ЛАБОРАТОРНАЯ РАБОТА №2

МЕТОДЫ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ (АППРОКСИМАЦИЯ, ИНТЕРПОЛЯЦИЯ).

Цели работы

Научиться обрабатывать данные, представленные в виде множества точек используя две технологии.

1. Сглаживающая аппроксимация экспериментального ряда Методом Наименьших Квадратов.

2. Интерполяция в межузловых интервалах экспериментального ряда. Сплайн-интерполяция.

РАЗДЕЛ 1. ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ (ОБЩАЯ ПОСТАНОВКА)

1. Задать функцию в табличном виде.
2. Произвести сглаживание аппроксимирующими полиномами, полученными методом наименьших квадратов (МНК).
3. Произвести интерполяцию узловых данных на весь отрезок.
4. Все действия производить с тремя разными степенями дискретизации аргумента.
5. Построить экстраполирующие графики на расширенный диапазон значений независимого аргумента.

Все графики должны быть сопровождены легендами.

План выполнения Лабораторной работы изложен в Разделе 5.

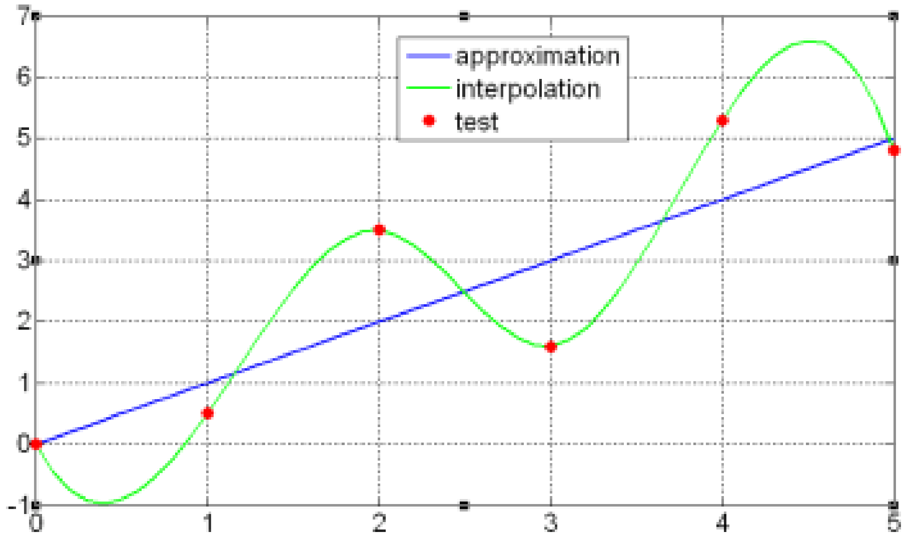
РАЗДЕЛ 2. МОТИВАЦИЯ

Для анализа результатов физического или численного эксперимента используются методы графического моделирования, большинство которых начинается с нанесения данных на двумерную координатную плоскость. Задачей анализа является выявление возможных зависимостей или других закономерностей между величинами.

Проведение физического или численного эксперимента, как правило, представляет собой сложный и трудоёмкий процесс, подверженный измерительным или вычислительным погрешностям. Поэтому полученные данные представляют собой небольшой дискретный набор разрозненных значений. Нанесённые на плоскость эти данные, зачастую представляют собой множество хаотично разбросанных точек. В свою очередь, это множество точек проигрывает в наглядности непрерывным функциям. Преимущества непрерывных функций заключается в возможности увидеть закономерности между двумя величинами: x и $y=f(x)$.

Таким образом, в задачах моделирования для повышения наглядности применяют методы замены дискретного набора данных непрерывными аналогами. При этом применяются два принципиально разных подхода.

- Можно построить "плавную" кривую, не требуя прохождения её через все имеющиеся точки. Это задача **аппроксимации**. На рисунке ей соответствует синяя линия. Красные маркеры – экспериментальные данные.
- Если требовать, чтобы построенная кривая проходила через точки, как зелёная линия на рисунке, то это задача **интерполяции**. На рисунке её соответствует зелёная линия.

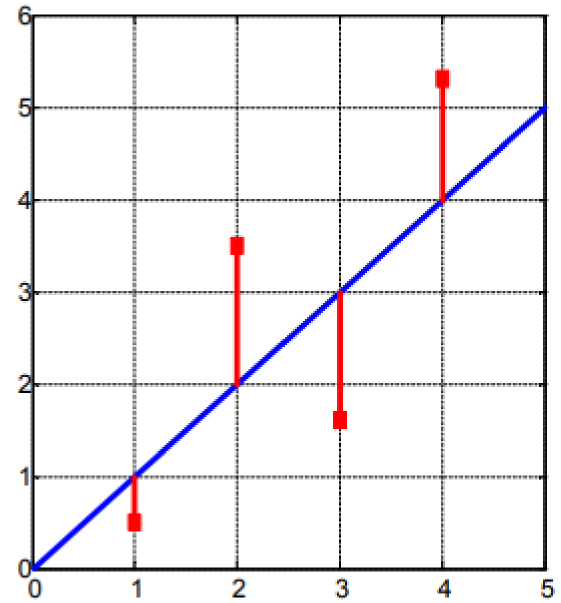


РАЗДЕЛ 3. АППРОКСИМАЦИЯ. МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

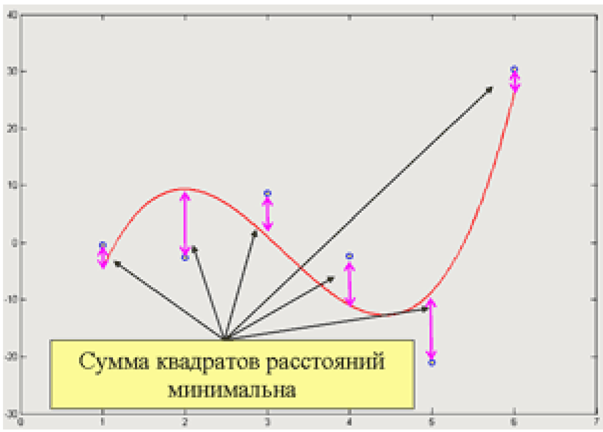
Аппроксимация (approximare – приближаться, лат.) – приближённое выражение каких-либо величин (геометрических объектов, экспериментальных данных и т.п.) через другие, известные и более удобные для обработки величины.

Среди технологий аппроксимации особую популярность получил метод наименьших квадратов (МНК). МНК заключается в том, что аппроксимирующая функция проводится так, чтобы сумма квадратов её отклонений от табличных значений по всем её узловым точкам была минимальна.

На рисунках маркерами показаны экспериментальные данные, которые аппроксимирует линейная функция.



На следующем рисунке для аппроксимации использован полином третьей степени (кубическая парабола).



Основным преимуществом МНК является возможность сглаживания данных, подверженных погрешностям.

3.1. Аппроксимация с использованием одной функции

Пусть результаты измерений представлены N парами значений (x_i, y_i). Пусть, при этом уместны следующие предположения

1) Известен общий характер функциональной зависимости между величинами y и x.

2) Эта зависимость $y = f(x)$ может быть выражена с помощью одной (как правило, элементарной) функции $g(x)$ умноженной на неизвестный коэффициент a. То есть

$$y = a \cdot g(x). \qquad (1)$$

3) Измерительный процесс подвержен большим погрешностям измерений или ошибкам.

При этих предположениях задача сводится к нахождению коэффициента a. Если бы не измерительные погрешности (п.3), для решения это задачи было бы достаточно одной пары значений (x_i, y_i) (одного измерения). Но экспериментатору, что бы уменьшить влияние отдельных ошибок, нужно использовать вычислительную схему, в которой вклад каждой ошибки уменьшается за счёт влияния других ошибок. Модели, на которых основываются такие вычислительные схемы (алгоритмы) называются *регрессионными*. Наиболее распространённой регрессионной моделью является МНК. Применим этот метод для решения поставленной задачи.

Сумма квадратов отклонений аппроксимирующей функции от табличных значений равна (см. п.2)

$$F = \sum_{i=1}^{N} (y - y_i)^2 = \sum_{i=1}^{N} (a \cdot g(x_i) - y_i)^2 . \qquad (2)$$

Полученное выражение – есть функция одной неизвестной a. Минимум этой функции определим из условия

$$\frac{\partial F}{\partial a} = 0 . \qquad (3)$$

Произведём дифференцирование функции (2) и подставим в (3)

$$\frac{\partial F}{\partial a} = \sum_{i=1}^{N} 2(a \cdot g(x_i) - y_i) \cdot g(x) = 0$$

Выполним следующие преобразования: Вынесем коэффициент 2 за знак суммы и сократим на него, разобъём сумму на две суммы, перенесём второе слагаемое-сумму в правую часть. В итоге, задача сводится к решению одного линейного уравнения с одним неизвестным a

$$a \cdot \sum_{i=1}^{N} g(x_i)^2 = \sum_{i=1}^{N} (y_i \cdot g(x_i)) \qquad (4)$$

В самом простом случае, когда $g(x) = 1$ аппроксимирующая функция равна среднему арифметическому результатов измерений

$$y = f(x) = \frac{\sum_{i=1}^{N} y_i}{N} = const$$

В случае если $g(x) = x$, после нахождения из уравнения (4) коэффициента *a*, получим аппроксимирующую функцию

$$y = f(x) = \frac{\sum_{i=1}^{N} (y_i \cdot x_i)}{\sum_{i=1}^{N} x_i^2} x$$

Ниже приведён список некоторых функций, из которого можно делать выбор при моделировании различных процессов.

$$y = \frac{a}{x},\ y = ax^n,\ y = ae^{nx},\ y = axe^{nx},\ y = a\ln(x),\ y = ax\ln(x),\ y = a\frac{\ln(x)}{x}.$$

3.2. Линейная аппроксимация

В Разделе 3.1. мы искали аппроксимирующую функцию в виде линейной зависимости $y = ax$. Особенностью такой зависимости является прохождение регрессионной линии через начало координат.

Нахождение линейной зависимости между двумя величинами в общем виде является более сложной задачей. Приступим к её решению. Запишем искомую функцию в виде

$$y = a_1 + a_2 x.$$

Задача заключается в определении коэффициентов a_1, a_2.

Рассчитаем сумму квадратов отклонений аппроксимирующей функции от табличных значений

$$F = \sum_{i=1}^{N} (y - y_i)^2 = \sum_{i=1}^{N} (a_1 + a_2 x_i - y_i)^2.$$

Полученная пара есть функция двух неизвестных a_1, a_2. Минимум её определим из условия

$$\frac{\partial F}{\partial a_1} = \frac{\partial F}{\partial a_2} = 0.$$

Получающуюся в результате дифференцирования систему линейных алгебраических уравнений запишем в матричной форме

$$\begin{pmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

где $s_{11} = N$, $s_{12} = s_{21} = \sum_{i=1}^{N} x_i$, $s_{22} = \sum_{i=1}^{N} x_i^2$, $b_1 = \sum_{i=1}^{N} y_i$, $b_2 = \sum_{i=1}^{N} x_i y_i$.

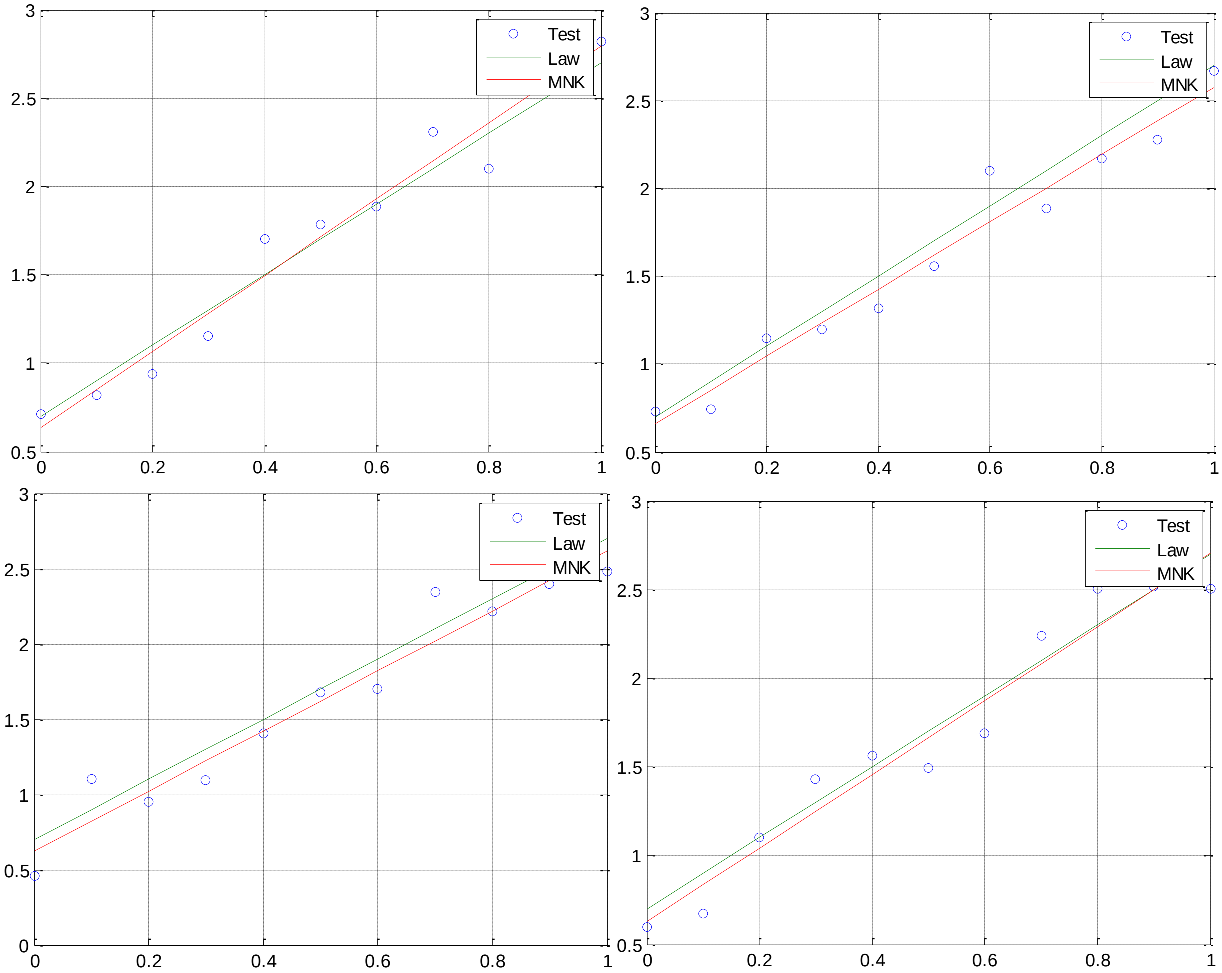
Решив систему, получим коэффициенты a_1, a_2 аппроксимирующей прямой.

Данный алгоритм, реализованный средствами MATLAB, приведён ниже. В качестве истинной выбирается прямая с коэффициентами $a_1 = 0.7$ и $a_2 = 2.0$. Экспериментальные данные получаем, искажая истинную функцию генератором случайных чисел `rand`.

```
N=11; %количество измерений
a=0;
b=1;
h=(b-a)/(N-1);
x=[0:h:1.0] %вектор-строка абсцисс измерений
x=x' %вектор-столбец абсцисс измерений
y1=0.7+2.0*x; %вектор-столбец ординат истинной прямой
y=y1+0.5*(rand(N,1)-0.5); %зашумление ординаты
%рассчитываем вектор-столбец правой части СЛАУ
s(1,1)=N;
s(2,1)=sum(x);
s(1,2)=s(2,1);
s(2,2)=sum(x.^2);
%вектор-столбец правой части СЛАУ
b=[sum(y);sum(y'*x)];
a=s\b; %решаем СЛАУ
y2=a(1)+a(2)*x; %аппроксимирующая кривая
plot(x,y,'o',x,y1,x,y2); %строим график
grid on;
legend Test Law MNK
```

Обратите внимание на отсутствие точки с запятой во второй и третьей строках.

На рисунках представлены: результаты измерения, истинная функция и аппроксимирующие кривые. Чтобы продемонстрировать работу функции `rand`, приводятся четыре реализации эксперимента.



При увеличении числа точек `N` расхождение красной и зелёной прямых уменьшается (рекомендуем поэкспериментировать).

MATLAB имеет встроенные функции, позволяющие решить данную задачу. Функция `p=polyfit(x,y,n)` рассчитывает коэффициенты полинома $p(x)$ степени n, аппроксимирующего функцию $y(x)$ в смысле МНК. Результатом работы функции является строка `p` длины $n+1$, содержащая коэффициенты аппроксимирующего полинома.

Зная строку коэффициентов полинома `p`, с помощью функции `y=polyval(p,s)` можно вычислить значения этого полинома для дискретного аргумента `s`.

В приведённом примере аппроксимирующий полином первой степени, поэтому код будет следующий:

```
N=81; %количество измерений
a=0;
b=1;
h=(b-a)/(N-1);
x=[0:h:1.0]; %вектор-строка абсцисс измерений
y1=0.7+2.0*x; %вектор-столбец ординат истинной прямой
y=y1+0.5*(rand(1,N)-0.5); %зашумление ординаты
ap=polyfit(x,y,1)
```

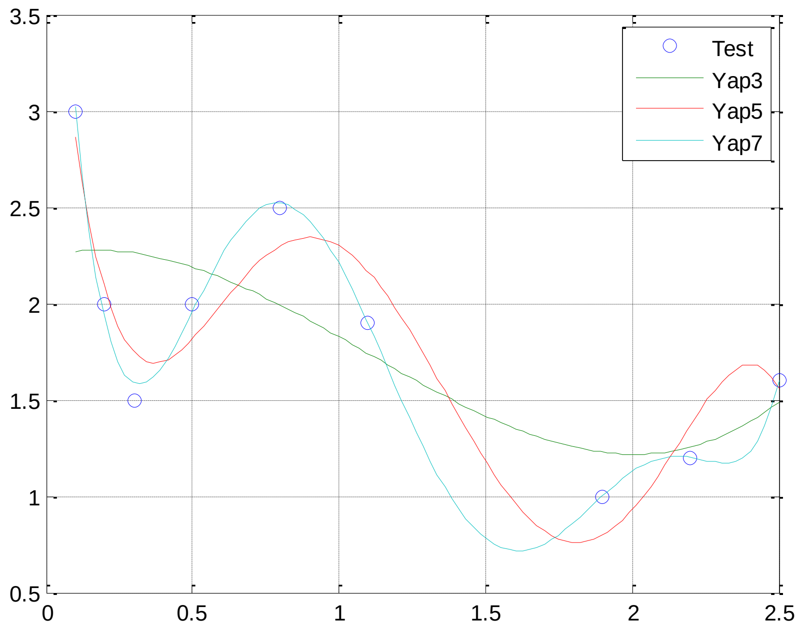
```
y2=polyval(ap,x)
plot(x,y,'o',x,y1,x,y2); %строим график
grid on;
legend Test Law MNK
```

3.3. Полиномиальная аппроксимация

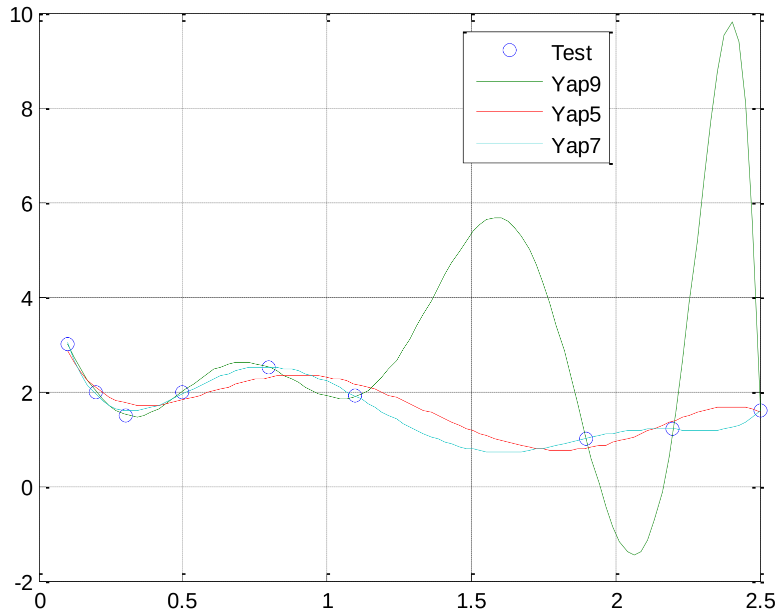
Линейная аппроксимация далеко не всегда применима. Аппроксимация в смысле МНК полиномом более высоких порядков, чем первый имеет широкую область применения. Пример такой аппроксимации полиномами 3, 5 и 7 степени, использующий встроенные функции, описанные ранее, приведён ниже.

```
clear
x=[0.1 0.2 0.3 0.5 0.8 1.1 1.9 2.2 2.5];
y=[3.0 2.0 1.5 2.0 2.5 1.9 1.0 1.2 1.6];
%вычисление коэффициентов разных порядков
p3=polyfit(x,y,3)
p5=polyfit(x,y,5)
p7=polyfit(x,y,7)
xx=linspace(x(1),x(end),100);%равномерная сетка из 100 элементов
%расчёт ординат аппроксимирующих полиномов
Yap3=polyval(p3,xx);
Yap5=polyval(p5,xx);
Yap7=polyval(p7,xx);
%строим график
plot(x,y,'o',xx,Yap3,xx,Yap5,xx,Yap7)
grid on
legend Test Yap3 Yap5 Yap7
```

Первая команда нужна для того, чтобы перед выполнением программы очистить память системы MATLAB от переменных, созданных при выполнении предыдущих задач в текущем сеансе работы с системой. Эти переменные не только занимают место в оперативной памяти, случайные совпадения с именами переменных новой задачи может привести к некорректной работе или ошибкам.



Результаты работы алгоритма приведены на рисунке. Повышение степени полинома улучшает качество аппроксимации. Возникает желание применять полином ещё более высокого порядка. Однако повышение степени приводит к обратному результату. На следующем рисунке полином третьей степени заменён полиномом девятой степени.



Удачной такую аппроксимацию назвать нельзя. На некоторых интервалах между точками полином начинает осциллировать, что объясняется погрешностью расчёта коэффициентов при высоких степенях.

3.4. Аппроксимация с использованием произвольного набора функций

В Разделе 3.2. (линейная аппроксимация) подбирались коэффициенты *a1* и *a2* при функциях *g1*(*x*)=*1* и *g2*(*x*)=*x* соответственно.

В Разделе 3.3. (полиномиальная аппроксимация) речь шла об n коэффициентах *a1*, *a2*, *a3*, …, *an* при функциях *g1*(*x*)=*1*, *g2*(*x*)=*x*, *g3*(*x*)=x^2, …, *gn*(*x*)=x^{n-1} соответственно.

Но в общем виде реализация МНК допускает произвольный набор (базис) функций например взятых из списка в конце Раздела 3.1. В практических приложениях, например при обработке сигналов, решение ищется в виде фурье-разложения с базисом: (*1*, *sin*(*x*), *sin*(*2*x*), *sin*(*3*x*).

В пакете MATLAB для произвольного базиса функций можно использовать такой же подход, что и в Разделе 3.2.

РАЗДЕЛ 4. МЕТОДЫ ИНТЕРПОЛЯЦИИ

Интерполяция – нахождение промежуточных значений по имеющемуся набору точных числовых значений. В случае если экспериментальные данные подвержены погрешностям, этот метод неприемлем.

Постановка задачи интерполяции:

На отрезке $[a,b]$ в точках $\{x_1, x_2, \ldots, x_n,\}$ известны значения функции $f(x)$. Требуется построить функцию $g(x)$, совпадающую с заданной функцией $f(x)$ в этих точках.

$$g(x_k) = f(x_k),\ k = 1, 2, \ldots, n,$$

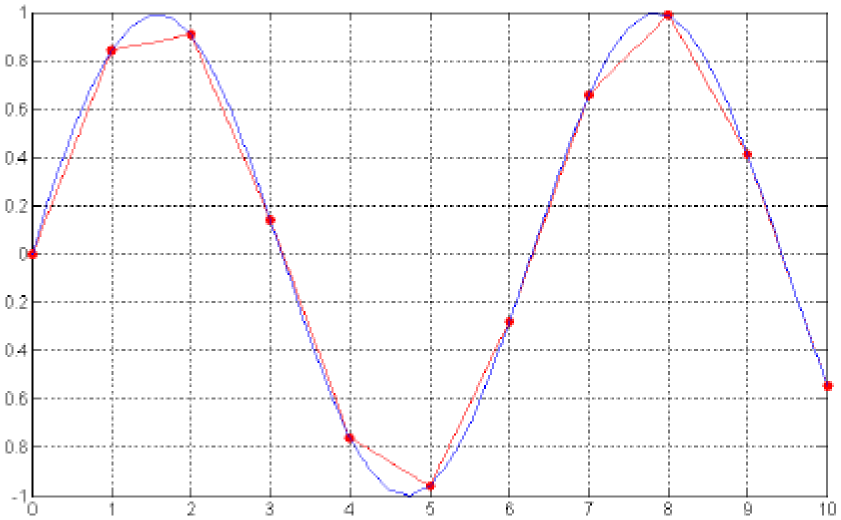
Точки $\{x_1, x_2, \ldots, x_n,\}$ называются узлами интерполяции, а построение функции – интерполированием.

Линейная интерполяция заключается в том, что узлы соединяются прямолинейными отрезками. Два соседних узла позволяют рассчитать параметры этих отрезков. Построенная таким образом функция $g(x)$ – ломанная с вершинами в узлах. Точность интерполяции в промежуточных точках невысока. Для практических приложений больший интерес представляют гладкие функции, то есть имеющие непрерывные производные.

Интерполяция полиномом высокой степени на всём отрезке называется глобальной. Так для интерполяции по трём точкам потребуется полином второй степени (парабола), по четырём точкам – полином третей степени (кубическая парабола), по пяти точкам – четвёртой и так далее по принципу:

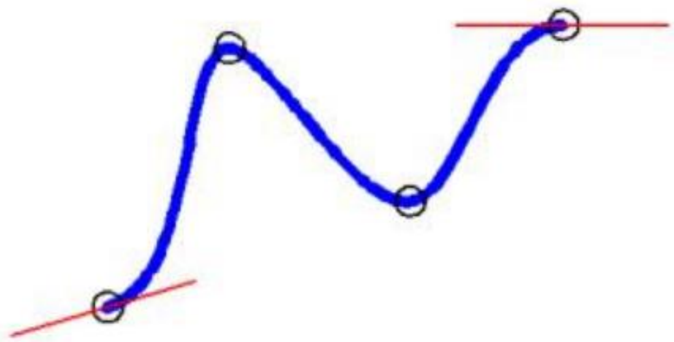
степень полинома на единицу меньше количества точек. Теоретически точность интерполяции возрастает с ростом степени полинома, однако на практике погрешность расчёта коэффициентов при высоких степенях приводят к осцилляциям интерполяционной кривой (как это было в Разделе 3.3., когда МНК реализовывался на базе полинома девятой степени).

На рисунке показан пример линейной и глобальной интерполяции. Функция, задаваемая линейной интерполяцией, является кусочно-непрерывной (то есть в конечном множестве точек производная терпит разрыв). Преимуществом глобальной интерполяции является гладкость (то есть непрерывной является не только сама функция, но и её производные нескольких первых порядков).



Идея локальной интерполяции заключается в том, что между соседними узлами строится свой отдельный интерполяционный полином невысокой степени. Затем эти полиномы «сшиваются». Такой метод называется сплайн-интерполяцией. Если использовать полином третьего порядка, то в узлах можно удовлетворить как условию непрерывности функции, так и непрерывности её первой и второй производных.

Название метода сплайн-интерполяции обязано гибкой линейке, рейке, зафиксированной в отдельных точках, которая называется сплайном. Рейка – физическая модель сплайн-функции, в свою очередь сплайн-функция – математическая модель рейки (см. рисунок).



Функция MATLAB `interp1()` строит интерполирующую кривую для одномерного массива. Синтаксис функции

`yi=interp1(x,y,xi,'metod')`, где

`x` – массив узлов (абсцисс экспериментальных данных),

`y` – массив значений функции (ординат экспериментальных данных),

`xx` – массив значений абсцисс, в которых вычисляются значения интерполирующей функции

`metod` – определяет метод построения сплайна. Принимает следующие значения:

1. `nearest` – ступенчатая (0-ой степени) интерполяция по соседним точкам – этот метод построения кусочной функции, при котором значение в любой точке равно значению в ближайшее узловой точке;
2. `linear` – линейная (1-ой степени) сплайн-интерполяция;
3. `cubic` – интерполяция кубическим полиномом;
4. `spline` – интерполяция кубическим сплайном;
5. `pchip` – интерполяция кубическим эрмитовым сплайном.

Так же в пакете MATLAB разработана функция для построения сплайн-интерполяции в несколько более простом виде:

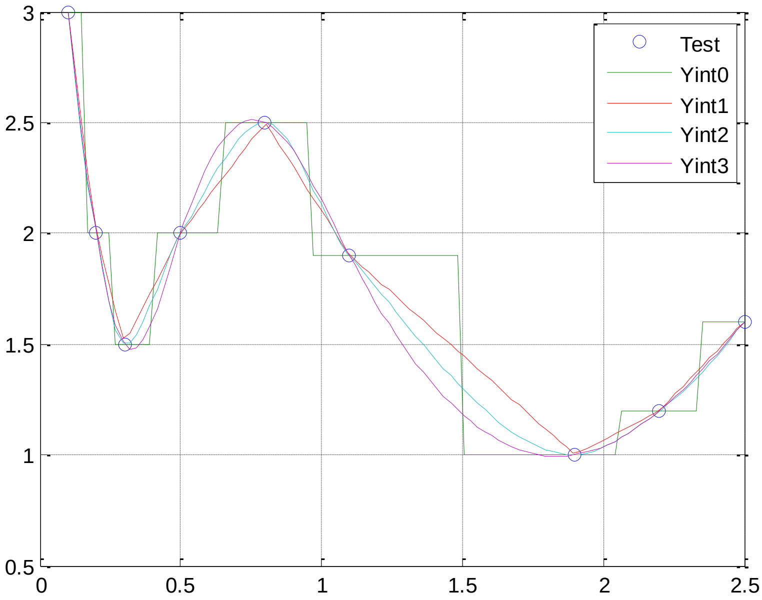
`yy=spline(x,y,xx)`

Как видите, при вызове этой функции, не требуется указания дополнительного параметра.

Технология сравнения методов анализа экспериментальных данных (из Раздела 3.3.) представлена нижеследующем листинге

```
clear
x=[0.1 0.2 0.3 0.5 0.8 1.1 1.9 2.2 2.5];
y=[3.0 2.0 1.5 2.0 2.5 1.9 1.0 1.2 1.6];
xx=linspace(x(1),x(end),100);%равномерная сетка из 100 элементов
```

```
Yint0=interp1(x,y,xx,'nearest'); %промежуточные значения
ступенчатой интерполяции
Yint1=interp1(x,y,xx,'linear'); %промежуточные значения линейной
интерполяции
Yint2=interp1(x,y,xx,'cubic'); %промежуточные значения
интерполяции кубическим полиномом
Yint3=interp1(x,y,xx,'spline'); %промежуточные значения
интерполяции кубическим сплайном
Yint3=interp1(x,y,xx,'spline'); %промежуточные значения
интерполяции кубическим сплайном
Yint4=spline(x,y,xx)
%строим график
plot(x,y,'o',xx,Yint0,xx,Yint1,xx,Yint2,xx,Yint3)
grid on
legend Test Yint0 Yint1 Yint2 Yint3
sum(abs(Yint3-Yint4))%оценка расхождения методов interp1() и
spline()
```



Проанализируйте результаты.

И, на последок, поменяйте четвёртую строку

`xx=linspace(x(1)-0.5,x(end)+2.0,100);`

Прогнозирование значений функции вне интервала значений независимого аргумента называется *экстраполяцией*.

Поэкспериментируйте с интервалами экстраполяции. Сделайте выводы. Повторите серию экспериментов с расширенным интервалом аргумента для скрипта из Раздела 3.3.

РАЗДЕЛ 5. ПЛАН ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. Воспроизведите п.1 и п.2 из задания к Лабораторной работе №1. В результате должны появиться скалярные переменные `W1`, `W2` и два множества `t` и `A`.

   Функция *A*(*t*) в Лабораторной работе №2 является основой для начала моделирования.

   Результаты аппроксимации и интерполяции будем сравнивать с этой функцией.

2. Сформируйте диапазон `t0` от `a=1` до `b=3` с шагом `h0=0.4`. Рассчитайте (на листочке или в уме) количество узлов `N` и задайте диапазон `t0` как в Разделах 3.2. и 3.3. (это пригодится в п.5 для повышения уровня дискретизации, при этом диапазон `t0` будет пересчитываться автоматически).

   Сформируйте дискретную функцию *A0*=*A*(*t0*).

3. Постройте на одной области непрерывный график функции `(A,t)` и точечный (с круглыми маркерами) для функции `(A0,t0)`. Графики должны быть сопровождены легендами, как, впрочем, и в остальных пунктах.

4. Постройте на основе МНК аппроксимирующие функции `Yap3`, `Yap5`, `Yap7`, `Yap9` для множеств `(A0,t0)` на диапазоне изменения дискретного аргумента `t`, как (см. Раздел 3.3.). Постройте графики полученных функций совместно с функциями п.2. Сделайте выводы.

5. Постройте аппроксимирующие функции `Yint0`, `Yint1`, `Yint2`, `Yint3` для множеств `(A0,t0)` на диапазоне изменения дискретного аргумента `t`, как (см. Раздел 4.). Постройте графики полученных функций совместно с функциями п.2. Сделайте выводы.

6. Повторите действия в пп. 2–4 с шагом `h=0.2`, а затем с шагом `h=0.1`. Сделайте выводы. Графики и выводы добавьте к отчёту

7. Постройте экстраполирующие графики при минимальном уровне дискретизации независимого аргумента `t0` (то есть при `h=0.1`). Диапазон независимого аргумента `t` увеличьте справа и слава на 0.5. То есть сделайте замену в одной из первых строчек:

   ```
   t=[0.5:0.01:3.5];
   ```
   Сделайте выводы.

РАЗДЕЛ 6. ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Отчет по лабораторной работе должен содержать следующее.

1. Титульный лист.

2. Цель Лабораторной работы.

3. Вариант задания.

4. Полный компилируемый листинг реализованной программы.

5. Графики функций с легендами.

6. Выводы.

РАЗДЕЛ 7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чём заключается задача аппроксимации экспериментальных данных?
2. В чём заключается задача интерполяции экспериментальных данных?
3. В чём заключается МНК?
4. Что такое линейная аппроксимация?
5. Что такое полиномиальная аппроксимация?
6. Что такое линейная интерполяция?
7. Что такое глобальная интерполяция?
8. Что такое сплайн-интерполяция?
9. В чём заключается задача экстраполяции экспериментальных данных?

Дополнительную информацию о реализации методов интерполяции и аппроксимации в среде MATLAB можно получить, например, воспользовавшись ресурсом

https://studfiles.net/preview/6654106/

или основной библиографией курса.