

## Лабораторная работа №3. Приближение данных по методу наименьших квадратов в MatLab

### А) Приближение данных полиномиальным МНК

#### 1. Приближение данных полиномиальным МНК при помощи функции **polyfit**

1.1. Построить график полинома по варианту (файл 3. Варианты полинома.pdf), подобрать отрезок, содержащий все корни	<code>a=-10; b=10; x=a:0.05:b; p=[1,2,-3,4]; y=polyval(p,x); plot(x,y,'b');</code>
1.2. Создать на отрезке сетку из 150 узлов	<code>x=linspace(a,b,150);</code>
1.3. Создать набор данных с ошибками	<code>y=polyval(p,x)+randn(size(x));</code>
1.4. Найти коэффициенты приближающего полинома 1 степени	<code>p1=polyfit(x,y,1);</code>
1.5. Построить данные, график полинома 1 степени и график ошибки – разности между данными и полиномом	<code>t=linspace(a,b,1001); y1=polyval(p1,t); subplot(2,1,1); plot(x,y,'b.',t,y1,'r') err1=y-polyval(p1,x); subplot(2,1,2); plot(x,err1,'r.')</code>
1.6. Найти коэффициенты и построить график полинома 2 степени и график ошибки	<code>p2=polyfit(x,y,2); y2=polyval(p2,t); subplot(2,1,1); plot(x,y,'b.',t,y2,'g') err2=y-polyval(p2,x); subplot(2,1,2); plot(x,err2,'g.')</code>
1.7. Аналогично для полиномов 3 и 4 степени	<code>p3=... p4=...</code>

**Вывод:** Ошибка НЕслучайна в случае, когда степень приближающего полинома меньше степени исходного полинома. Ошибка становится случайной, когда степень приближающего полинома равна или чуть больше степени исходного полинома.

#### 2. Приближение данных полиномиальным МНК в приложении **cftool**

2.1. Запустить приложение из командного окна	<code>&gt;&gt; cftool</code>
2.2. Импортировать данные из рабочей среды	X data: x Y data: y
2.3. Выбрать способ приближения полиномами и степень полинома	Polynomial Degree: 1
2.4. Показать график ошибки	в меню View поставить галочку на Residuals Plot
2.5. Аналогичные действия проделать для полиномов 2, 3 и 4 порядков. Убедиться, что графики получились такие же, как и при помощи функции <b>polyfit</b>	

**Вывод:** Чем больше степень приближающего полинома, тем точнее описываются ошибки в данных. Чему равна ошибка, если степень полинома на 1 меньше количества точек?

### Б) Приближение данных с выбросами

#### 3. Приближение данных с выбросами в приложении **cftool**

Создать выбросы в данных и приблизить их полиномом 3ей степени

3.1. В командном окне добавить в данные выбросы (например, взяв максимальное значение данных)	<code>&gt;&gt; y1=y; y1(1:10:end)=15;</code>
3.2. Импортировать новые значения y	Y data: y1
3.3. Выбрать адаптивный МНК – веса обратно пропорциональны ошибкам в 4 степени	Robust: Bisquare
3.4. Адаптивный МНК: LAR – least absolute residual (минимальная сумма модулей уклонов)	Robust: LAR

**Вывод:** выбросы в исходных данных можно учитывать по-разному. В зависимости от способа учета выбросов можно получить разные результаты – итоговое приближение

## В) Приближение данных сглаживающими сплайнами

### 4. Приближение данных сглаживающими сплайнами в **cftool**

Приблизить данные с ошибками (без выбросов) сглаживающим сплайном с различной степенью сглаживания

4.1. Импортировать старые значения $y$	<code>Y data: y</code>
4.2. Выбрать способ приближения сглаживающим сплайном	<code>Smoothing Spline</code>
4.3. Меняя параметр сглаживания от 0 до 1 проследить за изменениями графиков сплайна и ошибки	<code>Smoothing Parameter</code>

### 5. Приближение данные сглаживающим сплайном при помощи функции **csaps**

5.1. Создание и построение данных с ошибками	<code>a=-3; b=2; p=[1,2,-3,4]; x=linspace(a,b,150); y=polyval(p,x)+randn(size(x))*0.5; plot(x,y,'b.');</code> <code>hold on</code>
5.2. Создание структуры сглаживающего сплайна (0.8 – параметр сглаживания)	<code>pp = csaps(x, y, 0.8);</code>
5.3. Вычисление значений и построение сплайна	<code>yt = fnval(pp, t); plot(t, yt,'r')</code>

**Выводы:** данные с ошибками можно приблизить сглаживающими сплайнами. В зависимости от параметра сглаживания приближение меняется от полинома 1ой степени, совпадающего с приближением по МНК до интерполяционного кубического сплайна

## Г) Приближение данных нелинейными моделями МНК

### 6. Приближение данных нелинейными моделями МНК в приложении **cftool**

6.1. Построить график функции по варианту (файл 3. Варианты функции.pdf), задать параметры, выбрать отрезок	<code>a0=-3; b0=4; x=a0:0.05:b0; a = 0.82; b = 4.15; y = 2*exp(a*x).*sin(b*x); plot(x,y,'b');</code>
6.2. Создать на отрезке сетку из 150 узлов и набор данных с ошибками	<code>x=linspace(a0,b0,150); y=2*exp(a*x).*sin(b*x)+randn(size(x));</code>
6.3. Импортировать данные в <b>cftool</b>	<code>X data: x Y data: y</code>
6.4. Приблизить данные полиномами различных степеней, проследить за ошибкой	<code>Polynomial Degree: 6</code>
6.5. Создать нелинейную модель с двумя вычисляемыми параметрами	<code>Y=f(x) = 2*exp(a*x).*sin(b*x)</code>
6.6. Нажав кнопку <b>Fit options</b> , взять за начальные значения параметров целые числа, ближайшие к исходным	<code>Coefficients StartPoint a 1 b 4</code>

**Выводы:** Полиномиальные модели МНК часто не подходят для приближения функций, в которую коэффициенты входят нелинейно. Неверный выбор начальных значений параметров может сильно повлиять на конечный результат

### 7. Приближение данных нелинейными моделями МНК при помощи функции **fit**

7.1. Задание отрезка, сетки, коэффициентов и набора данных с ошибками	<code>a0=-3; b0=4; x=linspace(a0,b0,150); a = 0.82; b = 4.15; y=2*exp(a*x).*sin(b*x)+randn(size(x));</code>
7.2. Задание нелинейной модели как символьной функции	<code>NlModel = '2*exp(a*x).*sin(b*x)';</code>
7.3. Задание начальных значений	<code>StartPoint=[1,4];</code>
7.4. Создание структуры сглаживающего сплайна	<code>f1 = fit(x',y',NlModel,'Start', StartPoint)</code>
7.5. Построение сплайна и набора данных	<code>plot(f1,x,y)</code>