PointSize[0.015]]**



Out[16]= - Graphics -

Приложение 2. Тексты программ иллюстративных примеров, рассмотренных в пособии.

Примеры 1, 2 и 3: Формирование системы степенных базисных функций и аппроксимация экспоненциального и косинусоидального сигналов

$$ss[t_, T_, i_, m_] := \left(\frac{t}{T}\right)^{\text{If}[i/m < 1, \frac{i}{m}, i-m]};$$

```
s = ss[t, 1, i, 4];
```

```
S = Table[s, {i, 8}]
```

```
Plot[Evaluate[S], {t, 0.001, 1}, AxesLabel -> {t, si}, PlotLabel -> Basic functions]
```

$$W = \int_0^1 \text{Table}[S[[i]] * S[[j]], \{i, 8\}, \{j, 8\}] dt;$$

```
%//MatrixForm
```

$$Q = \int_0^1 \text{Table}[S[[i]] * e^{-t}, \{i, 8\}] dt;$$

```
Q1 = N[%]
```

```
X = Inverse[W].Q1
```

```
xa = X.S
```

```
Plot[{e^{-t}, xa}, {t, 0, 1}, AxesLabel -> {t, "e^{-t}, xa"}]
```

```
Plot[e^{-t} - xa, {t, 0, 1}, AxesLabel -> {t, "error"},
```

```
PlotLabel -> approximation error]
```

```
Q2 = Chop[Integrate[Table[S[[i]] * Cos[3.14 t^2], {i, 8}], {t, 0, 1}]]
```

```
Y = Inverse[W].Q2
```

```
ya = Y.S
```

```
Plot[{Cos[3.14 t^2], ya}, {t, 0, 1}]
```

```
Plot[Cos[3.14 t^2] - ya, {t, 0, 1}]
```

Пример 4: Аппроксимация затухающего сигнала в экспоненциальной системе базисных функций

```
f:=Exp[-t]*Sin[Pi*t];
Plot[f,{t,0,1.5}]
ss[t_,i_,m_]:=Table[Exp[-t*(i-1)],{i,m}];
s1=ss[t,i,5];
s2=ss[t,j,5];

W=N[Table[Integrate[s1[[i]]*s2[[j]],{t,0,Pi/2}],{i,5},
{j,5}]];
W//MatrixForm

Q:=Chop[N[Table[Integrate[f*s1[[i]],{t,0,Pi/2}],{i,5}]]];
Q
F:=Inverse[W].Q;
F
s1
fa=F.s1
Plot[{f,fa},{t,0,1.5}]
Plot[f-fa,{t,0,1.5}]
```

Пример 5. Аппроксимация затухающей синусоиды тригонометрическим полиномом"

```
f:=Exp[-t]*Sin[Pi*t];
Plot[f,{t,0,Pi/2}]
ss[t_,i_,m_]:=Table[If[i≤m,Cos[4t*(i-1)],Sin[4t*(i-
m)]],{i,2m-1}];
s1=ss[t,i,5];s2=ss[t,j,5];

W=Chop[N[Table[Integrate[s1[[i]]*s2[[j]],{t,0,Pi/2}],{
i,9},{j,9}]]];

Q=Chop[N[Table[Integrate[f*s1[[i]],{t,0,Pi/2}],{i,9}]]];
Q
F:=Inverse[W].Q;
fa=F.s1;
Plot[{f,fa},{t,0,Pi/2}]
Plot[f-fa,{t,0,Pi/2}]
```

Пример 6. Аппроксимация непрерывного сигнала с помощью системы полиномов Лежандра

```
SL[x_]:=Evaluate[Table[LegendreP[n,x],{n,0,15}]]];
sl=Simplify[SL[x/90]];
Plot[Evaluate[sl],{x,-90,90}]
f=Exp[-0.5(x/90)^2]-0.2Exp[-5(x/90)^2];
Plot[f,{x,-90,90},PlotRange->All]
W:=Table[Integrate[sl[[i]]*sl[[j]],{x,-
90,90}],{i,16},{j,16}];
W1:=Table[Integrate[sl[[i]]*sl[[j]],{x,-
90,90}],{i,5},{j,5}]/MatrixForm
W1
Q:=Table[Integrate[sl[[i]]*f,{x,-90,90}],{i,16}];
Q
F=Inverse[W].Q;
F
fa=F.sl;
Plot[fa,{x,-90,90},PlotRange->All]
Plot[fa-f,{x,-90,90},PlotRange->All]
SL2[x_]:=Evaluate[Table[LegendreP[n,x],{n,0,12,2}]]];
sl2=Simplify[SL2[x/90]];
Plot[Evaluate[sl2],{x,-90,90}]
W2:=Table[Integrate[sl2[[i]]*sl2[[j]],{x,-
90,90}],{i,7},{j,7}];
W2//MatrixForm
Q2:=Table[Integrate[sl2[[i]]*f,{x,-90,90}],{i,7}];
Q2
F2=Inverse[W2].Q2;
F2
fa2=F2.sl2;
Plot[fa2,{x,-90,90},PlotRange->All]
Plot[fa2-f,{x,-90,90},PlotRange->All]
F
```

Пример 7. Аппроксимация сигнала, заданного системой отсчетов

```

h=0.1;r=10;
tt:=Table[i*h,{i,0,r-1}]
tt
T= 1;

x:=Cos[(t/T)^2]+0.1Sin[7 t/T];
p1=Plot[x,{t,0,T},PlotRange->All]
xt:=Table[x/.t->tt[[i]],{i,1,r}]
xt

p2=ListPlot[Table[{tt[[i]],xt[[i]]},{i,1,r}],PlotStyle
->PointSize[0.015]]
m= 5;k= 10;

s[t_, T_, i_] := (t/T)^(i-1);

si= s[t, 1, i];
S:= Table[si, {i, 1, m}]
S
w:=Table[si/.t->tt[[j]],{j,1,k},{i,1,m}];
w//MatrixForm
Q=xt
X=PseudoInverse[w].Q
xa=X.S
p3=Plot[xa,{t,0,T}];
Show[p2,p3]
S1:=Table[s[t,T,i],{i,3}]
w1:=Table[si/.t->tt[[j]],{j,k},{i,3}];
w1//MatrixForm
X1=PseudoInverse[w1].Q
xa1=X1.S1
p4=Plot[xa1,{t,0,T}]
Show[p4,p2]
h1=0.2;r1=5;
tt1:=Table[i*h1,{i,0,r1-1}]
tt1
xt1:=Table[x/.t->tt1[[i]],{i,1,r1}]
xt1

```



```

p5=ListPlot[Table[{tt1[[i]],xt1[[i]]},{i,1,r1}],PlotSt
yle→PointSize[0.017]]
k=5;
w1:=Table[si/.t→tt1[[j]],{j,1,k},{i,1,m}];
w1//MatrixForm
Q1=xt1
X1=PseudoInverse[w1].Q1
xa1=X1.S
p6=Plot[xa1,{t,0,T}];
p7=Show[p5,p6]
m1=7;
w2:=Table[si/.t→tt1[[j]],{j,1,k},{i,1,m1}];
w2//MatrixForm
X2=PseudoInverse[w2].Q1
S2:=Table[si,{i,1,m1}]
xa2=X2.S2
p8=Plot[xa2,{t,0,T}];
p9=Show[p7,p8]

```

Пример 8. Аппроксимация сигнала, заданного в параметрической форме"

```

y[t_,ω_]:=Exp[-t]*Sin[ω*t];
x[t_,ω_]:=Exp[-2t]*Cos[ω*t];
y1=y[t,2π];x1=x[t,2π];
PP=ParametricPlot[{x1,y1},{t,0,2},PlotRange→All]
Plot[y1,{t,0,2}]
Plot[x1,{t,0,2}]
m=7;T=2;

```

$$s[t_, n_, T_] := \left(\frac{t}{T}\right)^{n-1};$$

```
S:=Table[s[t,n,T],{n,1,m+1}]
```

```
S
```

```
w1=Transpose[{S}].{S};
```

$$W = \int_0^T w1 dt;$$

```
N[%,4]//MatrixForm
```

$$Qx = N\left[\int_0^T x1 * S dt\right]$$

$$Qy = N\left[\int_0^T y1 * S dt\right]$$

```

X=Inverse[W].Qx
Y=Inverse[W].Qy
x1a=X.S
y1a=Y.S
Plot[{y1,y1a},{t,0,2}]
Plot[{x1,x1a},{t,0,2}]
PPa=ParametricPlot[{x1a,y1a},{t,0,2},PlotRange→All]
PPe=ParametricPlot[{x1-x1a,y1-
y1a},{t,0,2},PlotRange→All]
h=0.05;
X1=Table[x[i*h,2π},{i,0,40}];
Y1=Table[y[i*h,2π},{i,0,40}];
Tt=Table[i*h,{i,0,40}];

Px=ListPlot[X1,PlotStyle→PointSize[0.015],PlotRange→A
ll]
Py=ListPlot[Y1,PlotStyle→PointSize[0.015],PlotRange→A
ll]
Py2=ListPlot[Table[{Tt[[k]],Y1[[k]]},{k,41}],PlotStyle
→PointSize[0.015],PlotRange→All]
Px2=ListPlot[Table[{Tt[[k]],X1[[k]]},{k,41}],PlotStyle
→PointSize[0.015],PlotRange→All]
Pxy=ListPlot[Table[{X1[[k]],Y1[[k]]},{k,41}],PlotStyle
→PointSize[0.015],PlotRange→All]
Show[{PP,Pxy}]
m=7;T=2;
w:=Table[S[[p]]/.t→Tt[[r]],{p,m+1},{r,41}];
q1:=X1;
q2:=Y1;
X=PseudoInverse[Transpose[w]].q1;
Y=PseudoInverse[Transpose[w]].q2;
X
Y
xa=X.S;
ya=Y.S;
Pxa=Plot[{x1,xa},{t,0,2}]
Pya=Plot[{y1,ya},{t,0,2}]
Pza=ParametricPlot[{xa,ya},{t,0,2},PlotRange→All]
Show[{PP,Pza}]
Show[{Pza,Pxy}]
ParametricPlot[{x1-xa,y1-ya},{t,0,2},PlotRange→All]

```

Пример 9. Аппроксимация сигнала, заданного параметрическими составляющими в табличной форме системой полиномов Лежандра

```
SL[x_]:=Evaluate[Table[LegendreP[n,x],{n,0,12,2}]];
sl=Simplify[SL[x/90]];
Plot[Evaluate[sl],{x,-90,90}]
f=Exp[-0.5(x/90)^2]-0.2Exp[-5(x/90)^2];
h=9;
fd=Table[{k*h,f/.x->k*h},{k,-10,10}];
ListPlot[fd,PlotStyle->PointSize[0.015]]
fd
fd//TableForm

wd:=Table[sl[[i]]/.x->-90+(k-1)*h,{i,7},{k,21}];
qd:=Table[f/.x->-90+(k-1)*h,{k,21}];
qd
Fd=PseudoInverse[Transpose[wd]].qd;
Fd
fda=Fd.sl;
Plot[fda,{x,-90,90},PlotRange->All]
Show[{Out[7],Out[17]},PlotRange->All]
Plot[f-fda,{x,-90,90}]
```

Пример 10. Аппроксимация синусоидального сигнала с использованием локальных базисных систем на основе смещенных полиномов Лежандра

```
v[t_,m_,h_]:=Table[If[(i-1)*h<=t<i*h,1,0],{i,m}];
w[t_,m_,h_]:=Table[(2t/h+1-2i)*v[t,m,h][[i]],{i,m}];
u[t_, m_, h_] :=
  Table[ $\left(6 \left(\frac{t}{h}\right)^2 - \frac{6 (2 i - 1) * t}{h} + 6 i^2 - 6 i + 1\right) * v[t, m, h][[i]], \{i, m\}];$ 
v[t, m, h][[i]], {i, m}];
V=v[t,5,0.2];
W=w[t,5,0.2];
U=u[t,5,0.2];
h=0.2;m=5;
f[t_] := Sin[ $\frac{\pi * t}{2}$ ];
```

```

sp0 := Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} f[t] dt$ , {i, m}];
sp1 := Table[ $\frac{3}{h} * \int_{(i-1)*h}^{i*h} f[t] * (2t/h + 1 - 2i) dt$ , {i, m}];

sp2 :=
Table[
 $\frac{5}{h} * \int_{(i-1)*h}^{i*h} f[t] * \left( 6 \left( \frac{t}{h} \right)^2 - \frac{6(2i-1)*t}{h} + 6i^2 - 6i + 1 \right) dt$ ,
{i, m}];
sp0
sp1
sp2
fa0:=Evaluate[sp0.V];
fa1:=Evaluate[sp0.V+sp1.W];
fa2:=Evaluate[sp0.V+sp1.W+sp2.U];
Plot[{f[t],fa0},{t,0,1}]
Plot[{f[t],fa1},{t,0,1}]
Plot[{f[t],fa2},{t,0,1}]
Plot[f[t]-fa0,{t,0,1}]
Plot[f[t]-fa1,{t,0,1}]
Plot[f[t]-fa2,{t,0,1}]

```

Пример 11. Аппроксимация W-образного сигнала с помощью локальных базисных систем на основе смещенных полиномов Лежандра

```

v[t_,m_,h_]:=Table[If[(i-1)*h<=t<i*h,1,0],{i,m}];
w[t_,m_,h_]:=Table[(2t/h+1-2i)*v[t,m,h][[i]],{i,m}];
u[t_,m_,h_]:=
Table[ $\left( 6 \left( \frac{t}{h} \right)^2 - \frac{6(2i-1)*t}{h} + 6i^2 - 6i + 1 \right) * v[t,m,h][[i]]$ , {i, m}];
V=v[t,20,0.1];
W=w[t,20,0.1];
U=u[t,20,0.1];
h=0.1;m=20;
f[t_] := 5 Cos[ $\pi * t$ ] + 6 e-15 (t-1)2;

```

```

sp0 := Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} f[t] dt$ , {i, m}];
sp1 := Table[ $\frac{3}{h} * \int_{(i-1)*h}^{i*h} f[t] * (2t/h + 1 - 2i) dt$ , {i, m}];

sp2 :=
Table[
 $\frac{5}{h} * \int_{(i-1)*h}^{i*h} f[t] * \left( 6 \left( \frac{t}{h} \right)^2 - \frac{6(2i-1)*t}{h} + 6i^2 - 6i + 1 \right) dt$ ,
{i, m}];
sp0
sp1
sp2
fa0:=Evaluate[sp0.V];
fa1:=Evaluate[sp0.V+sp1.W];
fa2:=Evaluate[sp0.V+sp1.W+sp2.U];
Plot[{f[t],fa0},{t,0,2}]
Plot[{f[t],fa1},{t,0,2}]
Plot[{f[t],fa2},{t,0,2}]
Plot[f[t]-fa0,{t,0,2}]
Plot[f[t]-fa1,{t,0,2}]

```

Пример 12. Двумерные базисные системы локального типа

```

m:=10;n:=10;Tx:=1;Ty:=1;
v[x_,i_,m_,T_]:=If[(i-1) T/m<=x<i T/m,1,0];
vx=v[x,i,m,Tx];
vy=v[y,j,n,Ty];
v=vx*vy;
Vxy=Table[v,{i,m},{j,n}];

Plot3D[Vxy[[5,5]],{x,0.35,0.55},{y,0.35,0.55},PlotPoints->40]

w[x_,i_,m_,T_]:=1-2i+ $\frac{2m*x}{T}$ ;
wx=w[x,i,m,Tx]*v;
Wx=Table[w,{i,m},{j,n}];
wy=w[y,j,n,Ty]*v;
Wy=Table[w,{i,m},{j,n}];

```

```
Plot3D[Wx[[5,5]],{x,0.35,0.55},{y,0.35,0.55},PlotPoints→40]
```

```
Plot3D[Wy[[5,5]],{x,0.35,0.55},{y,0.35,0.55},PlotPoints→40]
```

```
Plot3D[Vxy[[2,2]]+Wx[[4,4]]+Wy[[6,6]],{x,0.05,0.65},{y,0.05,0.65},PlotPoints→40]
```

Пример 13. Аппроксимация фрагмента эллипсоида с помощью двумерных локально-импульсных базисных систем

```
v[x_,i_,m_,T_]:=If[(i-1) T/m≤x<i T/m,1,0];
vx=v[x,i,10,1];
vy=v[y,j,10,1];
Vxy=vx vy;
vxy1:=Table[Vxy,{i,1,10},{j,1,10}];
w[x_,i_,m_,T_]:= (1-2i+2m x/T) If[(i-1) T/m≤x<i T/m,1,0];
wx=w[x,i,10,1] vy;
wy=w[y,j,10,1] vx;
zx:=Table[wx,{i,1,10},{j,1,10}];
zy:=Table[wy,{i,1,10},{j,1,10}];
f[x_,y_] := 1-x2-y2;
f1=f[x,y];
Fv[i_,j_] := 100 ∫(i-1)/10i/10 ∫(j-1)/10j/10 f1 dy dx;
FV=Table[Fv[i,j],{i,1,10},{j,1,10}];
ListPlot3D[%]
fa = ∑i=110 ∑j=110 FV[[i,j]] vxy1[[i,j]];
Plot3D[fa,{x,0,0.99},{y,0,0.99},PlotPoints→60]
Plot3D[fa-f1,{x,0,0.99},{y,0,0.99},PlotPoints→60]
Fwx[i_,j_] := 300 ∫(i-1)/10i/10 ∫(j-1)/10j/10 f1 wx1 dy dx;
wx1=w[x,i,10,1] ;
wy1=w[y,j,10,1] ;
Fwy[i_,j_] := 300 ∫(j-1)/10j/10 ∫(i-1)/10i/10 f1 wy1 dx dy;
FWX=Table[Fwx[i,j],{i,1,10},{j,1,10}];
```

```

FWY=Table[Fwy[i,j],{i,1,10},{j,1,10}];
FWX
FWY
Rx=N[%%]
Ry=N[%%]
ListPlot3D[Rx]
ListPlot3D[Ry]
fa1 =

$$\sum_{i=1}^{10} \sum_{j=1}^{10} (FV[[i,j]] + Rx[[i,j]] \, zx[[i,j]] +$$


$$Ry[[i,j]] \, zy[[i,j]]) \, vxy1[[i,j]];
Plot3D[fa1,{x,0,0.99},{y,0,0.99},PlotPoints→60]
Plot3D[f1-fa1,{x,0,0.99},{y,0,0.99},PlotPoints→60]$$

```

Пример 14. Аппроксимация двумерного синусоидального сигнала с помощью блочно-импульсных базисных функций

```

v[x_,i_,m_,T_]:=If[(i-1) T/m≤x<i T/m,1,0];
vx=v[x,i,10,10];
vy=v[y,j,10,10];
vxy=vx vy;
vxy1:=Table[vxy,{i,10},{j,10}];
f[x_,y_]:=10Sin[π*(x+y)/10];
f1=f[x,y];
h:=1;T:=10;

$$Fv[i_,j_] := \frac{1}{h^2} \int_{(i-1)*h}^{i*h} \int_{(j-1)*h}^{j*h} f1 \, dy \, dx;$$

FV=Table[Fv[i,j],{i,10},{j,10}];
N[FV,3]//MatrixForm
Plot3D[f1,{x,0,10},{y,0,10}]
fa := 
$$\sum_{i=1}^{10} \sum_{j=1}^{10} FV[[i,j]] \, vxy1[[i,j]];
Plot3D[fa,{x,0,10},{y,0,10},PlotPoints→60]
Plot3D[f1-fa,{x,0,10},{y,0,10},PlotPoints→60]
ListPlot3D[FV]$$

```

Приложение 3. Генераторы локально-импульсных базисных функций на основе смещенных полиномов Лежандра

Программа-генератор блочно-импульсных функций (локально-импульсных функций нулевого порядка)

```
v[t_,m_,h_]:=Table[If[(i-1)*h<=t<i*h,1,0},{i,m}];
V=v[t,5,0.2];
P:=Table[Plot[V[[i]],{t,0,1.05}],{i,5}];
P
```

Программа-генератор локально-импульсных функций первого порядка

```
v[t_,m_,h_]:=Table[If[(i-1)*h<=t<i*h,1,0},{i,m}];
w[t_,m_,h_]:=Table[(2t/h-1-2(i-
1))*v[t,m,h][[i]],{i,m}];
V=v[t,5,0.2];
W=w[t,5,0.2];
P:=Table[Plot[{V[[i]],W[[i]]},{t,0,1.05}],{i,5}];
P
```

Программа-генератор локально-импульсных функций второго порядка

```
v[t_,m_,h_]:=Table[If[(i-1)*h<=t<i*h,1,0},{i,m}];
w[t_,m_,h_]:=Table[(2t/h-1-2(i-
1))*v[t,m,h][[i]],{i,m}];
u[t_,m_,h_]:=
Table[ $\left(6\left(\frac{t}{h}\right)^2 - \frac{6(2i-1)*t}{h} + 6i^2 - 6i + 1\right) * v[t,m,h][[i]],{i,m}];
V=v[t,5,0.2];
W=w[t,5,0.2];
U=u[t,5,0.2];
P:=Table[Plot[{V[[i]],W[[i]],U[[i]]},{t,0,1.05}],{i,5}];
P$ 
```


Приложение 4. Программы и решения к упражнениям

Упражнение 1. Аппроксимация непрерывного сигнала

Вариант 1.

m:=10;T:=2;

ss[t_, i_, T_] := $\left(\frac{t}{T}\right)^{i-1};$

S=Table[ss[t,i,2],{i,m}]

{1, $\frac{t}{2}, \frac{t^2}{4}, \frac{t^3}{8}, \frac{t^4}{16}, \frac{t^5}{32}, \frac{t^6}{64}, \frac{t^7}{128}, \frac{t^8}{256}, \frac{t^9}{512}$ }

W := \int_0^T Table[S[[i]] * S[[j]], {i, m}, {j, m}] dt;

W

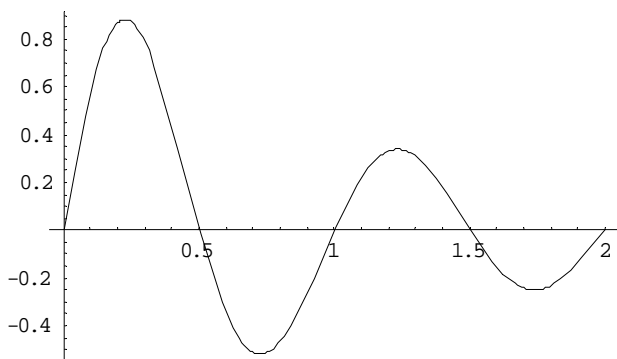
{{2, 1, $\frac{2}{3}, \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}$ },
{1, $\frac{2}{3}, \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}$ },
{ $\frac{2}{3}, \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}$ },
{ $\frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}$ },
{ $\frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}, \frac{1}{7}$ },
{ $\frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}, \frac{1}{7}, \frac{2}{15}$ },
{ $\frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}, \frac{1}{7}, \frac{2}{15}, \frac{1}{8}$ },
{ $\frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}, \frac{1}{7}, \frac{2}{15}, \frac{1}{8}, \frac{2}{17}$ },
{ $\frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}, \frac{1}{7}, \frac{2}{15}, \frac{1}{8}, \frac{2}{17}, \frac{1}{9}$ },
{ $\frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13}, \frac{1}{7}, \frac{2}{15}, \frac{1}{8}, \frac{2}{17}, \frac{1}{9}, \frac{2}{19}$ }}}

N[%,3]//MatrixForm

```
(
2.00  1.00  0.667  0.500  0.400  0.333  0.286  0.250  0.222  0.200
1.00  0.667  0.500  0.400  0.333  0.286  0.250  0.222  0.200  0.182
0.667  0.500  0.400  0.333  0.286  0.250  0.222  0.200  0.182  0.167
0.500  0.400  0.333  0.286  0.250  0.222  0.200  0.182  0.167  0.154
0.400  0.333  0.286  0.250  0.222  0.200  0.182  0.167  0.154  0.143
0.333  0.286  0.250  0.222  0.200  0.182  0.167  0.154  0.143  0.133
0.286  0.250  0.222  0.200  0.182  0.167  0.154  0.143  0.133  0.125
0.250  0.222  0.200  0.182  0.167  0.154  0.143  0.133  0.125  0.118
0.222  0.200  0.182  0.167  0.154  0.143  0.133  0.125  0.118  0.111
0.200  0.182  0.167  0.154  0.143  0.133  0.125  0.118  0.111  0.105
)
```

```
xx[t_, a_, b_, c_] :=  $\frac{1}{\sqrt{a + b * t^c}}$  * Sin[2  $\pi$  * t];
```

```
x=xx[t,1,5,2];
Plot[x,{t,0,T}]
```



```
-Graphics-
```

```
Q := N[ $\int_0^T$  Table[S[[i]] * x, {i, m}] dt];
```

```
Q
```

```
{0.13997,-0.0297076,-0.0366147,-0.0346723,-
0.0333227,-0.0320294,-0.0304303,-0.0285507,-
0.0265254,-0.0244791}
```

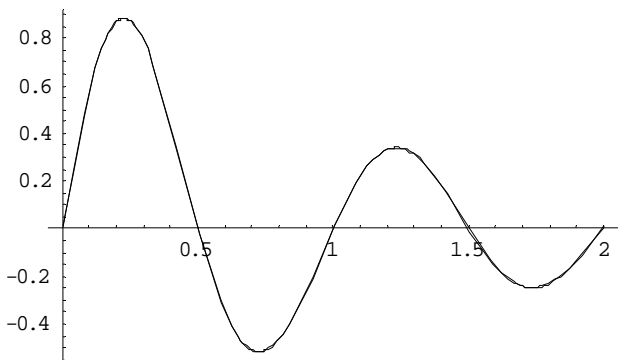
```
X=Inverse[W].Q
```

```
{0.0157383,10.6832,56.7837,-1202.18,5067.11,-
9110.2,7000.15,-467.311,-2234.64,879.6}
```

```
xa=X.S
```

```
0.0157383 + 5.3416 t + 14.1959 t2 - 150.272 t3 + 316.694 t4 -
284.694 t5 + 109.377 t6 - 3.65087 t7 - 8.72905 t8 + 1.71797 t9
```

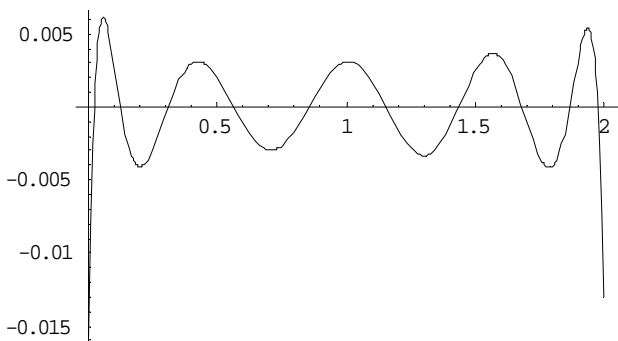
```
Plot[{x,xa},{t,0,T}]
```



-Graphics-

e:=x-xa;

Plot[x-xa,{t,0,2}]



-Graphics-

Вариант 2.

m:=7;T:=2;

ss[t_, i_, T_] := $\left(\frac{t}{T}\right)^{i-1}$;

S=Table[ss[t,i,2],{i,m}]

$\left\{1, \frac{t}{2}, \frac{t^2}{4}, \frac{t^3}{8}, \frac{t^4}{16}, \frac{t^5}{32}, \frac{t^6}{64}\right\}$

W := $\int_0^T \text{Table}[S[[i]] * S[[j]], \{i, m\}, \{j, m\}] dt$;

W

$$\left\{ \left\{ 2, 1, \frac{2}{3}, \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7} \right\}, \right. \\ \left. \left\{ 1, \frac{2}{3}, \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4} \right\}, \left\{ \frac{2}{3}, \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9} \right\}, \right. \\ \left. \left\{ \frac{1}{2}, \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5} \right\}, \left\{ \frac{2}{5}, \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11} \right\}, \right. \\ \left. \left\{ \frac{1}{3}, \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6} \right\}, \left\{ \frac{2}{7}, \frac{1}{4}, \frac{2}{9}, \frac{1}{5}, \frac{2}{11}, \frac{1}{6}, \frac{2}{13} \right\} \right\}$$

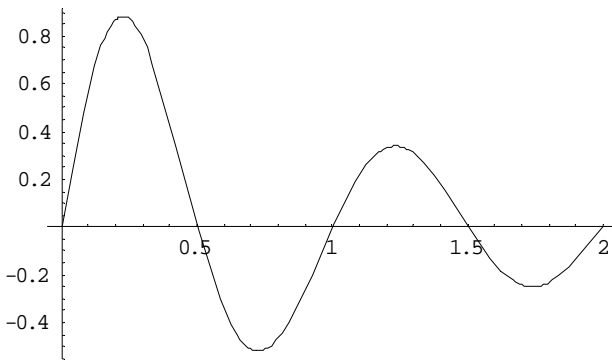
N[%3]//MatrixForm

$$\begin{pmatrix} 2.00 & 1.00 & 0.667 & 0.500 & 0.400 & 0.333 & 0.286 \\ 1.00 & 0.667 & 0.500 & 0.400 & 0.333 & 0.286 & 0.250 \\ 0.667 & 0.500 & 0.400 & 0.333 & 0.286 & 0.250 & 0.222 \\ 0.500 & 0.400 & 0.333 & 0.286 & 0.250 & 0.222 & 0.200 \\ 0.400 & 0.333 & 0.286 & 0.250 & 0.222 & 0.200 & 0.182 \\ 0.333 & 0.286 & 0.250 & 0.222 & 0.200 & 0.182 & 0.167 \\ 0.286 & 0.250 & 0.222 & 0.200 & 0.182 & 0.167 & 0.154 \end{pmatrix}$$

$$\mathbf{xx}[t_ , a_ , b_ , c_] := \frac{1}{\sqrt{a + b * t^c}} * \text{Sin}[2 \pi * t] ;$$

x=xx[t,1,5,2];

Plot[x,{t,0,T}]



-Graphics-

$$\mathbf{Q} := \mathbf{N} \left[\int_0^T \text{Table}[S[[i]] * \mathbf{x}, \{i, m\}] dt \right] ;$$

Q

{0.13997,-0.0297076,-0.0366147,-0.0346723,-0.0333227,-0.0320294,-0.0304303}

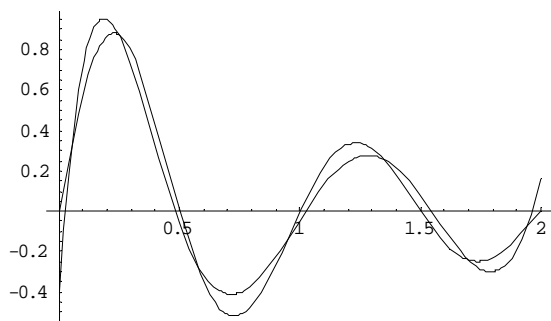
X=Inverse[W].Q

{-0.398397,35.4024,-301.655,953.808,-1388.84,939.6,-237.761}

xa=X.S

$$-0.398397 + 17.7012 t - 75.4137 t^2 + 119.226 t^3 - 86.8023 t^4 + 29.3625 t^5 - 3.71501 t^6$$

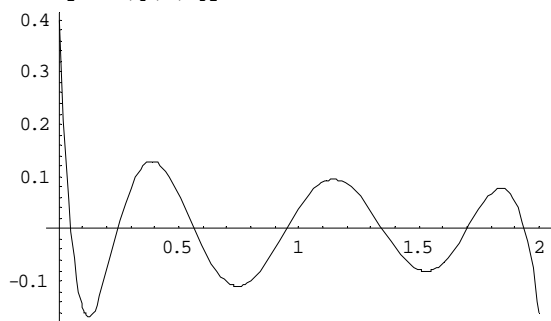
Plot[{x,xa},{t,0,T}]



- Graphics -

e:=x-xa;

Plot[x-xa,{t,0,2}]



- Graphics -

Вариант 3.

m:=5;T:=2;

ss[t_, i_, T_] := $\left(\frac{t}{T}\right)^{i-1}$;

S=Table[ss[t,i,2],{i,m}]

$\left\{1, \frac{t}{2}, \frac{t^2}{4}, \frac{t^3}{8}, \frac{t^4}{16}\right\}$

```
W := ∫0T Table[S[[i]] * S[[j]], {i, m}, {j, m}] dt;
```

W

```
{ {2, 1, 2/3, 1/2, 2/5}, {1, 2/3, 1/2, 2/5, 1/3},
  {2/3, 1/2, 2/5, 1/3, 2/7}, {1/2, 2/5, 1/3, 2/7, 1/4}, {2/5, 1/3, 2/7, 1/4, 2/9} }
```

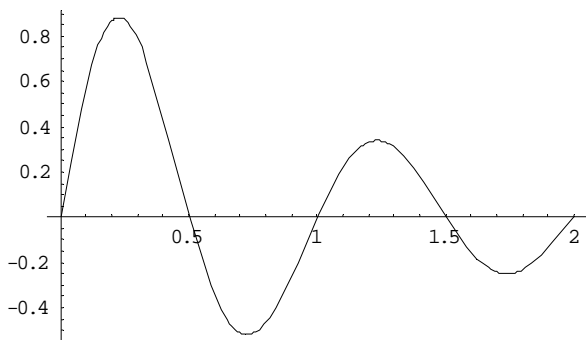
N[%3]//MatrixForm

```
( 2.00  1.00  0.667  0.500  0.400
  1.00  0.667  0.500  0.400  0.333
  0.667  0.500  0.400  0.333  0.286
  0.500  0.400  0.333  0.286  0.250
  0.400  0.333  0.286  0.250  0.222 )
```

```
xx[t_, a_, b_, c_] := 1 / √(a + b * t^c) * Sin[2 π * t];
```

x=xx[t,1,5,2];

Plot[x,{t,0,T}]



-Graphics -

```
Q := N[ ∫0T Table[S[[i]] * x, {i, m}] dt];
```

Q

```
{0.13997,-0.0297076,-0.0366147,-0.0346723,-0.0333227}
```

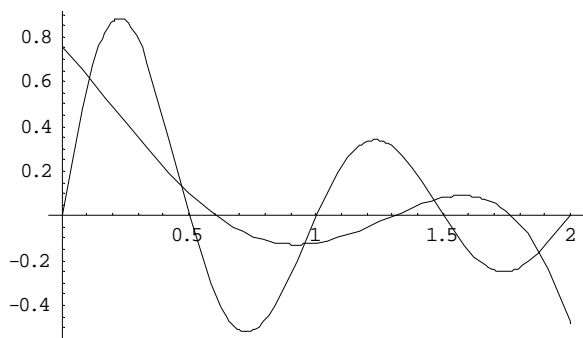
X=Inverse[W].Q

```
{0.757004,-2.34745,-4.98406,18.5888,-12.4957}
```

xa=X.S

```
0.757004 - 1.17372 t - 1.24601 t2 + 2.3236 t3 - 0.780982 t4
```

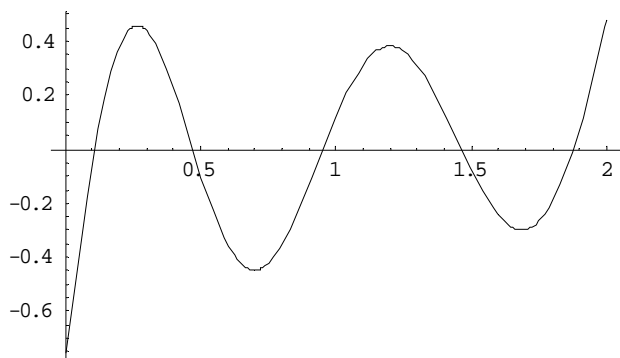
Plot[{x,xa},{t,0,T}]



- Graphics -

e:=x-xa;

Plot[x-xa,{t,0,2}]



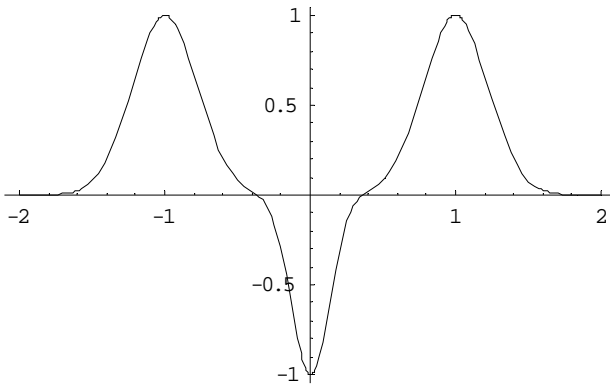
- Graphics -

Упражнение 2. Аппроксимация непрерывного сигнала с помощью блочно-импульсной системы базисных функций

Вариант 1. m=100; h=0.04.

$x[t] := e^{-10(t-1)^2} + e^{-10(t+1)^2} - e^{-30t^2};$

Plot[x[t],{t,-2,2}]



- Graphics -

m=100;h=0.04;

v[t_,i_,h_]:=If[-2+(i-1)*h<=t<-2+i*h,1,0];

V=Table[v[t,i,0.04],{i,1,m}];

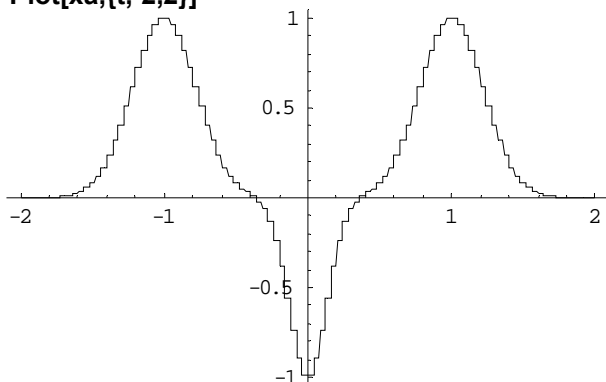
X := Table[$\frac{1}{h} * \int_{-2+(i-1)*h}^{-2+i*h} x[t] dt$, {i, 1, m}]

X

```
Out[7]= {0.0000691053, 0.000148652, 0.000309718, 0.000625032,
0.00122173, 0.00231307, 0.00424171, 0.00753412,
0.0129617, 0.0215989, 0.034861, 0.0544988, 0.0825227,
0.121032, 0.171935, 0.236575, 0.315292, 0.407002,
0.508885, 0.616286, 0.722911, 0.821346, 0.903873,
0.963448, 0.994692, 0.994692, 0.963448, 0.903873,
0.821346, 0.722911, 0.616286, 0.508885, 0.407002,
0.31529, 0.236565, 0.17189, 0.120862, 0.0819383,
0.0526659, 0.0296351, 0.00805315, -0.0189593,
-0.0608541, -0.128962, -0.233564, -0.378523,
-0.555194, -0.739317, -0.894658, -0.984128,
-0.984128, -0.894658, -0.739317, -0.555194,
-0.378523, -0.233564, -0.128962, -0.0608541,
-0.0189593, 0.00805315, 0.0296351, 0.0526659,
0.0819383, 0.120862, 0.17189, 0.236565, 0.31529,
0.407002, 0.508885, 0.616286, 0.722911, 0.821346,
0.903873, 0.963448, 0.994692, 0.994692, 0.963448,
0.903873, 0.821346, 0.722911, 0.616286, 0.508885,
0.407002, 0.315292, 0.236575, 0.171935, 0.121032,
0.0825227, 0.0544988, 0.034861, 0.0215989, 0.0129617,
0.00753412, 0.00424171, 0.00231307, 0.00122173,
0.000625032, 0.000309718, 0.000148652, 0.0000691053}
```

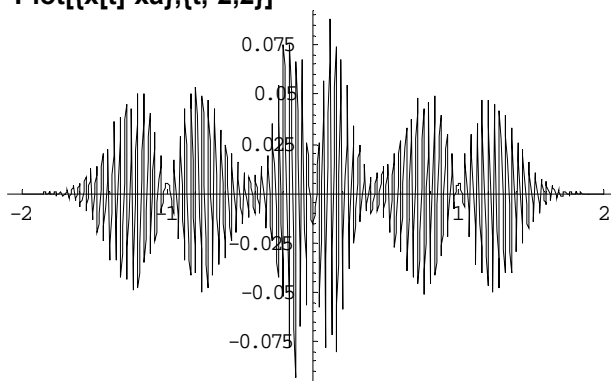
xa=X.V;

Plot[xa,{t,-2,2}]



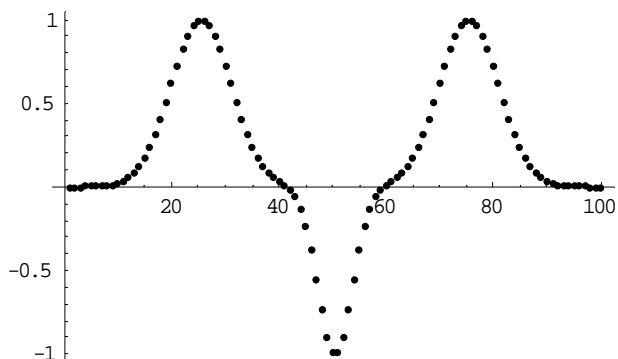
- Graphics -

Plot[{x[t]-xa},{t,-2,2}]



- Graphics -

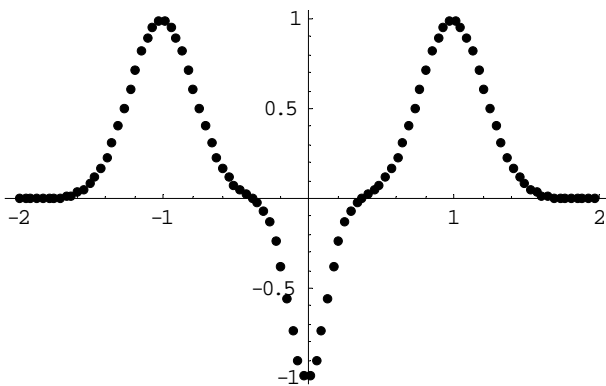
ListPlot[X,PlotStyle \rightarrow PointSize[0.015]]



- Graphics -

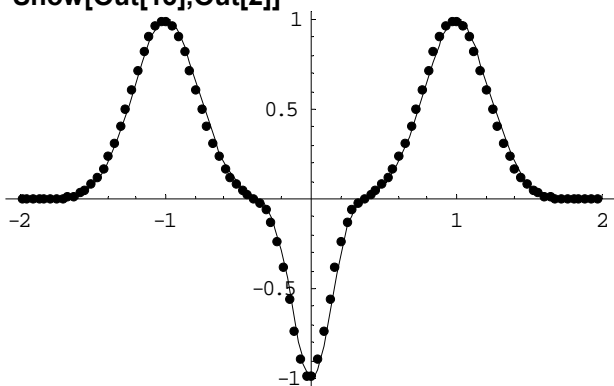
```
Z:=Table[{-2+(i-1)*h,X[[i]]},{i,100}];
```

```
ListPlot[Z,PlotStyle → PointSize[0.015]]
```



- Graphics -

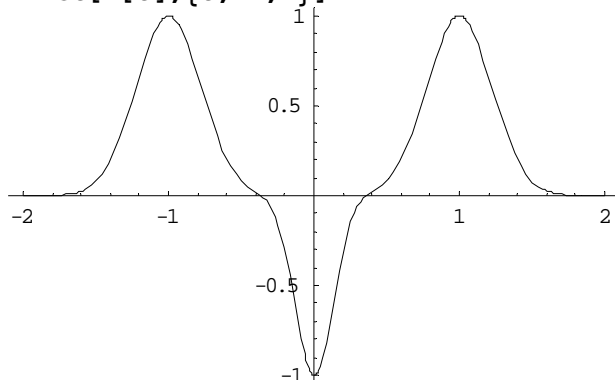
```
Show[Out[13],Out[2]]
```



- Graphics -

Вариант 2. m=50; h=0.08.

```
x[t_] := e-10 (t-1)^2 + e-10 (t+1)^2 - e-30 t^2;
Plot[x[t], {t, -2, 2}]
```



-Graphics-

```
m=50;h=0.08;
```

```
v[t_,i_,h_]:=If[-2+(i-1)*h<=t<=-2+i*h,1,0];
```

```
V=Table[v[t,i,0.08],{i,1,m}];
```

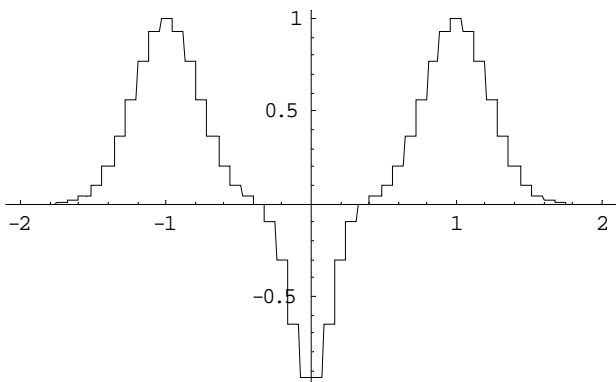
```
X:=Table[ $\frac{1}{h} \int_{-2+(i-1)h}^{-2+i h} x[t] dt$ , {i, 1, m}]
```

```
X
```

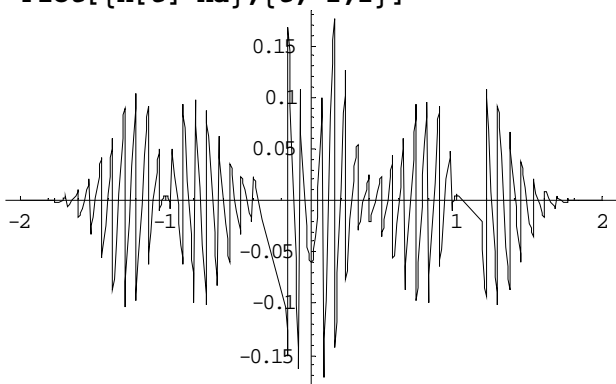
```
{0.000108878, 0.000467375, 0.0017674, 0.00588792,
 0.0172803, 0.0446799, 0.101777, 0.204255, 0.361147,
 0.562586, 0.772128, 0.933661, 0.994692, 0.933661,
 0.772128, 0.562586, 0.361146, 0.204228, 0.1014,
 0.0411505, -0.00545309, -0.0949081, -0.306044,
 -0.647255, -0.939393, -0.939393, -0.647255,
 -0.306044, -0.0949081, -0.00545309, 0.0411505,
 0.1014, 0.204228, 0.361146, 0.562586, 0.772128,
 0.933661, 0.994692, 0.933661, 0.772128, 0.562586,
 0.361147, 0.204255, 0.101777, 0.0446799, 0.0172803,
 0.00588792, 0.0017674, 0.000467375, 0.000108878}
```

```
xa=X.V;
```

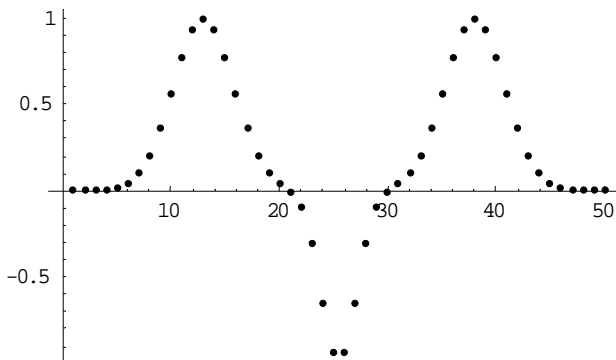
```
Plot[xa,{t,-2,2}]
```



```
-Graphics-
Plot[{x[t]-xa},{t,-2,2}]
```

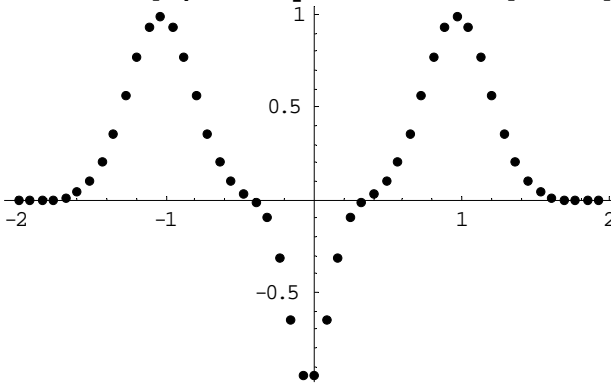


```
-Graphics-
ListPlot[X,PlotStyle->PointSize[0.015]]
```

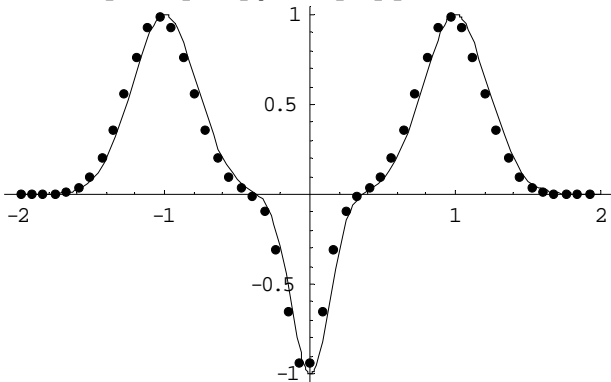


```
-Graphics-
```

```
Z:=Table[{-2+(i-1)*h,X[[i]]},{i,50}];
ListPlot[Z,PlotStyle->PointSize[0.015]]
```



```
-Graphics-
Show[Out[13],Out[2]]
```



```
-Graphics-
```

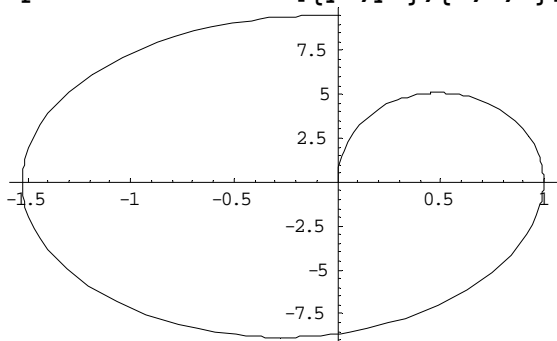
Упражнение 3. Аппроксимация фазового портрета непрерывного сигнала с помощью системы базисных функций на основе смещенных полиномов Лежандра

```
yo[a_, b_, c_, t_] := a*t* $e^{-b*t}$ *Sin[c*t];
y1=yo[5,1.2,2 $\pi$ ,t];
y2=Simplify[D[y1,t]];
y1
 $5 e^{-1.2 t} t \sin[2 \pi t]$ 
y2
```

```

e-1.2t (31.4159 t Cos[2 π t] + (5. - 6. t) Sin[2 π t])
p1=ParametricPlot[{y1,y2},{t,0,1}]

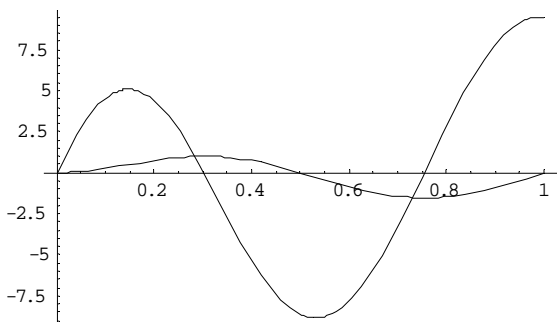
```



```

-Graphics-
p2=Plot[{y1,y2},{t,0,1}]

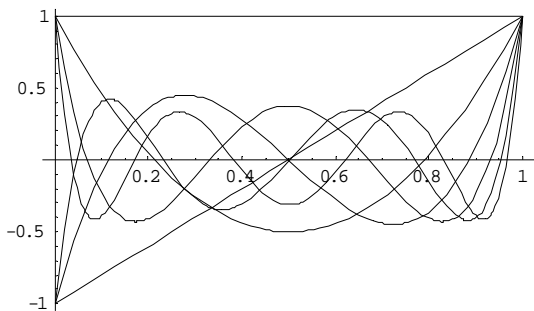
```



```

-Graphics-
SL[x_]:=Evaluate[Table[LegendreP[n,x],{n,0,6}]];
s1=Simplify[SL[2t-1]];
p3=Plot[Evaluate[s1],{t,0,1}]

```



```

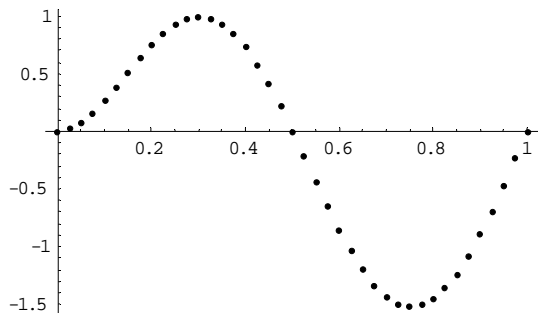
-Graphics-
h=0.025;

```

```

yd1:=Table[{k*h,y1/.t->k*h},{k,0,40}];
p4=ListPlot[yd1,PlotStyle->PointSize[0.015]]

```

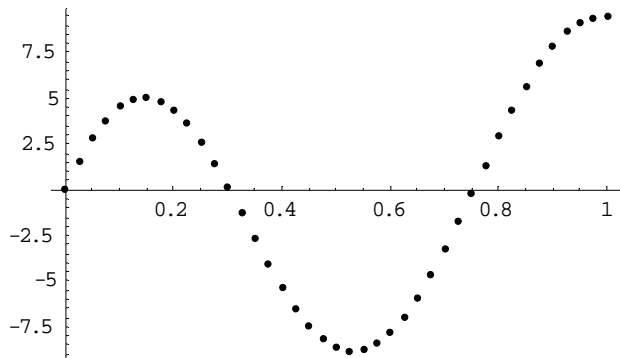


-Graphics-

```

yd2:=Table[{k*h,y2/.t->k*h},{k,0,40}];
p5=ListPlot[yd2,PlotStyle->PointSize[0.015]]

```



-Graphics-

yd1//TableForm

0	0
0.025	0.0189764
0.05	0.0727553
0.075	0.155594
0.1	0.260659
0.125	0.380383
0.15	0.506811
0.175	0.631956
0.2	0.748128
0.225	0.848229
0.25	0.926023
0.275	0.97635
0.3	0.995294
0.325	0.980301
0.35	0.930234
0.375	0.845383
0.4	0.727424
0.425	0.579316
0.45	0.405178
0.475	0.210111
0.5	1.6802×10^{-16}
0.525	-0.218704
0.55	-0.439219
0.575	-0.654668
0.6	-0.858317
0.625	-1.04379
0.65	-1.20529
0.675	-1.33775
0.7	-1.43703
0.725	-1.5
0.75	-1.52464
0.775	-1.51007
0.8	-1.45661
0.825	-1.36569
0.85	-1.23984
0.875	-1.08256
0.9	-0.898242
0.925	-0.691977
0.95	-0.46944
0.975	-0.236691
1.	-3.68844×10^{-16}

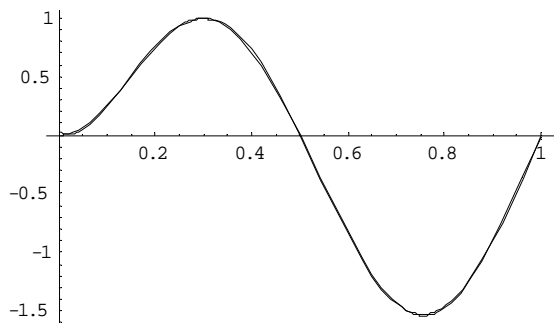
yd2//TableForm

0	0
0.025	1.48909
0.05	2.77472
0.075	3.80656
0.1	4.548
0.125	4.97662
0.15	5.08416
0.175	4.876
0.2	4.37021
0.225	3.59615
0.25	2.59286
0.275	1.40712
0.3	0.0913707
0.325	-1.29843
0.35	-2.70499
0.375	-4.07181
0.4	-5.34515
0.425	-6.47589
0.45	-7.42101
0.475	-8.14499
0.5	-8.62071
0.525	-8.83021
0.55	-8.76497
0.575	-8.42596
0.6	-7.82333
0.625	-6.97586
0.65	-5.91009
0.675	-4.65929
0.7	-3.26221
0.725	-1.76171
0.75	-0.203285
0.775	1.36637
0.8	2.90089
0.825	4.35564
0.85	5.68905
0.875	6.86382
0.9	7.84789
0.925	8.61537
0.95	9.14705
0.975	9.43093
1.	9.4623

```

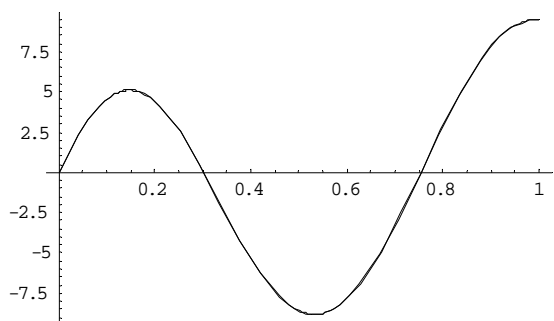
wd:=Table[s1[[i]]/.t→(k-1)*h,{i,7},{k,41}];
qd1:=Table[y1/.t→(k-1)*h,{k,41}];
qd1
{0, 0.0189764, 0.0727553, 0.155594, 0.260659,
 0.380383, 0.506811, 0.631956, 0.748128, 0.848229,
 0.926023, 0.97635, 0.995294, 0.980301, 0.930234,
 0.845383, 0.727424, 0.579316, 0.405178, 0.210111,
 1.6802×10-16, -0.218704, -0.439219, -0.654668,
 -0.858317, -1.04379, -1.20529, -1.33775,
 -1.43703, -1.5, -1.52464, -1.51007, -1.45661,
 -1.36569, -1.23984, -1.08256, -0.898242,
 -0.691977, -0.46944, -0.236691, -3.68844×10-16}
qd2:=Table[y2/.t→(k-1)*h,{k,41}];
qd2
{0, 1.48909, 2.77472, 3.80656, 4.548, 4.97662,
 5.08416, 4.876, 4.37021, 3.59615, 2.59286,
 1.40712, 0.0913707, -1.29843, -2.70499, -4.07181,
 -5.34515, -6.47589, -7.42101, -8.14499, -8.62071,
 -8.83021, -8.76497, -8.42596, -7.82333, -6.97586,
 -5.91009, -4.65929, -3.26221, -1.76171, -0.203285,
 1.36637, 2.90089, 4.35564, 5.68905, 6.86382,
 7.84789, 8.61537, 9.14705, 9.43093, 9.4623}
D1=PseudoInverse[Transpose[wd]].qd1;
D1
{-0.199787, -1.0832, -0.113181,
 1.50117, 0.351035, -0.446925, -0.0378222}
D2=PseudoInverse[Transpose[wd]].qd2;
D2
{-0.0016217, 1.19825, 10.8528,
 4.37982, -7.36109, -0.842044, 1.2846}
fda1=D1.s1;
fda2=D2.s1;
p6=Plot[{fda1,y1},{t,0,1},PlotRange→All]

```



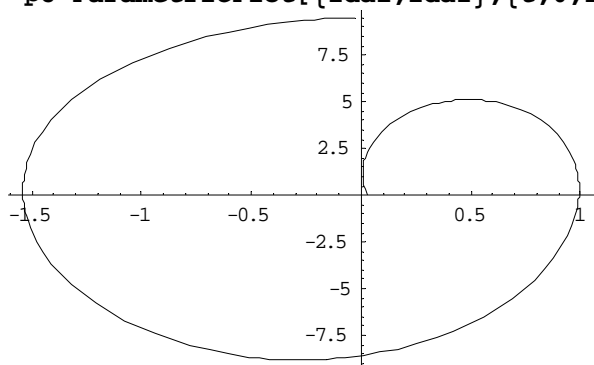
-Graphics-

p7=Plot[{fda2,y2},{t,0,1},PlotRange→All]



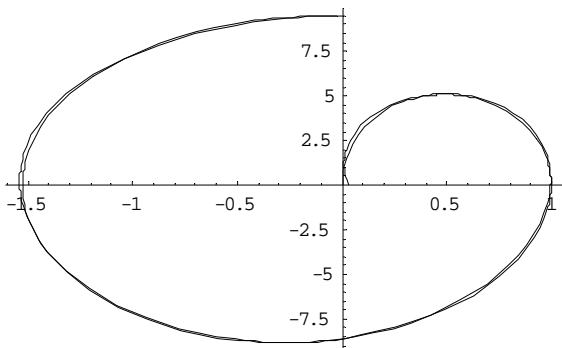
-Graphics-

p8=ParametricPlot[{fda1,fda2},{t,0,1}]



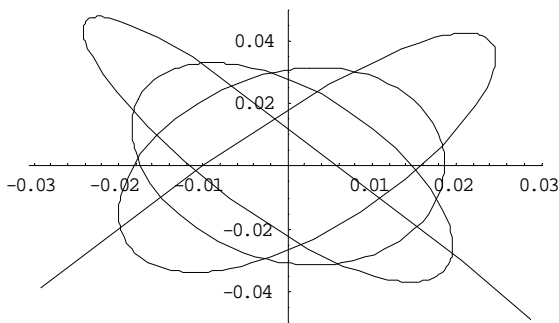
-Graphics-

Show[{p1,p8}]



-Graphics-

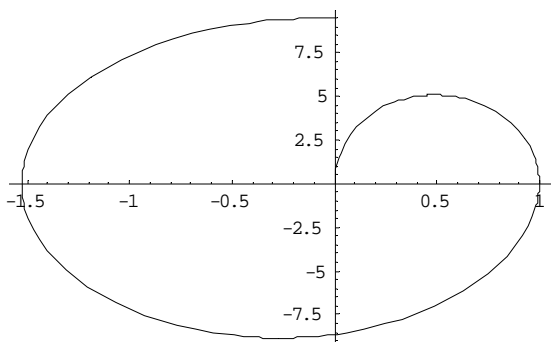
```
ParametricPlot[{y1-fda1,y2-fda2},{t,0,1}]
```



-Graphics-

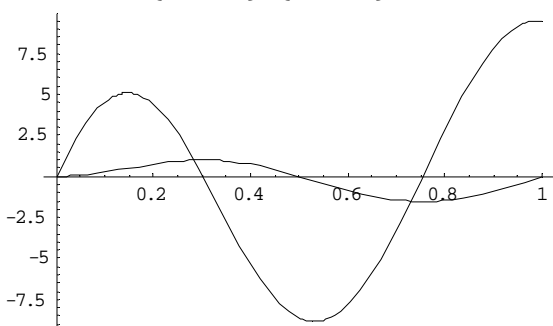
Упражнение 4. Аппроксимация фазового портрета непрерывного сигнала с помощью блочно-импульсной системы базисных функций

```
yo[a_, b_, c_, t_] := a*t*e-b*t*Sin[c*t];
y1=yo[5,1.2,2π,t];
y2=Simplify[D[y1,t]];
y1
5 e-1.2 t Sin[2 π t]
y2
e-1.2 t (31.4159 t Cos[2 π t] + (5. - 6. t) Sin[2 π t])
p1=ParametricPlot[{y1,y2},{t,0,1}]
```



-Graphics-

p2=Plot[{y1,y2},{t,0,1}]



-Graphics-

v[t_,m_,h_]:=Table[If[(i-1)*h<t<i*h,1,0],{i,m}];

V=v[t,50,0.02];

h=0.02;m=50;

sp1:=Chop[Table[$\frac{1}{h} * \int_{(i-1)*h}^{i*h} y1 dt$, {i, m}]]];

sp1

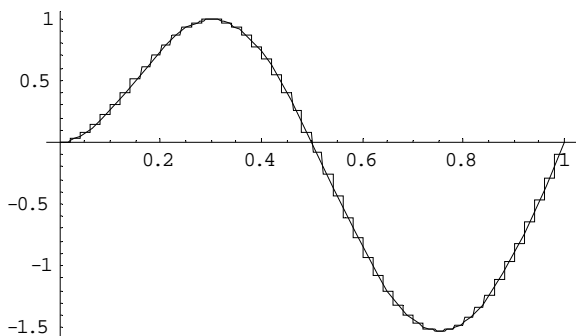
```
{0.00410763, 0.0280163, 0.0735324, 0.137649, 0.216912,
 0.307538, 0.405525, 0.506776, 0.607205, 0.70285, 0.789967,
 0.865127, 0.925288, 0.967867, 0.990792, 0.992541, 0.972165,
 0.929303, 0.864178, 0.777577, 0.670826, 0.54575, 0.404618,
 0.25009, 0.0851464, -0.0869832, -0.262894, -0.439083,
 -0.612031, -0.778273, -0.934478, -1.07751, -1.20451,
 -1.31293, -1.40059, -1.46572, -1.50701, -1.52359, -1.51508,
 -1.48158, -1.42364, -1.3423, -1.239, -1.11559, -0.974261,
 -0.817535, -0.648179, -0.469167, -0.28362, -0.0947412}
```

```
sp2 := Chop[Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} y2 dt$ , {i, m}]];
```

```
sp2
```

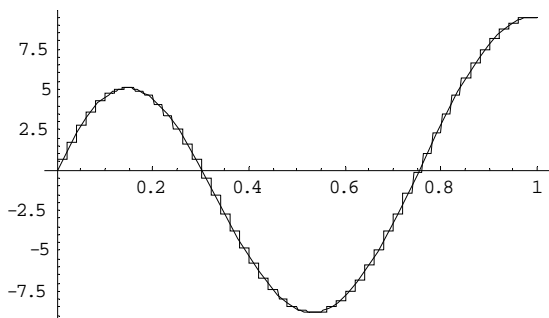
```
{0.611805, 1.75854, 2.76792, 3.61485, 4.27985, 4.74926,  
5.01524, 5.07571, 4.93411, 4.59908, 4.08401, 3.40654,  
2.58799, 1.65265, 0.62716, -0.46023, -1.58037, -2.70392,  
-3.80203, -4.84699, -5.81282, -6.67584, -7.4151,  
-8.0128, -8.45462, -8.72991, -8.83189, -8.75766,  
-8.50819, -8.08823, -7.50612, -6.77352, -5.90515,  
-4.91834, -3.8327, -2.66963, -1.45187, -0.202986,  
1.05309, 2.29254, 3.49218, 4.62983, 5.68481, 6.63824,  
7.47341, 8.17609, 8.73467, 9.14043, 9.38757, 9.47332}
```

```
ya1:=Evaluate[sp1.V];  
ya2:=Evaluate[sp2.V];  
p6=Plot[{ya1,y1},{t,0,1},PlotRange->All]
```



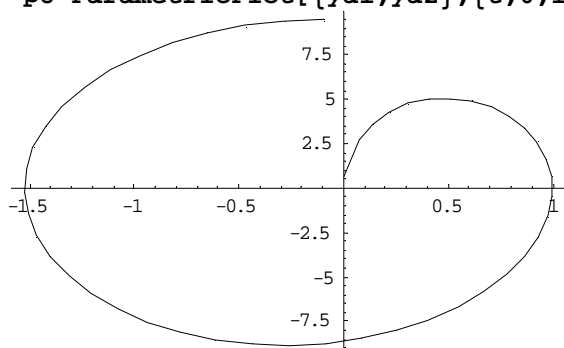
```
-Graphics-
```

```
p7=Plot[{ya2,y2},{t,0,1},PlotRange->All]
```



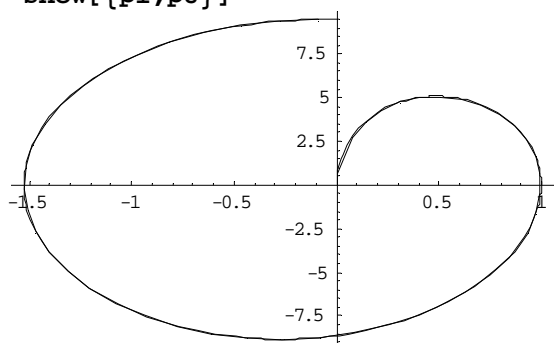
```
-Graphics-
```

```
p8=ParametricPlot[{ya1,ya2},{t,0,1}]
```



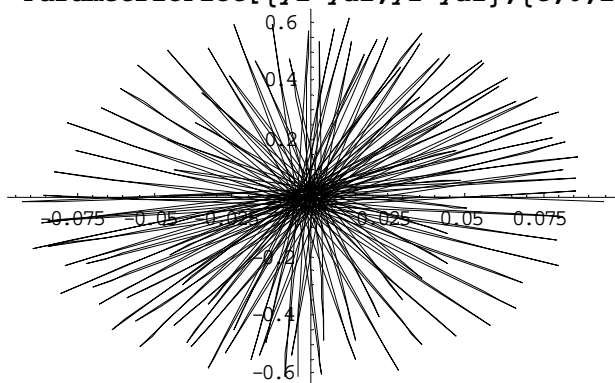
```
-Graphics-
```

```
Show[{p1,p8}]
```



```
-Graphics-
```

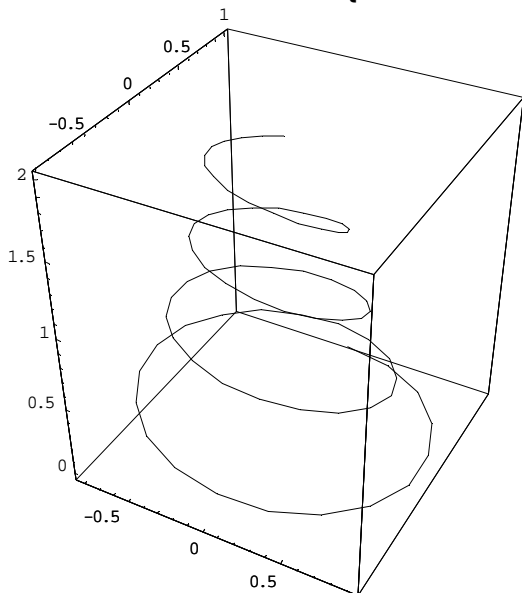
```
ParametricPlot[{y1-ya1,y2-ya2},{t,0,1}]
```



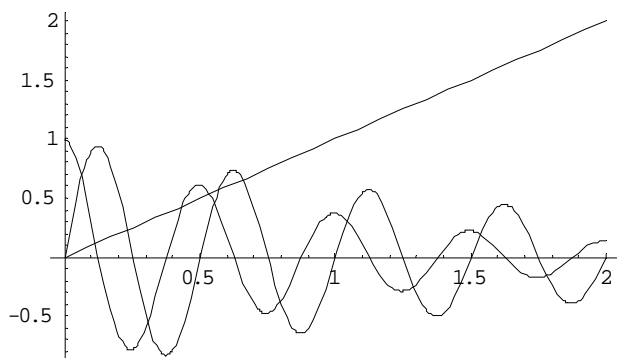
```
-Graphics-
```

Упражнение 5. Аппроксимация пространственной кривой, заданной параметрическими составляющими с помощью блочно-импульсной системы базисных функций

```
fx:=Exp[-0.5t]*Sin[4π*t];
fy:=Exp[-t]*Cos[4π*t];
fz:=t;
p1=ParametricPlot3D[{fx, fy, fz}, {t, 0,2}]
```



```
-Graphics3D-
p2=Plot[{fx,fy,fz},{t,0,2}]
```




```
-Graphics-
v[t_,m_,h_]:=Table[If[(i-1)*h<t<i*h,1,0],{i,m}];
V=v[t,100,0.02];
h=0.02;m=100;
```

```
sp1:=Chop[Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} f_x dt$ , {i, m}]]];
```

```
sp1
{0.124174, 0.3615, 0.571603, 0.74193, 0.862656, 0.927242,
 0.932775, 0.88008, 0.773592, 0.621013, 0.432761,
 0.221272, 0.000184538, -0.216537, -0.415475, -0.584587,
 -0.713926, -0.796221, -0.827275, -0.80617, -0.735252,
 -0.619927, -0.468259, -0.290418, -0.0980037,
 0.096707, 0.281537, 0.445165, 0.577815, 0.671837,
 0.722137, 0.726446, 0.685407, 0.602474, 0.483645,
 0.337034, 0.172327, 0.000143719, -0.168639, -0.323572,
 -0.455277, -0.556006, -0.620097, -0.644283, -0.627846,
 -0.572615, -0.482799, -0.36468, -0.226178, -0.0763254,
 0.0753155, 0.219261, 0.346694, 0.450003, 0.523227,
 0.562401, 0.565757, 0.533795, 0.469207, 0.376663,
 0.262483, 0.134208, 0.000111928, -0.131336, -0.251998,
 -0.35457, -0.433018, -0.482932, -0.501768, -0.488967,
 -0.445953, -0.376005, -0.284013, -0.176147, -0.0594422,
 0.0586558, 0.170761, 0.270006, 0.350463, 0.40749,
 0.437998, 0.440612, 0.41572, 0.365419, 0.293346,
 0.204422, 0.104522, 0.0000871698, -0.102285, -0.196256,
 -0.276139, -0.337235, -0.376108, -0.390777, -0.380808,
 -0.347308, -0.292833, -0.22119, -0.137184, -0.0462937}
```

```
sp2:=Chop[Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} f_y dt$ , {i, m}]]];
```

```
sp2
```

```
{0.979728, 0.900089, 0.767784, 0.593078, 0.388462,
0.167792, -0.0546255, -0.264936, -0.450579, -0.601015,
-0.708301, -0.767478, -0.776766, -0.737544, -0.654146,
-0.533473, -0.384466, -0.217472, -0.0435465, 0.126258,
0.281579, 0.413357, 0.514347, 0.579498, 0.606175,
0.594235, 0.545932, 0.465684, 0.35972, 0.235614,
0.101771, -0.033132, -0.160692, -0.27329, -0.364534,
-0.429606, -0.465499, -0.471132, -0.447343, -0.396759,
-0.323567, -0.23319, -0.131903, -0.0264123, 0.0765794,
0.170786, 0.250714, 0.311967, 0.351483, 0.367664,
0.360422, 0.331124, 0.282452, 0.218181, 0.142907,
0.0617272, -0.0200956, -0.0974645, -0.165759,
-0.221101, -0.260569, -0.282339, -0.285756, -0.271327,
-0.240647, -0.196254, -0.141437, -0.0800034,
-0.0160199, 0.0464478, 0.103587, 0.152066, 0.189218,
0.213185, 0.222999, 0.218607, 0.200837, 0.171316,
0.132333, 0.0866776, 0.0374394, -0.0121886,
-0.0591152, -0.100538, -0.134105, -0.158043,
-0.171248, -0.17332, -0.164568, -0.14596, -0.119034,
-0.0857859, -0.0485245, -0.00971655, 0.028172,
0.0628288, 0.0922324, 0.114766, 0.129303, 0.135256}
```

```
sp3 := Chop[Table[ $\frac{1}{h} * \int_{(i-1)*h}^{i*h} fz dt$ , {i, m}]];
```

```
sp3
```

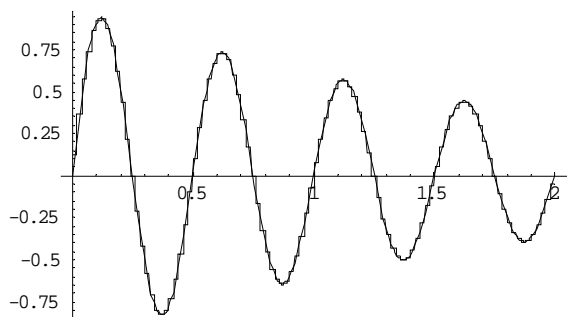
```
{0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15, 0.17,
0.19, 0.21, 0.23, 0.25, 0.27, 0.29, 0.31, 0.33, 0.35,
0.37, 0.39, 0.41, 0.43, 0.45, 0.47, 0.49, 0.51, 0.53,
0.55, 0.57, 0.59, 0.61, 0.63, 0.65, 0.67, 0.69, 0.71,
0.73, 0.75, 0.77, 0.79, 0.81, 0.83, 0.85, 0.87,
0.89, 0.91, 0.93, 0.95, 0.97, 0.99, 1.01, 1.03,
1.05, 1.07, 1.09, 1.11, 1.13, 1.15, 1.17, 1.19,
1.21, 1.23, 1.25, 1.27, 1.29, 1.31, 1.33, 1.35,
1.37, 1.39, 1.41, 1.43, 1.45, 1.47, 1.49, 1.51,
1.53, 1.55, 1.57, 1.59, 1.61, 1.63, 1.65, 1.67,
1.69, 1.71, 1.73, 1.75, 1.77, 1.79, 1.81, 1.83,
1.85, 1.87, 1.89, 1.91, 1.93, 1.95, 1.97, 1.99}
```

```
yal:=Evaluate[sp1.V];
```

```

ya2:=Evaluate[sp2.V];
ya3:=Evaluate[sp3.V];
p3=Plot[{ya1,fx},{t,0,2},PlotRange→All]

```

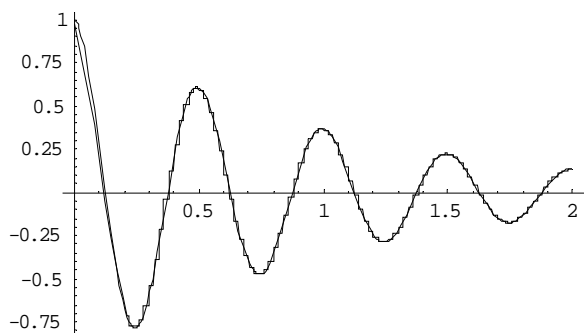


-Graphics-

```

p4=Plot[{ya2,fy},{t,0,2},PlotRange→All]

```

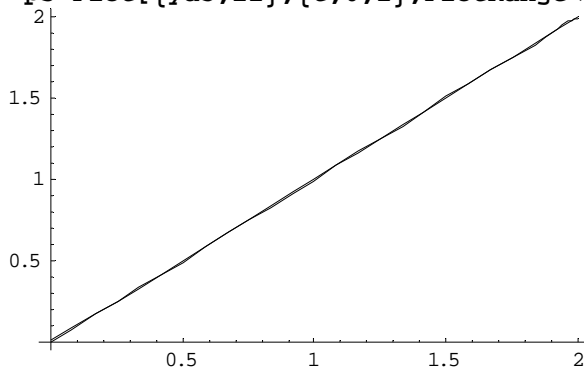


-Graphics-

```

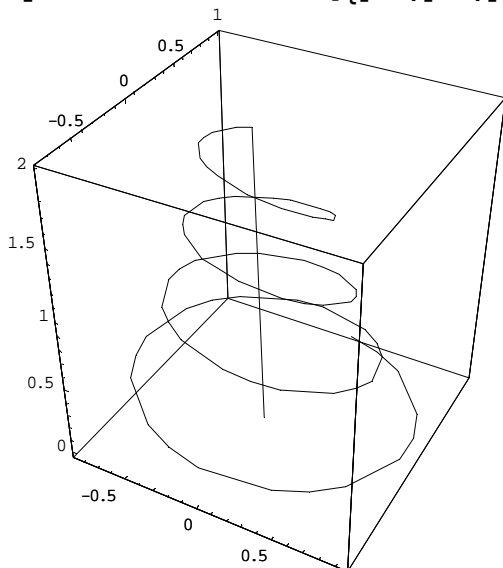
p5=Plot[{ya3,fz},{t,0,2},PlotRange→All]

```

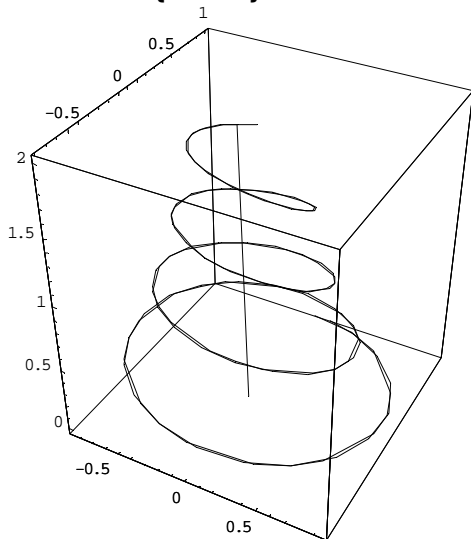


-Graphics-

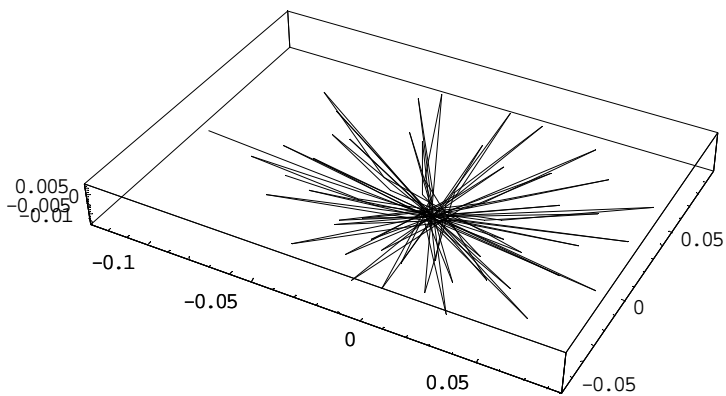
```
p6=ParametricPlot3D[{ya1,ya2,ya3},{t,0,2}]
```



```
-Graphics3D-  
p7=Show[{p1,p6}]
```



```
-Graphics3D-  
p8=ParametricPlot3D[{fx-ya1,fy-ya2,fz-ya3},{t,0,1}]
```



-Graphics3D-

Список литературы

1. *Андре Анго* Математика для электро- и радиоинженеров. – М.: «Наука», 1965. – 780 с.
2. *Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н.* MATLAB 7. – СПб. 6: БХВ-Петербург, 2005. – 1104 с.
3. *Васильев А.Н.* Maple 8. Самоучитель. — Издательский дом «Вильямс», 2003. - 352 с.
4. *Васильев В.В., Грездов Г.И., Симак Л.А.* и др. Моделирование динамических систем: Аспекты мониторинга и обработки сигналов. – К.: НАН Украины, 2002.— ISBN 966-02-0966-5. – 344 с.
5. *Гантмахер Ф.Р.* Теория матриц. – М.: «Наука», 1966. – 576 с.
6. *Гончаров В.Л.* Теория интерполирования и приближения функций.— М.: ГИТТЛ, 1954. – 328 с.
7. *Кирьянов Д.В.* Самоучитель Mathcad 11. – СПб.:БХВ-Петербург, 2003. – 560 с.
8. *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. – М.: «Наука», 1977. – 832 с.
9. *Пухов Г.Е.* Методы анализа и синтеза квазианалоговых электронных цепей. – К.: «Наукова Думка», 1967. – 568 с.
10. *Симак Л.А.* Аппрокимирующие полиномиальные спектры непрерывных сигналов и их применение // Препринт № 112. – К.: Институт проблем моделирования в энергетике АН УССР, 1987.—45 с.
11. *Симак Л.А.* Метод определения аппроксимирующего спектра функций, заданных реальными сигналами // Электронное моделирование. – 1984, Т.6, № 4. – С. 89 -96.
12. *Симак Л.А.* Аппрокимирующие импульсные спектры в приложении к дробно-дифференциальному анализу // Препринт № 8-89. – К.: Институт проблем моделирования в энергетике АН УССР, 1989. – 53 с.
13. *Симак Л.А.* Аппрокимирующие импульсные спектры нескольких переменных и их применение к моделированию дифференциальных уравнений в частных производных, включающих дифференциальные операторы нецелых порядков // Препринт № 18-89. – К.: Институт проблем моделирования в энергетике АН УССР, 1989. – 56 с.
14. *Шмидский Я.К.* Mathematica 5. Самоучитель.: М.: Издательский дом «Вильямс», 2004ю – 592 с.
15. *Stephen Wolfram* The Mathematica Book. – Wolfram Media & Cambridge University Press, 1999. -1470 p.
16. *Taan S. ElAli, Mohammad A. Karim* Continuous Signals and Systems with MATLAB. – CRC Press, 2001. – 532 с.

**Национальная Академия наук Украины
Институт проблем моделирования в энергетике им. Г.Е.Пухова
Отделение гибридных моделирующих и управляющих систем в
энергетике**

**Министерство образования и наук Украины
Национальный авиационный университет
Кафедра электротехники и светотехники**

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ И СИСТЕМ

Аппроксимация сигналов с применением системы Mathematica®

**Всеволод Викторович Васильев
Лилия Алексеевна Симак**

Печатается по решению ученого совета Отделения гибридных
моделирующих и управляющих систем в энергетике ИПМЭ им.
Г.Е.Пухова НАН Украины

Подготовка к печати, компьютерный набор, верстка
Виктория Владимировна Чёчь

Подписано к печати 04.06.07. Формат 60 x 84/16 Бумага офсетная

Усл. печ. л. 8,0. Уч.- изд. л. 7.4

Тираж 200 экз.

Заказ № _____

Отпечатано ООО «Академпресс»

Свидетельство о внесении в Государственный реестр издателей, изготовителей и
распространителей продукции № ДК 2634 от 04.10.06 г.

Адрес: 01133, г. Киев, ул. Кутузова, 18/7

Телефон: 286-42-38

