Пловдивски университет „Паисий Хилендарски”
Факултет по математика и информатика
Катедра “Приложна математика и моделиране”

“Компютърни числени методи”
ЛЕКЦИЯ 6

Проф. д-р Снежана Гочева-Илиева, snow@uni-plovdiv.bg

Он-лайн обучение: www.fmi-plovdiv.org/evlm
www.fmi-plovdiv.org/evlm/DBbg - числени методи

Литература:

1. Бояджиев Д., Гочева С., Макрелов И., Попова Л. – Ръководство по числени методи – част 1, Издания: 2003, 2006, 2010.
2. Семерджиев Х., Боянов Б., Числени методи, ПУ.
3. Гочева-Илиева С., Въведение в система Mathematica, ЕксПрес, Габрово, 2009.

Съдържание

Метод на най-малките квадрати
Интерполационни сплайни – линеен, квадратичен

1. Приближаване на експериментални данни

Често при провеждане на експерименти се натрупва голямо количество таблици от данни, за която експериментаторът желае да установи аналитичен закон (формула), която възможно най-добре да описва данните. Обикновено по характер тези данни са получени с голяма неотстранима грешка, най-вече зависеща от точността на прибора, методиката на измерване и други фактори. Ще разгледаме случая, когато търсим да приближим реална функция на една реална променлива.

Изходна постановка. Нека функцията $y = f(x)$ е дефинирана в някакъв интервал, в който е известна таблица от стойностите й

x_i	x_1	x_2	...	x_N
y_i	y_1	y_2	...	y_N

Тук точките $x_1, x_2, \dots, x_N$ не е задължително да са различни.

Да предположим, че е избрана система от базисни функции $\varphi_0(x), \varphi_1(x), \dots, \varphi_m(x)$ и се търси приближение от вида

$$f(x) \approx c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_m\varphi_m(x). \qquad (1)$$

Този тип приближение е линейна комбинация на избраните m+1 на брой базисни функции $\varphi_j(x), j = 0,1,\dots m$ и се нарича обобщен полином. Очевидно (1) е линеен математически модел спрямо базисните функции. Коефициентите $c_j, j = 0,1,\dots m$ са неизвестни.

Примери за базисни функции:

1) Най-често това са едночлените $\varphi_j(x) = x^j, \; j = 0,1,\dots m$.

Така приближаващата функция като тяхна линейна комбинация е алгебричен полином $P_m(x)$ от степен m, т.е.

$$f(x) \approx P_m(x) = c_0 + c_1 x + \dots + c_m x^m \qquad (2)$$

2) Ако данните наподобяват тригонометрична функция, можем да изберем базисни полиноми $\varphi_j(x) = \sin(jx),\ j = 0,1,\dots m$. Тогава

$$f(x) \approx T_m(x) = c_0 + c_1 \sin(x) + c_2 \sin(2x) + \ldots + c_m \sin(mx). \tag{3}$$

3) Друг вариант е експоненциалната апроксимация

$$f(x) \approx E_m(x) = c_0 e^{\lambda_0 x} + c_1 e^{\lambda_1 x} + \ldots + c_m e^{\lambda_m x}, \tag{4}$$

където λ_j са зададени (а ако не са – става нелинеен модел).

2. Метод на най-малките квадрати (МНМК) с алгебрични полиноми

Основно предположение: нека $m \ll N$, т.е. броят на данните N е много по-голям от степента m на обобщения полином.

Означаваме разликата (остатъка) между стойностите на функцията y_i и полинома $P_m(x)$ за всяка точка x_i с

$$r_i = P_m(x_i) - y_i = c_0 + c_1 x_i + c_2 x_i^2 + \dots c_m x_i^m - y_i, \quad i = 1, \dots, N. \qquad (5)$$

Критерият за намиране на коефициентите c_j в (1) по МНМК е сумата от квадратите на всички остатъци да бъде възможно най-малка. Стигаме до задачата: Измежду всички полиноми от дадена степен m да се намери този, за който

$$\sum_{i=1}^{N} r_i^2 = \min. \qquad (6)$$

Минимумът тук зависи от $c_0, c_1, \ldots c_m$. По-подробно от (5) и (6) трябва да намерим минимума на функцията с променливи c_j:

$$\Phi(c_0, c_1, \ldots, c_m) = \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} (c_0 + c_1 x_i + c_2 x_i^2 + \ldots c_m x_i^m - y_i)^2 . \tag{7}$$

От математическия анализ е известно, че необходимото условие за екстремум на диференцируема функция на много променливи е всички частни производни да са равни на нула, т.е.

$$\left|\begin{array}{l} \dfrac{\partial \Phi(c_0, c_1, \ldots, c_m)}{\partial c_0} = 0 \\ \dfrac{\partial \Phi(c_0, c_1, \ldots, c_m)}{\partial c_1} = 0 \\ \\ \ldots \\ \\ \dfrac{\partial \Phi(c_0, c_1, \ldots, c_m)}{\partial c_m} = 0 \end{array}\right. . \qquad (8)$$

За първото уравнение от (7) диференцираме спрямо c_0 при фиксирани останали коефициенти c_j (тук x_i, y_i са дадени). Имаме:

$$\frac{\partial \Phi}{\partial c_0} = 2\sum_{i=1}^{N} (c_0 + c_1 x_i \ + c_2 x_i^2 + \ldots c_m x_i^m - y_i) = 0 .$$

За второто уравнение аналогично:

$$\frac{\partial \Phi}{\partial c_1} = 2\sum_{i=1}^{N}(c_0 + c_1 x_i \ + c_2 x_i^2 + \ldots c_m x_i^m - y_i).x_i = 0, \ \ldots$$

$$\frac{\partial \Phi}{\partial c_m} = 2\sum_{i=1}^{N}(c_0 + c_1 x_i \ + c_2 x_i^2 + \ldots c_m x_i^m - y_i).x_i^m = 0 \ .$$

След привеждение пред неизвестните стигаме до системата

$$\left| \begin{array}{l} Nc_0 \quad + \left(\sum_{i=1}^{N} x_i\right) c_1 + \left(\sum_{i=1}^{N} x_i^2\right) c_2 + \ldots + \left(\sum_{i=1}^{N} x_i^m\right) c_m = \sum_{i=1}^{N} y_i \\ \left(\sum_{i=1}^{N} x_i\right) c_0 + \left(\sum_{i=1}^{N} x_i^2\right) c_1 + \ldots \qquad + \left(\sum_{i=1}^{N} x_i^{m+1}\right) c_m = \sum_{i=1}^{N} x_i y_i \\ \ldots \\ \left(\sum_{i=1}^{N} x_i^m\right) c_0 + \left(\sum_{i=1}^{N} x_i^{m+1}\right) c_1 + \ldots \qquad + \left(\sum_{i=1}^{N} x_i^{2m}\right) c_m = \sum_{i=1}^{N} x_i^m y_i \end{array} \right. \ . \qquad (9)$$

Очевидно това е линейна система за намиране на $c_0, c_1, \dots c_m$.

Може да се покаже, че тази система има единствено решение. При не-много големи *m* тя може да се решава по метода на квадратния корен или по метода на Гаус. При голяма степен *m* на полинома е известно, че (9) е лошо обусловена, т.к. детерминантата е малко число. Затова МНМК с алгебрични полиноми от вида (2) се прилага за $m \le 5$. За по-големи *m* се избират система ортогонални полиноми или друга система базисни функции.

Определение. Полиномът $P_m^*(x) = c_0^* + c_1^* x + \ldots + c_m^* x^m$, чиито коефициенти $c_0^*, c_1^*, \ldots, c_m^*$ са решения на (9) се нарича полином на най-добро приближение по МНМК към функцията $f(x)$.

Определение. Грешка на приближението по МНМК или средноквадратична грешка се нарича отклонението на полинома на най-добро приближение по МНМК

$$\delta_m = \sqrt{\sum_{i=1}^{N} r_i^2} = \sqrt{\sum_{i=1}^{N} (c_0^* + c_1^* x_i + c_2^* x_i^2 + \ldots c_m^* x_i^m - y_i)^2} . \qquad (10)$$

Как се избира най-добрата степен *m* за полинома по МНМК? Това в общия случай е труден проблем. Обикновено постъпват така:

Нека данните имат неотстранима грешка ε. Започва се с $m=1$, изчислява се δ_1. Ако $\delta_1 \approx \varepsilon$, значи решението е добро. Ако ср.кв. грешка не е приблизително равна на ε, се продължава с $m=2$ и δ_2 и т.н., докато се намери оптималното m, за което $\delta_m \approx \varepsilon$. Ако има резки скоци и не може да се опредли m, то се избират други базисни функции и т.н.

2.1 Пример. Приближение на данни с МНМК с Mathematica

http://www.fmi-plovdiv.org/evlm/DBbg/database/numan/leastsquares_BG/index.html

Задача 1. **Дадена е функцията** $f(t) = t + \sin(t^2)$ **в интервала** [1, 2].

а) Да се табулира функцията в 11 точки в дадения интервал.

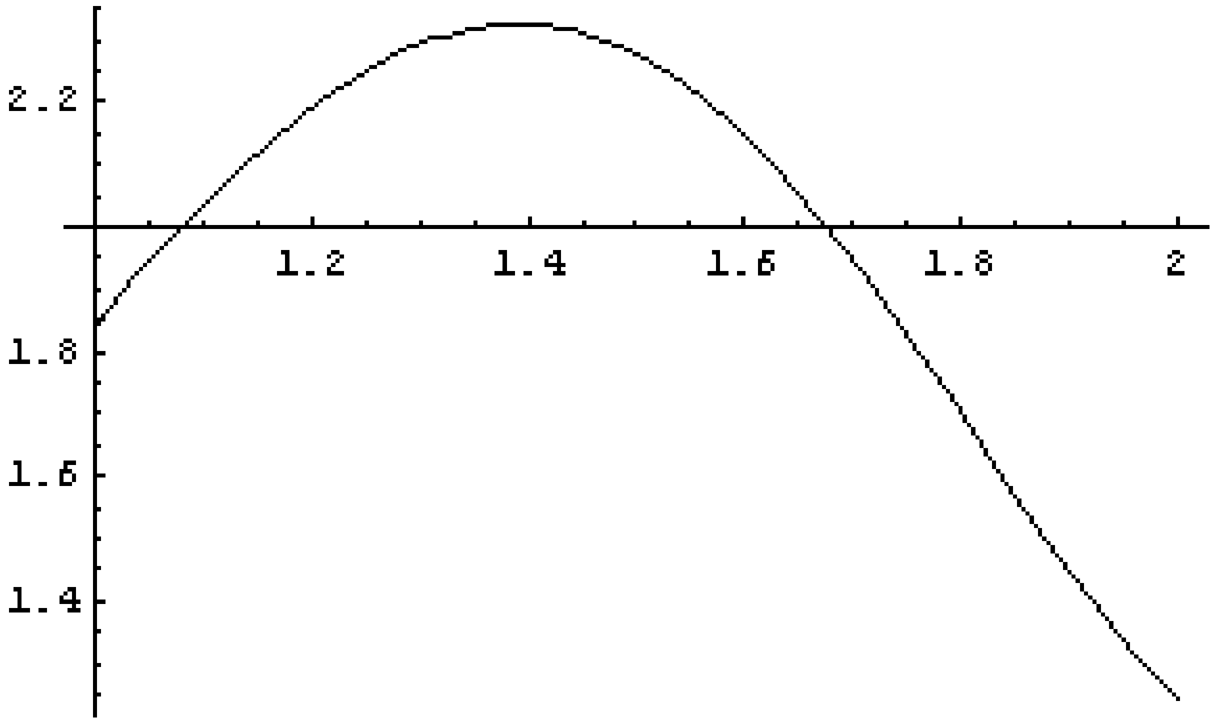
б) Да се намери полиномът на най – добро приближение по МНМК от първа степен.

в) Да се оцени грешката на приближението.

Решение :

Дефинираме функцията и начертаваме графиката й в интервала [1, 2] :

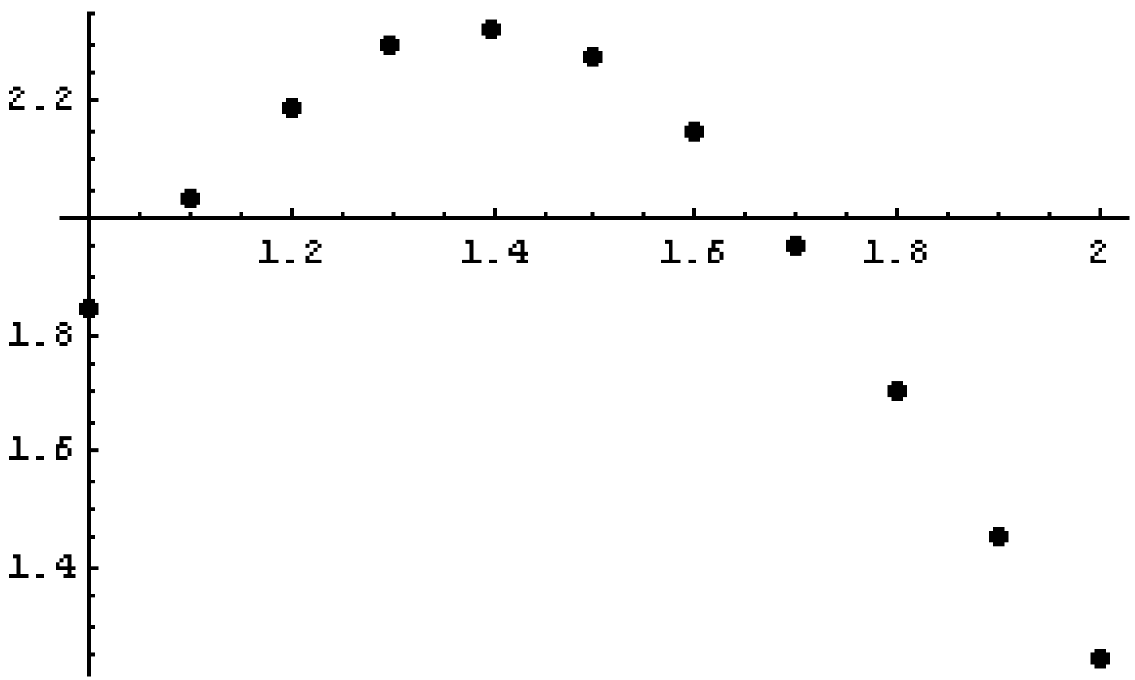
```
f[t_] := t + Sin[t^2]
a = 1.; b = 2.;
gf = Plot[f[t], {t, a, b}]
```



а) Пресмятаме стойностите на масив от точки x_i, $y_i = f[x_i]$, $i = 1, 2, \ldots n1$,
като избираме $n1 = 11$ равноотдалечени точки със стъпка h в интервала [a, b] :

```
n1 = 11;  h = (b - a)/(n1 - 1); Print["Step h=  ", h]
x = Table[a + h * (i - 1), {i, 1, n1}]; Print["x=", x]
y = Table[f[x[[i]]], {i, 1, n1}]; Print["y=", y]
xy = Table[{x[[i]], y[[i]]}, {i, 1, n1}];
gy = ListPlot[xy, PlotStyle → PointSize[0.02]]
```

```
Step h= 0.1
x={1., 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.}
y={1.84147, 2.03562, 2.19146, 2.2929, 2.32521, 2.27807, 2.14936, 1.94895, 1.70175, 1.44853, 1.2432}
```



б) Пресмятаме всички суми, необходими за МНМК от първа степен, решаваме получената система с LinearSolve и извличаме c0 и c1 :

$$s0 = n1;\ s1 = \sum_{i=1}^{n1} x_{[[i]]};\ s2 = \sum_{i=1}^{n1} x_{[[i]]}{}^2;\ d1 = \sum_{i=1}^{n1} y_{[[i]]};\ d2 = \sum_{i=1}^{n1} x_{[[i]]} * y_{[[i]]};$$

$$s = \begin{pmatrix} s0 & s1 \\ s1 & s2 \end{pmatrix}$$

$$d = \begin{pmatrix} d1 \\ d2 \end{pmatrix}$$

```
c = LinearSolve[s, d]
```

$$c0 = c_{[[1]][[1]]}$$

$$c1 = c_{[[2]][[1]]}$$

```
{{11, 16.5}, {16.5, 25.85}}
```

```
{{21.4565}, {31.4175}}
```

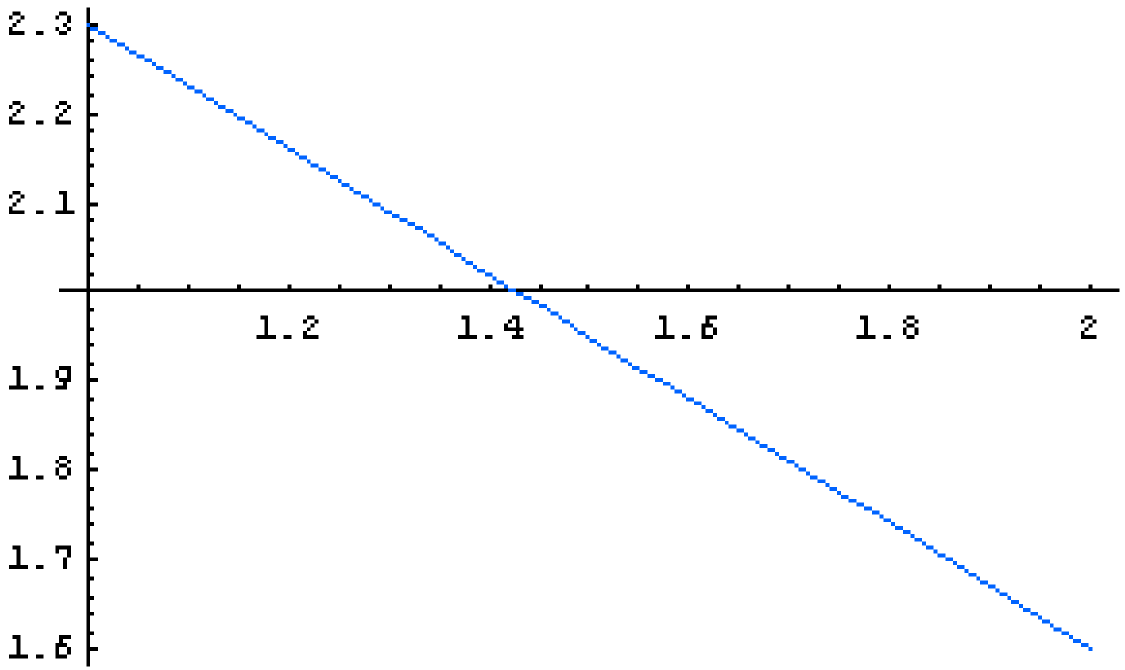
```
{{2.99685}, {-0.697508}}
```

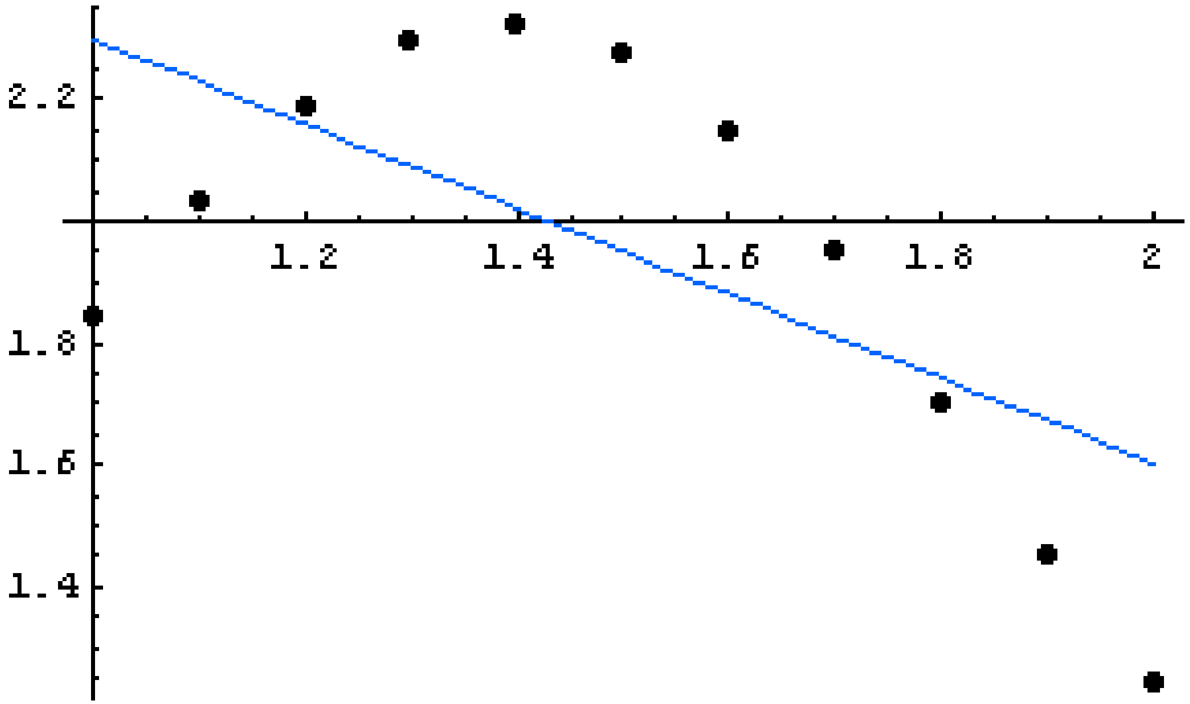
```
2.99685
```

```
-0.697508
```

Дефинираме като функция полинома на най – добро приближение от 1 – ва степен по МНМК и рисуваме графиката му. Накрая показваме с Show едновременно графиките на таблицата на f[t] и приближението му p1[t] :

```
p1[t_] := c0 + c1 * t
gp1 = Plot[p1[t], {t, a, b}, PlotStyle → Hue[.6]]
Show[gy, gp1]
```





в) Изчисляваме средноквадратичната грешка на приближението :

$$\delta_1 = \sqrt{\sum_{i=1}^{n1} (f[x_{[[i]]}] - p1[x_{[[i]]}])^2}$$

0.87215

<u>Заключение:</u>   Получената грешка очевидно е голяма, както се вижда и от графиката на двете функции. Може да се потърси приближение с полином от по-висока степен.

Задача 2. За функцията от задача 1 :

а) Да се намери полиномът на най – добро приближение по МНМК от втора степен.

в) Да се оцени грешката на приближението.

Решение :

а) Пресмятаме всички суми, необходими за МНМК от втора степен, решаваме получената система с LinearSolve и извличаме коефициентите на полинома, означаваме ги с v0, v1 и v2 :

```
s0 = n1; s1 = Sum[x[[i]], {i, 1, n1}]; s2 = Sum[x[[i]]^2, {i, 1, n1}]; s3 = Sum[x[[i]]^3, {i, 1, n1}]; s4 = Sum[x[[i]]^4, {i, 1, n1}];
d1 = Sum[y[[i]], {i, 1, n1}]; d2 = Sum[x[[i]]*y[[i]], {i, 1, n1}]; d3 = Sum[x[[i]]^2*y[[i]], {i, 1, n1}];
```

$$s0 = n1;\ s1 = \sum_{i=1}^{n1} x_{[[i]]};\ s2 = \sum_{i=1}^{n1} x_{[[i]]}^{2};\ s3 = \sum_{i=1}^{n1} x_{[[i]]}^{3};\ s4 = \sum_{i=1}^{n1} x_{[[i]]}^{4};$$

$$d1 = \sum_{i=1}^{n1} y_{[[i]]};\ d2 = \sum_{i=1}^{n1} x_{[[i]]} * y_{[[i]]};\ d3 = \sum_{i=1}^{n1} x_{[[i]]}^{2} * y_{[[i]]};$$

$$s = \begin{pmatrix} s0 & s1 & s2 \\ s1 & s2 & s3 \\ s2 & s3 & s4 \end{pmatrix}$$

$$d = \begin{pmatrix} d1 \\ d2 \\ d3 \end{pmatrix}$$

```
c = LinearSolve[s, d]
```

$$v0 = c_{[[1]][[1]]}$$

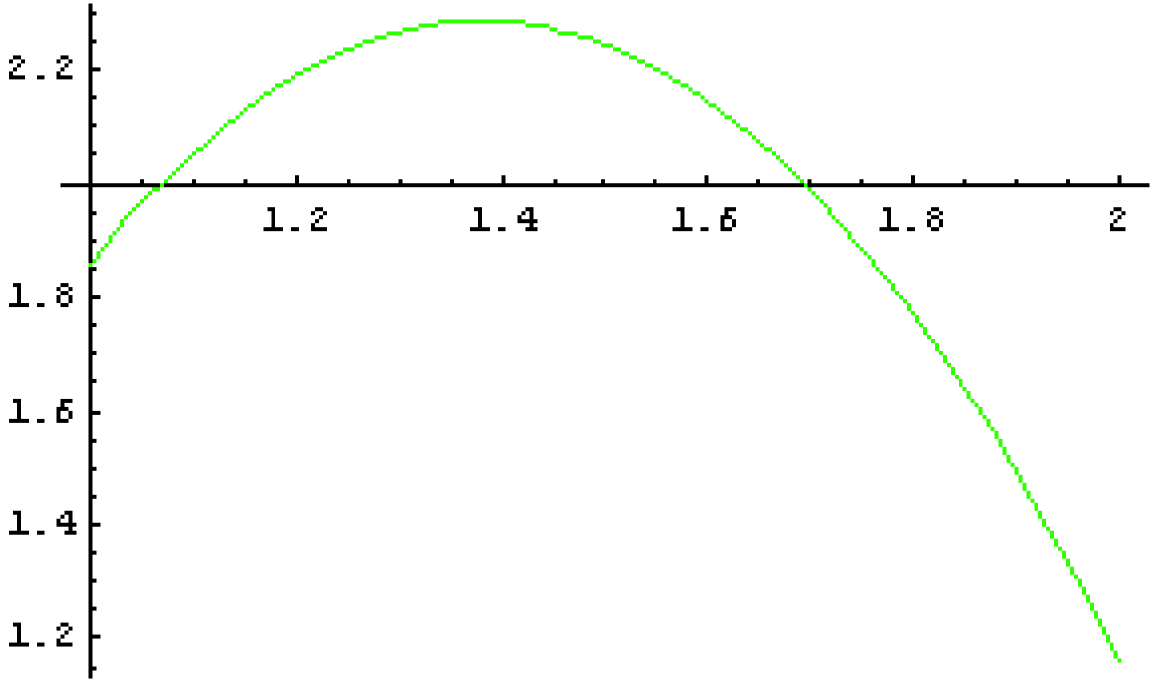
$$v1 = c_{[[2]][[1]]}$$

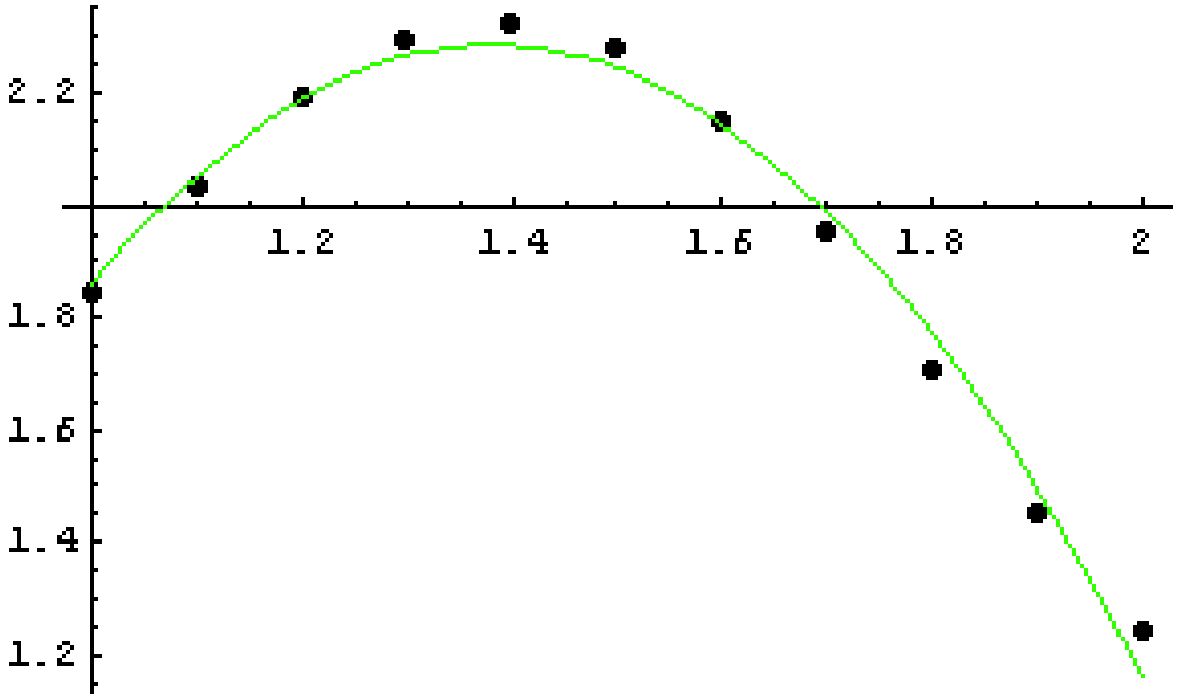
$$v2 = c_{[[3]][[1]]}$$

```
{{11, 16.5, 25.85}, {16.5, 25.85, 42.075}, {25.85, 42.075, 70.7333}}
{{21.4565}, {31.4175}, {47.8688}}
{{-3.32315}, {8.1211}, {-2.93954}}
-3.32315        8.1211        -2.93954
```

Дефинираме като функция полинома на най – добро приближение от 2 – ра степен по МНМК и рисуваме графиката му. Накрая показваме с Show едновременно графиките на таблицата на f[t] и приближението му – p2[t] :

```
p2[t_] := v0 + v1 * t + v2 * t^2
gp2 = Plot[p2[t], {t, a, b}, PlotStyle → Hue[.3]]
Show[gy, gp2]
```





в) Изчисляваме средноквадратичната грешка на приближението :

$$\delta_2 = \sqrt{\sum_{i=1}^{n1} (f[x_{[[i]]}] - p2[x_{[[i]]}])^2}$$

0.138776

<u>Заключение:</u> **Получената грешка очевидно е все още голяма, но по-приемлива от предишното приближение (сравни графиките на двете функции). Може да се потърси приближение с полином от още по-висока степен. -→ прегледайте уебстраницата на примера.**

2.2 Пример. Приближение с МНМК с Mathematica – данни за лазер с пàри на меден бромид

http://www.fmi-plovdiv.org/evlm/DBbg/database/numan/MNMK_CuBr_laser/2leastsquares_CuBr.html

3. Интерполационни сплайни

3.1 Постановка на задачата

Нека $y = f(x)$ е функция, дефинирана в интервала $[a,b]$ и е известна таблица от стойности $y_i = f(x_i)$ в точките (възлите) $a \le x_0 < x_1 < x_2 < \ldots < x_n \le b$. Стъпките между x_{i-1} и x_i ще означаваме с $h_i = x_i - x_{i-1}$. Нека таблицата е:

x_i	x_0	x_1	…	x_n
y_i	y_0	y_1	…	y_n

Определение. Интерполационният сплайн $S_k(f,x)$ от ред k е функция със следните свойства:

(1) $S_k(f,x)$ е полином $f_i(x)$ от степен k във всеки подинтервал $[x_{i-1}, x_i]$, $i = \overline{1,n}$.

(2) $S_k(f,x)$ интерполира функцията, т.е. $S_k(f,x_i) = y_i,\ \ i=\overline{0,n}$.
(3) $S_k(f,x)$ и производните му до ред (k-1) са непрекъснати в $[a,b]$.

В общия случай сплайнът от ред k не е единствен. За единственост се налагат допълнителни условия. Най-използвани са линеен, квадратичен и кубичен сплайн.

3.2 Линеен сплайн

Тук $k = 1$ и във всеки подинтервал $[x_{i-1}, x_i], i = \overline{1,n}$, сплайнът $S_1(f, x)$ е полином от първа степен (отсечка). Тъй като през две точки минава само една отсечка, линейният сплайн е единствен. Като се интерполира таблицата са всеки подинтервал, получаваме коефициентите на линейния сплайн по изходна таблица с данни. Графиката на сплайна е начупена линия.

Как се използва сплайнът за приближение в произволна точка z от $[a,b]$?

а) определяме в кой подинтервал $[x_{i-1}, x_i]$ се намира z.

б) заместваме $x = z$ в съответния ред f_i на $S_1(f, x)$.

Извод на формулите за линеен сплайн $S_1(f,x)$

Във всеки подинтервал $[x_{i-1}, x_i]$ търсим сплайна като интерполационен полином от 1-ва степен по x във вида:

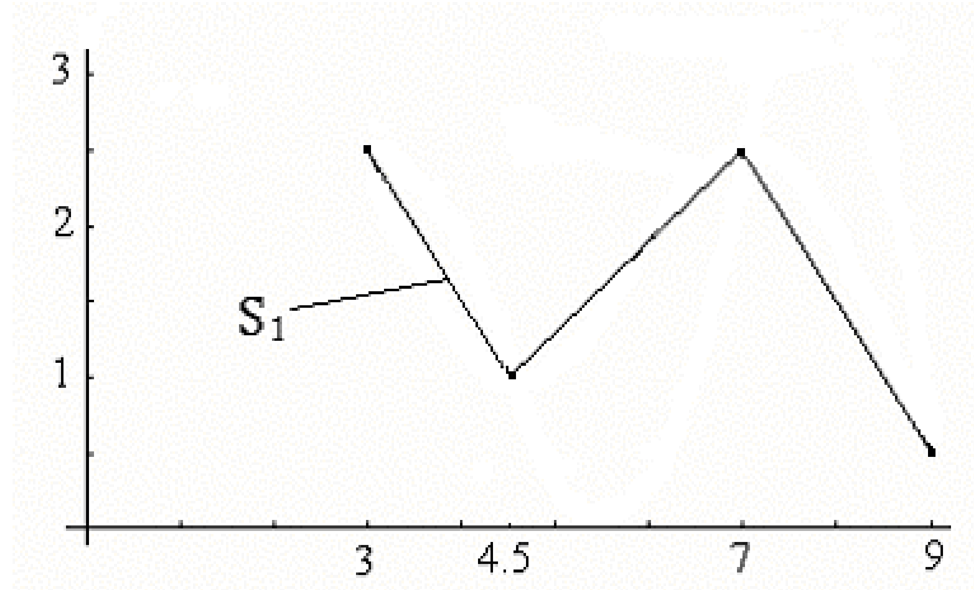
$$\boxed{f_i(x) = a_i + b_i(x - x_{i-1})}. \tag{11}$$

От условие 2) трябва $f_i(x_{i-1}) = y_{i-1}$ и $f_i(x_i) = y_i$. Т.е. $f_i(x_{i-1}) = a_i + b_i(x_{i-1} - x_{i-1}) = y_{i-1}$, откъдето $a_i = y_{i-1}$ и $f_i(x_i) = a_i + b_i(x_i - x_{i-1}) = y_i$ или $b_i = \dfrac{y_i - y_{i-1}}{h_i}$.

Обща формула на линеен сплайн	*Коефициенти на сплайна*
$S_1(f,x) = \begin{cases} f_1 = a_1 + b_1(x - x_0), & x \in [x_0, x_1] \\ \dots & \\ f_i = a_i + b_i(x - x_{i-1}), & x \in [x_{i-1}, x_i] \\ \dots & \\ f_n = a_n + b_n(x - x_{n-1}), & x \in [x_{n-1}, x_n] \end{cases}$	$a_i = y_{i-1}$, $b_i = \dfrac{y_i - y_{i-1}}{h_i}$, $i = \overline{1,n}$

<u>Пример.</u> Дадена е следната таблица на функцията $y = f(x)$:

x_i	3	4.5	7	9
y_i	2.5	1	2.5	0.5



Да се построи линеен сплайн и с негова помощ да се намерят приближените стойности на функцията в точката $z = 5$.

<u>Решение:</u>

Изчисляваме стъпките: $h_1 =$ 4.5 - 3 = 1.5; $h_2 =$ 7 – 4.5 = 2.5; $h_3 =$ 9 – 7 = 2.

По формулите пресмятаме последователно коефициентите a_i, b_i:

при $i = 1$, интервал [3, 4.5]: $a_1 = y_0 = 2.5$, $b_1 = \dfrac{y_1 - y_0}{h_1} = \dfrac{1 - 2.5}{1.5} = -1$;

при $i = 2$, интервал [4.5, 7]: $a_2 = y_1 = 1$, $b_2 = \dfrac{y_2 - y_1}{h_2} = \dfrac{2.5-1}{2.5} = 0.6$;

при $i = 3$, интервал [7, 9]: $a_3 = y_2 = 2{,}5$,

$$b_3 = \frac{y_3 - y_2}{h_3} = \frac{0.5-2.5}{2} = -1.$$

Нанасяме коефициентите и получаваме следната таблица:

i	a_i	b_i	Линеен сплайн	
1	2.5	-1.0	$f_1 = 2.5-(x-3)$	при $x \in [3,\ 4.5]$
2	1.0	0.6	$f_2 = 1+0.6(x-4.5)$	при $x \in [4.5,\ 7]$
3	2.5	-1.0	$f_3 = 2.5-(x-7)$	при $x \in [7, 9]$

За да изчислим приближената стойност на функцията в точката $z = 5$ виждаме, че тя се намира във втория подинтервал и ще се апроксимира по формулата за f_2. Тогава

$$f(5) \approx f_2(5) = a_2 + b_2(z - x_1) = 1 + 0.6(5-4.5) = 1.3.$$

3.3 Квадратичен сплайн

При $k = 2$ съгласно свойство 1) търсеният сплайн във всеки подинтервал $[x_{i-1}, x_i]$ е полином от втора степен (част от парабола) и коефициентите му са $3n$ на брой. Всъщност търсим $3n$ неизвестни a_i, b_i, c_i – коефициенти на сплайна.

Извод на формулите за квадратичен сплайн $S_2(f, x)$

От определението за сплайн имаме условията:

1) $\boxed{S_2(x) = f_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2}$ за $x \in [x_{i-1}, x_i]$, $i=1,2,\ldots,n$ (12)

2) $f_i(x_{i-1}) = y_{i-1}$, $f_i(x_i) = y_i$ $i=1,\ldots,n$ - интерполиране, ($2n$ условия)(13)

3) $S_2'(x)$ е непрекъсната във вътрешните точки $x_1, x_2, \ldots, x_{n-1}$

$f_i'(x_i) = f_{i+1}'(x_i)$ (n-1 условия). (14)

Така получихме общо $3n$-1 условия за определяне на $3n$ неизвестни. Едно условие остава свободно, т.е. квадратичният сплайн не е единствен и се определя със задаване на едно допълнително условие. Ако се зададе $b_1 = 0$ сплайнът се нарича **естествен.** Изчисляването на коефициентите му е рекурентно.

От условие (13а) получаваме със заместване на $x = x_{i-1}$ в (12)-

$f_i(x_{i-1}) = y_{i-1} = a_i$, или $\boxed{a_i = y_{i-1}}$ (15)

От (13б) $f_i(x_i) = a_i + b_i(x_i - x_{i-1}) + c_i(x_i - x_{i-1})^2 = a_i + b_i h_i + c_i h_i^2 = y_i$. Оттук

$$b_i + c_i h_i = \frac{y_i - y_{i-1}}{h_i}. \qquad (16)$$

Производните са: $f_i'(x) = b_i + 2c_i(x - x_{i-1})$ и $f_{i+1}'(x) = b_{i+1} + 2c_{i+1}(x - x_i)$. Оттук за $x = x_i$ от (14) намираме: $b_{i+1} = b_i + 2c_i h_i$. Изразяваме

$$\boxed{c_i = \frac{b_{i+1} - b_i}{2h_i}}. \qquad (17)$$

Последното можем да заместим в (16) и излючвайки $c_i h_i$,

$$\boxed{b_{i+1} = -b_i + 2\frac{y_i - y_{i-1}}{h_i}}. \tag{18}$$

Така стигаме до следната система уравнения (15),(17) и (18) за изчисляване коефициентите на сплайна S_2:

$$a_i = y_{i-1},$$

$$b_1 = \gamma_1,$$

$$b_{i+1} = -b_i + 2\frac{y_i - y_{i-1}}{h_i},$$

$$c_i = \frac{b_{i+1} - b_i}{2h_i}, \quad i = \overline{1,n},$$

където γ_1 е някакво начално значение на b_1. Ако $\gamma_1 = 0$ сплайнът е естествен.

Формулите са:

Обща формула на квадратичен сплайн	*Коефициенти на сплайна*
$S_2 = \begin{cases} f_1 = a_1 + b_1(x - x_0) + c_1(x - x_0)^2, & x \in [x_0, x_1] \\ \dots & \\ f_i = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2, & x \in [x_{i-1}, x_i] \\ \dots & \\ f_n = a_n + b_n(x - x_{n-1}) + c_n(x - x_{n-1})^2, & x \in [x_{n-1}, x_n] \end{cases}$	$a_i = y_{i-1},$ $b_1 = \gamma_1,$ $b_{i+1} = -b_i + 2\dfrac{y_i - y_{i-1}}{h_i},$ $c_i = \dfrac{b_{i+1} - b_i}{2h_i}, \quad i = \overline{1, n}$