

Записки от лекциите по  
„Математически методи в инженерните изследвания“  
за специалност „Транспортна техника“  
ОКС „Магистър“

## ЧИСЛЕНИ МЕТОДИ

от доц. д-р Румен Улучев

### §1. ГРЕШКИ

#### 1. Грешки при числените пресмятания.

Обикновено в практиката почти никога не работим с точните стойности на величините. Било поради неточности в измерването, било поради случайни грешки или други причини, вместо с точната стойност  $x$  ние си служим с някакво нейно приближение  $x^*$ . Много често това не е фатално за нашите изследвания. Все пак, ние трябва да държим сметка за грешката  $r = x - x^*$  която правим. Колкото по-сложни са изчисленията, в които използваме приближената стойност вместо точната, толкова по-голямо влияние може да окаже това върху крайния резултат.

Аналогичен ефект може да се получи при грешки от закръгляване при последователни компютърни изчисления, където грешката след всеки цикъл пресмятания се натрупва. Тъй като реалните числа, с които извършваме тези пресмятания, се „помнят“ от компютъра с краен брой цифри, е ясно, че грешките от такъв вид не могат да бъдат избегнати. При умножаване с големи числа натрупването на грешки от закръгляване може да доведе до неочаквани резултати, твърде далеч от истинските.

Нека  $x^*$  е едно приближение на числото  $x$  (което ще бележим с  $x^* \approx x$ ). Величината

$$A(x^*) = |x - x^*|$$

се нарича *абсолютна грешка* на това приближение.

**Пример 1.1.** Ако  $x^* \approx x$  и  $y^* \approx y$ , да оценим абсолютната грешка на сумата:

$$\begin{aligned} A(x^* + y^*) &= |(x + y) - (x^* + y^*)| = |(x - x^*) + (y - y^*)| \\ &\leq |x - x^*| + |y - y^*| = A(x^*) + A(y^*), \end{aligned}$$

т.е.  $A(x^* + y^*) \leq A(x^*) + A(y^*)$ . □

**Пример 1.2.** Нека функцията  $f(x, y)$  е диференцируема,  $x^* \approx x$  и  $y^* \approx y$ . Да оценим абсолютната грешка на стойността  $f(x^*, y^*)$ .

Използваме формулата на Тейлор, като пренебрегваме остатъчния член

$$f(x, y) - f(x^*, y^*) = (x - x^*) \frac{\partial f}{\partial x}(x^*, y^*) + (y - y^*) \frac{\partial f}{\partial y}(x^*, y^*)$$

и получаваме

$$\begin{aligned} A(f(x^*, y^*)) &= |f(x, y) - f(x^*, y^*)| \\ &\leq |x - x^*| \left| \frac{\partial f}{\partial x}(x^*, y^*) \right| + |y - y^*| \left| \frac{\partial f}{\partial y}(x^*, y^*) \right| \\ &= A(x^*) \left| \frac{\partial f}{\partial x}(x^*, y^*) \right| + A(y^*) \left| \frac{\partial f}{\partial y}(x^*, y^*) \right|. \end{aligned}$$

□

Абсолютната грешка не дава представа за значимостта на направената грешка. Грешка в радиуса от 0,1 мм не би била от голямо значение за колелото на трактор например, но със сигурност е съществена, ако става дума за зъбно колелце на механичен часовник. Ета защо, ще въведем една друга мярка за грешката. Числото

$$B(x^*) = \frac{|x - x^*|}{|x^*|}$$

се нарича *относителна грешка* на приближението  $x^*$ .

## 2. Разстояние между функции.

В разглежданията по-нататък често ще *апроксимираме* (приближаваме) една функция  $f$  с друга, обикновено по-проста,  $f^*$ . За да предпочетем едно приближение пред друго, трябва да можем да ги сравняваме. По-точно, ще въведем мярка, за *грешката*  $R = f - f^*$  между функцията

$f$  и нейното приближение. Тогава колкото по-малка е грешката, толкова по-добро ще бъде приближението. Взимайки предвид стойностите на грешката във всяка точка от разглеждания интервал, ще дефинираме понятието *разстояние между функции*, като обобщена мярка за грешката  $f - f^*$ .

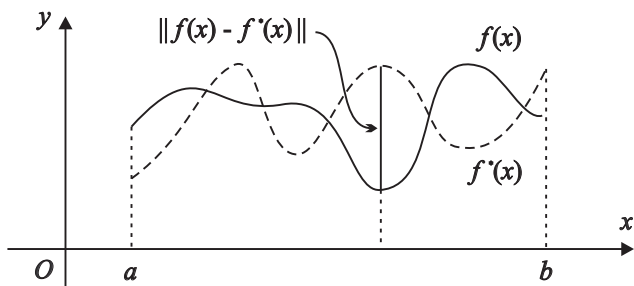
Множеството от всички непрекъснати в интервала  $[a, b]$  функции ще означаваме с  $C[a, b]$ . Това множество е линейно пространство относно операциите събиране на функции и умножаване на функция с число.

**Определение.** Нека  $f, f^* \in C[a, b]$ . Величината

$$\|f - f^*\| := \max_{x \in [a, b]} |f(x) - f^*(x)|$$

се нарича *равномерно разстояние между функциите  $f$  и  $f^*$*  в пространството  $C[a, b]$ .

Геометрическият смисъл на така дефинираното разстояние може да се види на фиг. 1.1.



Фигура 1.1

**Пример 1.3.** Функциите  $f(x) = \sin x$  и  $f^*(x) = 0$  са непрекъснати в  $[0, \pi]$ , т.е.  $f, f^* \in C[0, \pi]$ . За равномерното разстояние между тях имаме

$$\|f - f^*\| := \max_{x \in [0, \pi]} |\sin x - 0| = \max_{x \in [0, \pi]} |\sin x| = 1. \quad \square$$

**Пример 1.4.** Функциите  $g(x) = x^2$  и  $g^*(x) = x$  са непрекъснати в интервала  $[0, 1]$ , т.е.  $g, g^* \in C[0, 1]$ . Изследвайки за екстремуми грешката  $R(x) = x^2 - x$ , за равномерното разстояние между  $g$  и  $g^*$  намираме

$$\|g - g^*\| := \max_{x \in [0, 1]} |x^2 - x| = \left| \left(\frac{1}{2}\right)^2 - \frac{1}{2} \right| = \frac{1}{4}. \quad \square$$

Освен равномерното разстояние, се използват и т.нар. интегрални разстояния. Да означим с  $L_p[a, b]$  пространството от функции  $f$ , такива че  $|f|^p$  е интегрируема в интервала  $[a, b]$ .

**Определение.** Нека  $f, f^* \in L_p[a, b]$ . Величината

$$\|f - f^*\|_p := \left( \int_a^b |f(x) - f^*(x)|^p dx \right)^{1/p}$$

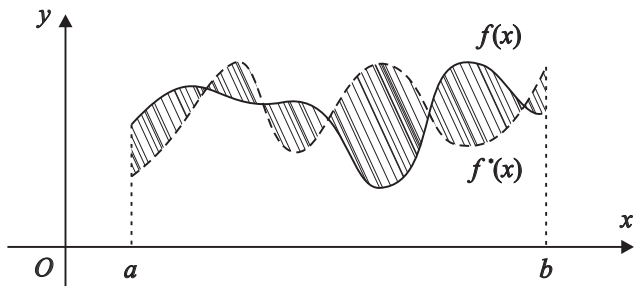
се нарича  $L_p$ -разстояние между функциите  $f$  и  $f^*$  в пространството  $L_p[a, b]$ .

Без да се спираме подробно на свойствата на тези пространства, ще споменем, че най-интересни са случаите  $p = 1$  и  $p = 2$ .

Например,  $L_1$ -разстоянието

$$\|f - f^*\|_1 := \int_a^b |f(x) - f^*(x)| dx$$

има ясен геометричен смисъл – лицето на фигурата, оградена от графиките на функциите  $f, f^*$  и правите с уравнения  $x - a = 0, x - b = 0$  (фиг. 1.2).



Фигура 1.2

В практическите задачи за обработка на данни голямо приложение има  $L_2$ -разстоянието

$$\|f - f^*\|_2 := \sqrt{\int_a^b |f(x) - f^*(x)|^2 dx},$$

известно още като *средноквадратично разстояние*.

**Пример 1.5.** Да пресметнем  $L_1$ - и  $L_2$ -разстоянията за интервала  $[0, 1]$  между функциите  $g(x) = x^2$  и  $g^*(x) = x$ .

Имаме

$$\|g - g^*\|_1 := \int_0^1 |x^2 - x| dx = \int_0^1 (-x^2 + x) dx = \frac{1}{6},$$

$$\|g - g^*\|_2 := \sqrt{\int_0^1 |x^2 - x|^2 dx} = \sqrt{\int_0^1 (x^4 - 2x^3 + x^2) dx} = \frac{1}{\sqrt{30}}.$$

□

### 3. Означенията $O(h^n)$ и $o(h^n)$ .

Нека са дадени точки  $x_0, x_1, \dots, x_n$  върху реалната права. Ще казваме, че това са *равноотдалечени възли*, ако  $x_k = x_0 + kh$ ,  $k = 0, 1, \dots, n$ , за някакво  $h > 0$ . Числото  $h$  ще наричаме *стъпка* на мрежата от възли. Всъщност, за равноотдалечените възли разстоянието между всеки два съседни възела е равно на стъпката.

Предвид това, че обикновено в изследванията се използват равноотдалечени възли, ще въведем две означения които ще ни дават представа за порядъка на грешката в различните методи, в зависимост от стъпката  $h$  на равноотдалечените възли.

**Определение.** Нека  $n \in \mathbb{N}$ . Под  $O(h^n)$  (четем: „о голямо от  $h^n$ “) ще разбираме коя да е функция  $\varphi(h)$ , такава че

$$\lim_{h \rightarrow 0} \frac{\varphi(h)}{h^n} = C, \quad C = \text{const.}, \quad C \neq 0.$$

Можем да смятаме, че за достатъчно малки  $h$  имаме  $O(h^n) \approx C h^n$ , за някаква константа  $C \neq 0$ .

**Пример 1.6.** Функцията  $f(h) = 3h^5$  е от порядъка на  $O(h^5)$ , защото

$$\lim_{h \rightarrow 0} \frac{3h^5}{h^5} = 3.$$

За функцията  $g(h) = 7h^2 \cos h$  можем да запишем, че  $g(h) = O(h^2)$ , тъй като

$$\lim_{h \rightarrow 0} \frac{7h^2 \cos h}{h^2} = \lim_{h \rightarrow 0} 7 \cos h = 7.$$

□

**Определение.** Нека  $n \in \mathbb{N}$ . Под  $o(h^n)$  (четем: „о малко от  $h^n$ “) ще разбираме коя да е функция  $\psi(h)$ , такава че

$$\lim_{h \rightarrow 0} \frac{\psi(h)}{h^n} = 0 .$$

Т.е.,  $o(h^n)$  клони към 0 по-бързо от  $h^n$  при  $t \rightarrow 0$ .

**Пример 1.7.** Функцията  $f(h) = 4,3h^{7/2}$  е от порядъка на  $o(h^3)$ , понеже

$$\lim_{h \rightarrow 0} \frac{4,3h^{7/2}}{h^3} = \lim_{h \rightarrow 0} 4,3\sqrt{h} = 0 .$$

За функцията  $g(h) = 5h^4 \operatorname{tg} h$  можем да запишем, че  $g(h) = o(h^4)$ , защото

$$\lim_{h \rightarrow 0} \frac{5h^4 \operatorname{tg} h}{h^4} = \lim_{h \rightarrow 0} 5 \operatorname{tg} h = 0 . \quad \square$$

## §2. ИНТЕРПОЛАЦИОННА ЗАДАЧА НА ЛАГРАНЖ ЗА АЛГЕБРИЧНИ ПОЛИНОМИ

### 1. Постановка на задачата.

Да предположим, че в нашите изследвания искаме да намерим явна зависимост на даден параметър  $f$  в зависимост от друг  $x$ . Стойностите на  $f$  можем да отчитаме по някакъв начин в точките  $x_i$ . Макар да не знаем точната аналитична зависимост  $f = f(x)$ , как да намерим сравнително проста функция  $\varphi(x)$ , която да има същите стойности като неизвестната за нас функция  $f$  в точките  $x_i$ ?

До аналогична задача можем да стигнем, ако вместо известната, но сложна и трудно поддаваща се на изследване функция  $f(x)$  искаме да работим с по-проста  $\varphi(x)$ , като условието е  $\varphi$  да приема същите стойности като  $f$  в определени точки.

Обикновено функцията  $\varphi(x)$  е алгебричен полином, рационална функция, тригонометричен полином, и т.н. В кой от функционалните класове ще търсим  $\varphi(x)$  зависи от физическата същност на нашите изследвания или от точността, която можем да постигнем със съответните методи. Количеството и поведението на отчетените данни също може да ни насочи към конкретен клас от функции.

След тези по-общи разсъждения да прецизираме задачата от математическа гледна точка.

Нека са дадени стойностите  $f_0, f_1, \dots, f_n$  на функцията  $f(x)$  в точките  $x_0 < x_1 < \dots < x_n$ , т.е. знаем  $f_i = f(x_i)$ ,  $i = 0, 1, \dots, n$ . Търсим функция  $\varphi$  от определен клас функции  $\Phi$ , такава че

$$\varphi(x_i) = f_i, \quad i = 0, 1, \dots, n. \quad (2.1)$$

Така формулираната задача се нарича *интерполационна задача на Лагранж*. Различните точки  $x_0, x_1, \dots, x_n$  се наричат *интерполационни възли*, условията (2.1) се наричат *интерполационни условия*, а  $\varphi(x)$  – *интерполираща функция* или накратко *интерполант*.

Ще разгледаме подробно случая, когато интерполантът е алгебричен полином. С  $\mathcal{P}_n$  ще бележим множеството от всички алгебрични полиноми с реални коефициенти от степен най-много  $n$ , т.е.

$$\mathcal{P}_n := \{p(x) : p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n\}.$$

Да отбележим, че един полином от  $\mathcal{P}_n$  е определен, ако са известни неговите  $n + 1$  коефициента, а за да определим еднозначно тези коефициенти са ни необходими  $n + 1$  условия.

*Интерполационната задача на Лагранж за алгебрични полиноми* се формулира както следва. При дадени интерполационни възли  $x_0 < x_1 < \dots < x_n$  и стойности  $f_0, f_1, \dots, f_n$  да се намери полином  $p \in \mathcal{P}_n$ , такъв че

$$p(x_i) = f_i, \quad i = 0, 1, \dots, n. \quad (2.2)$$

Геометричният смисъл на тази задача е следният: през точките с координати  $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$  в равнината да се прекара крива, явяваща се графика на полином от степен  $n$ .

Следващата теорема дава отговор на въпросите за съществуване и единственост на полиномиалния интерполант.

**Теорема 2.1.** *При всеки избор на възлите  $x_0 < x_1 < \dots < x_n$  и стойностите  $f_0, f_1, \dots, f_n$  съществува и то единствен полином  $p \in \mathcal{P}_n$ , удовлетворяващ интерполационните условия (2.2).*

## 2. Формула на Лагранж.

Тук ще дадем едно решение на задачата (2.2).

Нека са дадени точки  $x_0 < x_1 < \dots < x_n$  и стойности  $f_0, f_1, \dots, f_n$ . Полиномите от степен  $n$

$$\ell_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j}, \quad k = 0, 1, \dots, n, \quad (2.3)$$

се наричат *базисни полиноми на Лагранж* за дадените възли. За стойностите им в интерполационните възли имаме

$$\ell_k(x_i) = \begin{cases} 0, & i \neq k \\ 1, & i = k \end{cases}. \quad (2.4)$$

Да конструираме следния полином:

$$L_n(f; x) = \sum_{k=0}^n f_k \ell_k(x) = \sum_{k=0}^n f_k \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j}. \quad (2.5)$$

Като използваме (2.4) лесно съобразяваме, че полиномът  $L_n(f; x)$  удовлетворява интерполационните условия (2.2) и следователно е решение на задачата на Лагранж. Наистина,

$$\begin{aligned} L_n(f; x_j) &= f_0 \ell_0(x_j) + \cdots + f_{j-1} \ell_{j-1}(x_j) + f_j \ell_j(x_j) + \\ &\quad + f_{j+1} \ell_{j+1}(x_j) + \cdots + f_n \ell_n(x_j) = \\ &= 0 + \cdots + 0 + f_j \cdot 1 + 0 + \cdots + 0 = f_j. \end{aligned}$$

Формулата (2.5) се нарича *формула на Лагранж*. Поради факта, че тази формула се използва най-често когато  $n = 1, 2, 3$ , ще дадем явните формули в тези случаи.

При  $n = 1$  имаме два възела  $x_0 < x_1$  и две дадени стойности  $f_0, f_1$ ;

$$L_1(f; x) = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0}.$$

При  $n = 2$  имаме  $x_0 < x_1 < x_2$  и три дадени стойности  $f_0, f_1, f_2$ ;

$$\begin{aligned} L_2(f; x) &= f_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + \\ &\quad + f_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \end{aligned}$$

В случая, когато  $n = 3$  имаме  $x_0 < x_1 < x_2 < x_3$  и четири дадени стойности  $f_0, f_1, f_2, f_3$ . Тогава

$$\begin{aligned} L_3(f; x) &= f_0 \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} + f_1 \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} + \\ &\quad + f_2 \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} + f_3 \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}. \end{aligned}$$



**Пример 2.1.** Да се намери интерполационният полином на Лагранж, който във възлите  $x_0 = -1$ ,  $x_1 = 1$ ,  $x_2 = 2$  приема стойности, съответно  $f_0 = 2$ ,  $f_1 = 4$ ,  $f_2 = 8$ .

Заместваме във формулата за случая  $n = 2$ :

$$\begin{aligned} L_2(f; x) &= 2 \frac{(x-1)(x-2)}{(-1-1)(-1-2)} + 4 \frac{(x+1)(x-2)}{(1+1)(1-2)} + 8 \frac{(x+1)(x-1)}{(2+1)(2-1)} = \\ &= 2 \frac{x^2 - 3x + 2}{6} + 4 \frac{x^2 - x - 2}{-2} + 8 \frac{x^2 - 1}{3} = x^2 + x + 2. \end{aligned}$$

Непосредствената проверка показва, че за така намерения полином  $L_2(f; x) = x^2 + x + 2$  имаме  $L_2(f; -1) = 2 = f_0$ ,  $L_2(f; 1) = 4 = f_1$  и  $L_2(f; 2) = 8 = f_2$ .  $\square$

### 3. Оценка на грешката.

Нека  $L_n(f; x)$  е интерполационният полином от степен  $n$ , приемащ стойности  $f(x_0), f(x_1), \dots, f(x_n)$  във възлите  $x_0 < x_1 < \dots < x_n$ , съответно. Да означим с

$$R_n(f; x) = f(x) - L_n(f; x)$$

грешката на приближението  $f(x) \approx L_n(f; x)$ .

**Теорема 2.2.** Ако  $f \in C^{n+1}[a, b]$  и  $a \leq x_0 < \dots < x_n \leq b$ , съществува точка  $\xi \in (a, b)$ , такава че

$$R_n(f; x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\cdots(x-x_n). \quad (2.6)$$

**Пример 2.2.** С полином от каква степен  $n$  трябва да интерполираме функцията  $f(x) = \sin x$ ,  $x \in [0, 2]$ , че за грешката от интерполацията да е изпълнено  $|R_n(f; x)| \leq 0,1$ ?

Нас, разбира се, ни интересува полиномът да бъде от възможно най-ниска степен. Тогава и изчисленията за неговото намиране ще са по-малко.

При интерполиране в произволни различни  $n+1$  възела от интервала  $[0, 2]$  имаме

$$|R_n(f; x)| = \frac{|f^{(n+1)}(\xi)|}{(n+1)!} |x-x_0| \cdot |x-x_1| \cdots |x-x_n|.$$

Тъй като  $x, x_0 \in [0, 2]$ , то  $|x-x_0| \leq 2$ . Аналогично  $|x-x_k| \leq 2$  за всяко  $k = 0, 1, \dots, n$ .

Да забележим, че  $|f^{(n+1)}(x)| = |\sin x|$  или  $|f^{(n+1)}(x)| = |\cos x|$  (в зависимост от четността на  $n$ ), но винаги ще е изпълнено

$$|f^{(n+1)}(x)| \leq 1, \quad x \in [0, 2].$$

Тогава

$$|R_n(f; x)| \leq \frac{1}{(n+1)!} 2^{n+1}, \quad x \in [0, 2].$$

Сега да намерим най-малкото  $n$ , за което

$$\frac{2^{n+1}}{(n+1)!} \leq 0,1.$$

От последните две неравенства се вижда, че то ще ни осигури необходимата точност, а именно  $|R_n(f; x)| \leq 0,1$ .

Непосредствената проверка показва, че

$$\frac{2^{n+1}}{(n+1)!} > 0,1, \quad n = 1, 2, 3, 4, 5, \quad \frac{2^7}{7!} = \frac{128}{840} > 0,1, \quad \frac{2^8}{8!} = \frac{256}{6720} \leq 0,1.$$

Следователно, достатъчно е да интерполираме с полином от  $\mathcal{P}_7$  с  $n+1 = 8$  възела, без значение как сме избрали възлите в интервала  $[0, 2]$ .  $\square$

Ще споменем само, че е възможно интерполационните възли да се подберат *оптимално*, т.е. така, че изразът

$$\max_{a \leq x_0 < \dots < x_n \leq b} |(x-x_0)(x-x_1)\cdots(x-x_n)|,$$

участващ във формулата за грешката (2.6), да е възможно най-малък. Оптималните интерполационни възли са известни като *Чебишеви възли*. За интервала  $[-1, 1]$  те са

$$x_k = \cos \frac{(2k+1)\pi}{2(n+1)}, \quad k = 0, 1, \dots, n,$$

а за произволен интервал  $[a, b]$  са

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{(2k+1)\pi}{2(n+1)}, \quad k = 0, 1, \dots, n.$$

За сравнение с Пример 2.2, интерполирайки  $\sin x$  в Чебишевите възли с полином само от четвърта степен, получаваме грешка по-малка от 0,1.

#### 4. Формула на Нютон с крайни разлики.

Ще дадем едно друго представяне на интерполационния полином на Лагранж при равноотдалечени възли, но преди това ще въведем едно ново понятие.

Нека  $f$  е дадена функция, а  $x_k = x_0 + kh$ ,  $k = 0, 1, \dots$ , са равноотдалечени възли със стъпка  $h > 0$ . Величините

$$\Delta f_i = f_{i+1} - f_i, \quad i = 0, 1, \dots,$$

се наричат *крайни разлики от първи ред* на функцията  $f$ . Можем да дефинираме рекурентно *крайни разлики от втори ред*:

$$\Delta^2 f_i = \Delta f_{i+1} - \Delta f_i, \quad i = 0, 1, \dots,$$

и въобще, *крайни разлики от  $k$ -ти ред*:

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i, \quad i = 0, 1, \dots$$

За пълнота полагаме

$$\Delta^0 f_i = f_i = f(x_i), \quad i = 0, 1, \dots$$

Крайните разлики можем да поместим в таблица:

$f_0$					
	$\Delta f_0$				
$f_1$		$\Delta^2 f_0$			
	$\Delta f_1$		$\dots$		
$f_2$		$\Delta^2 f_1$	$\dots$		
$\dots$	$\dots$	$\dots$	$\dots$	$\Delta^{n-1} f_0$	
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\Delta^n f_0$
$\dots$	$\dots$	$\dots$	$\dots$	$\Delta^{n-1} f_0$	
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	
$f_{n-2}$		$\Delta^2 f_{n-3}$	$\dots$		
	$\Delta f_{n-2}$		$\dots$		
$f_{n-1}$		$\Delta^2 f_{n-2}$	$\dots$		
	$\Delta f_{n-1}$		$\dots$		
$f_n$					

Нека  $t = \frac{x - x_0}{h}$  и както обикновено,  $\binom{t}{k} = \frac{t(t-1)\dots(t-k+1)}{k!}$ , дори за  $t \in \mathbb{R}$ . Тогава

$$L_n(f; x) = \sum_{k=0}^n \binom{t}{k} \Delta^k f_0 \tag{2.7}$$

(формула на Нютон с крайни разлики). Да забележим, че в (2.7) участват крайните разлики от горния диагонал на горната таблица. Освен това, изразът в дясната страна не зависи явно от променливата  $x$ ; зависимостта е неявна, чрез  $t$ .

Интерполационната формула на Нютон с крайни разлики при  $n = 1$  (два възела), при  $n = 2$  (три възела) и при  $n = 3$  (четири възела) има вида, съответно:

$$\begin{aligned} L_1(f; x) &= f_0 + t \Delta f_0, \\ L_2(f; x) &= f_0 + t \Delta f_0 + \frac{t(t-1)}{2} \Delta^2 f_0, \\ L_3(f; x) &= f_0 + t \Delta f_0 + \frac{t(t-1)}{2} \Delta^2 f_0 + \frac{t(t-1)(t-2)}{6} \Delta^3 f_0, \end{aligned}$$

като отново напомняме, че навсякъде  $t = \frac{x - x_0}{h}$ .

Ще отбележим, че връзката

$$L_{k+1}(f; x) = L_k(f; x) + \binom{t}{k+1} \Delta^{k+1} f_0$$

ще ни спести много изчисления, ако вече сме намерили интерполационния полином от степен  $k$  при направени  $k+1$  измервания за  $f$ , и след направеното поредно  $(k+2)$ -во наблюдение трябва да намерим интерполационния полином от степен  $k+1$ .

Изведената тук формула на Нютон с крайни разлики (2.7) стои в основата на други числени методи – за числено интегриране, числено решаване на диференциални уравнения и др.

## 5. Формула на Нютон с разделени разлики.

Следващата формула за решаването на интерполационната задача на Лагранж има няколко предимства пред изложените дотук. Едно от тях е, че от нея можем лесно да получим формула, с която да интерполираме не само стойности, но и производни на дадената функция.

Нека  $f$  е дадена функция, а  $x_0 < x_1 < \dots < x_n$  са възлите на интерполиране. Величините

$$f[x_i, x_{i+1}] = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}, \quad i = 0, 1, \dots, n-1$$

се наричат *разделени разлики от първи ред* на функцията  $f$ . *Разделени*

разлики от втори ред дефинираме рекурентно:

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}, \quad i = 0, 1, \dots, n-2,$$

и въобще, *разделени разлики от k-ти ред*:

$$f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

за  $i = 0, 1, \dots, n-k$ . Накрая, за пълнота полагаме

$$f[x_i] = f_i = f(x_i), \quad i = 0, 1, \dots, n.$$

Тогава интерполационният полином се записва във вида

$$\begin{aligned} L_n(f; x) &= \sum_{k=0}^n (x - x_0) \cdots (x - x_{k-1}) f[x_0, \dots, x_k] = \\ &= f_0 + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + \\ &\quad + \cdots + (x - x_0) \cdots (x - x_{n-1}) f[x_0, \dots, x_n] \end{aligned}$$

(формула на Нютон с разделени разлики). Да забележим, че в тази формула участват разделените разлики от горния диагонал на следната таблица:

$x_0$	$f_0$			
		$f[x_0, x_1]$		
$x_1$	$f_1$		$f[x_0, x_1, x_2]$	
		$f[x_0, x_1]$		$\dots$
$x_2$	$f_2$		$f[x_1, x_2, x_3]$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$x_{n-1}$	$f_{n-2}$		$f[x_{n-3}, x_{n-2}, x_{n-1}]$	$\dots$
		$f[x_{n-2}, x_{n-1}]$		$\dots$
$x_{n-1}$	$f_{n-1}$		$f[x_{n-2}, x_{n-1}, x_n]$	
		$f[x_{n-1}, x_n]$		
$x_n$	$f_n$			

Ето интерполационната формула на Нютон с разделени разлики при  $n = 1$  (два възела), при  $n = 2$  (три възела) и при  $n = 3$  (четири възела),

СЪОТВЕТНО:

$$\begin{aligned}L_1(f; x) &= f_0 + (x - x_0)f[x_0, x_1], \\L_2(f; x) &= f_0 + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2], \\L_3(f; x) &= f_0 + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \\&\quad + (x - x_0)(x - x_1)(x - x_2)f[x_0, x_1, x_2, x_3].\end{aligned}$$

Подобно на формулата с крайни разлики и тук имаме връзка от вида:

$$L_{k+1}(f; x) = L_k(f; x) + (x - x_0) \cdots (x - x_k)f[x_0, \dots, x_{k+1}],$$

която ще ни спести много изчисления, ако вече сме намерили интерполационния полином от степен  $k$  при направени  $k + 1$  измервания за  $f$ , и след направеното поредно  $(k + 2)$ -во наблюдение трябва да намерим интерполационния полином от степен  $k + 1$ .

**Пример 2.3.** Да се намери полином от втора степен, интерполиращ функцията  $f(x) = \frac{2x + 1}{x - 1}$  във възлите  $x_0 = -2$ ,  $x_1 = 0$  и  $x_2 = 4$ .

Последователно пресмятаме разделените разлики:

$$\begin{aligned}x_0 = -2 \quad f(x_0) &= 1 && f[x_0, x_1] = -1 \\x_1 = 0 \quad f(x_1) &= -1 && f[x_0, x_1, x_2] = \frac{1}{3}. \\&&& f[x_1, x_2] = 1 \\x_2 = 4 \quad f(x_2) &= 3\end{aligned}$$

Тогава

$$\begin{aligned}L_2(f; x) &= f_0 + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] = \\&= 1 + (x + 2)(-1) + (x + 2)(x - 0)\frac{1}{3} = 1 - x - 2 + \frac{1}{3}x^2 + \frac{2}{3}x = \\&= \frac{1}{3}x^2 - \frac{1}{3}x - 1,\end{aligned}$$

$$\text{т.е. } L_2(f; x) = \frac{1}{3}x^2 - \frac{1}{3}x - 1. \quad \square$$

### §3. СПЛАЙН-ФУНКЦИИ

#### 1. Определение.

Полиномите, безспорно, са най-простият математически апарат с който можем да интерполираме и апроксимираме функции и данни. Ако трябва обаче да интерполираме по-голямо количество данни (например 10-15 стойности), полученият полином също ще се окаже от висока степен и неминуемо ще срещнем проблеми при работата с него. Дори използването на калкулатори и компютри може да не ни помогне много в случай, че се наложи да пресметнем стойност на полином от петнадесета степен при  $x = 10,3$ . От друга страна, точността при приближаване с полиноми съществено зависи от интервала, в който правим нашите изследвания.

Идеята да се използват прости функции, каквито са полиномите от ниски степени, приближението да бъде добро в голям интервал и в същото време формулите и алгоритмите да са удобни за практически нужди, е довела до създаването на нов апарат за приближаване на функции, а именно *начастни полиномиалните функции* (или още *сплайн-функции*).

**Определение.** Функцията  $s(x)$  се нарича сплайн от степен  $r$  с възли  $x_1 < x_2 < \dots < x_k$ , ако:

- 1)  $s(x) = p_j(x)$ ,  $x \in (x_j, x_{j+1})$ ,  $p_j \in \mathcal{P}_r$ , за всяко  $j = 0, 1, \dots, k$   
(тук сме положили  $x_0 = -\infty$  и  $x_{k+1} = +\infty$ );
- 2)  $s \in C^{r-1}(-\infty, +\infty)$ .

С други думи, сплайнът е функция, съвпадаща с някакъв полином от степен  $r$  във всеки от интервалите между възлите и имаща непрекъснати производни до  $(r - 1)$ -ви ред, включително. Така графиката на сплайна е съставена от парчета от полиномиални криви, залепени гладко във възлите му. Това изяснява другото наименование – *начастни полиномиална функция*.

При  $r = 1$  получаваме сплайн-функция от първа степен (*линеен сплайн*). Графиката на един линеен сплайн е добре известна на читателя под името *начупена линия*.

*Квадратичен сплайн* (сплайн-функция от втора степен) имаме при  $r = 2$ . Графиката на един квадратичен сплайн се състои от парчета квадратични параболи.

Едни от най-ефективните методи за интерполиране и приближаване се базират на използване на *кубични сплайни*, т.е. сплайн-функции от

степен  $r = 3$ .

## 2. Линејни сплайни.

Тук ще разгледаме задачата за интерполиране на дадена функција  $f(x)$  в точките  $x_1 < x_2 < \dots < x_k$  с линеен сплайн  $s(x)$  имащ за възли същите тези точки. Търсениот сплайн от прва степен трябва да удовлетворява условията

$$s(x_j) = f(x_j), \quad j = 1, 2, \dots, k.$$

Във всеки един от интервалите  $[x_j, x_{j+1}]$  сплайнът  $s(x)$  съвпада с полинома от прва степен, интерполирац  $f(x)$  в точките  $x_j$  и  $x_{j+1}$ , т.е. с

$$L_1(f; x) = f(x_j) + (x - x_j)f[x_j, x_{j+1}] = f(x_j) + (x - x_j) \frac{f(x_{j+1}) - f(x_j)}{x_{j+1} - x_j}.$$

Следователно

$$s(x) = f(x_j) + (x - x_j) \frac{f(x_{j+1}) - f(x_j)}{x_{j+1} - x_j}, \quad x \in [x_j, x_{j+1}], \quad j = 1, \dots, k-1.$$

Да оценим грешката на тази интерполация при равноотдалечени възли. Нека  $x_j = x_1 + (j-1)h$ ,  $j = 1, 2, \dots, k$ , и  $h > 0$  е стъпката между възлите. Да предположим, че  $f \in C^2[x_1, x_k]$ . Тогава втората производна на  $f$  е ограничена в  $[x_1, x_k]$ , т.е. съществува константа  $M$ , такава че  $|f''(x)| \leq M$  за всяко  $x \in [x_1, x_k]$ .

Да вземем произволно фиксирано  $x \in [x_1, x_k]$ . То със сигурност попада в някое от малките интервалчета с дължина  $h$ , например  $[x_j, x_{j+1}]$ . Грешката  $R(x) = f(x) - s(x)$  от интерполацията на  $f$  с линейния сплайн  $s$  в точката  $x$  е точно грешката

$$R_1(f; x) = f(x) - L_1(f; x) = \frac{f''(\xi)}{2} (x - x_j)(x - x_{j+1}),$$

където  $\xi$  е точка от интервала  $(x_j, x_{j+1})$  съгласно Теорема 2.2. Следователно

$$|R(x)| = |R_1(f; x)| = \frac{|f''(\xi)|}{2} (x - x_j)(x_j + h - x).$$

Тъй като

$$\max_{x \in [x_j, x_j+h]} (x - x_j)(x_j + h - x) = \frac{h^2}{4},$$



то

$$|R(x)| \leq \frac{|f''(\xi)|}{2} \cdot \frac{h^2}{4} \leq \frac{Mh^2}{8}, \quad (3.1)$$

откъдето

$$|f(x) - s(x)| = O(h^2).$$

**Пример 3.1.** Интерполираме функцията  $f(x) = \sin x$  в равноотдалечени възли  $0 = x_1 < x_2 < \dots < x_k = 2$  с линеен сплайн. Колко възела ще ни осигурят грешката от интерполацията да бъде по-малка от 0,1?

В случая имаме  $h = \frac{2}{k-1}$  и  $M = 1$ , защото  $|f''(x)| = |-\sin x| \leq 1$  за всяко  $x$ . Търсим най-малкото  $k$ , за което

$$\frac{4}{8(k-1)^2} \leq 0,1, \quad \text{т.е. } 5 < (k-1)^2.$$

Оттук намираме  $k = 4$ . Тогава съгласно (3.1) имаме

$$|R(x)| \leq \frac{Mh^2}{8} \leq \frac{1 \cdot 2^2}{3^2 \cdot 8} = \frac{1}{18} < 0,1,$$

като  $h = \frac{2}{3}$  и възлите са  $x_0 = 0$ ,  $x_1 = \frac{2}{3}$ ,  $x_2 = \frac{4}{3}$  и  $x_4 = 2$ . □

### 3. Кубични сплайни.

Нека  $f(x)$  е дадена функция и  $x_1 < x_2 < \dots < x_k$ . Ще намерим кубичен сплайн  $s(x) \in C^1[x_1, x_n]$ , такъв че

$$s(x_j) = f(x_j) \quad \text{и} \quad s'(x_j) = f'(x_j), \quad j = 1, 2, \dots, k. \quad (3.2)$$

Интерполационна задача от такъв вид се нарича *задача на Ермит*. Особеното в нея е, че освен стойностите на функцията  $f(x)$  интерполираме и стойностите на производната  $f'(x)$ .

Ще решим задачата (3.2) локално, както задачата за интерполиране с линейни сплайни. Ако  $s(x)$  съвпада с полинома от трета степен  $p_j(x)$  в интервала  $[x_j, x_{j+1}]$ , достатъчно е да намерим полинома  $p_j(x)$  за който

$$\begin{aligned} p_j(x_j) &= f(x_j), & p'_j(x_j) &= f'(x_j), \\ p_j(x_{j+1}) &= f(x_{j+1}), & p'_j(x_{j+1}) &= f'(x_{j+1}). \end{aligned}$$

За целта ще използваме формулата на Нютон с разделени разлики.

В § 2 дефинирахме разделени разлики при различни възли. Например

$$f[t_i, t_{i+1}] = \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i}, \quad t_i \neq t_{i+1}.$$

Да забележим, че за диференцируема функция  $f$  имаме

$$\lim_{t_{i+1} \rightarrow t_i} f[t_i, t_{i+1}] = \lim_{t_{i+1} \rightarrow t_i} \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} = f'(t_i),$$

т.е. можем да дефинираме разделена разлика от първи ред в два съвпадащи възела по следния начин:

$$f[x_i, x_i] = f'(x_i).$$

Тогава таблицата с разделените разлики за намирането на  $p_j(x)$  ще изглежда по следния начин:

$$\begin{array}{ccccccc} x_j & f(x_j) & & & & & \\ & & f[x_j, x_j] = f'(x_j) & & & & \\ x_j & f(x_j) & & f[x_j, x_j, x_{j+1}] & & & \\ & & f[x_j, x_{j+1}] & & f[x_j, x_j, x_{j+1}, x_{j+1}] & & \\ x_{j+1} & f(x_{j+1}) & & f[x_j, x_{j+1}, x_{j+1}] & & & \\ & & f[x_{j+1}, x_{j+1}] = f'(x_{j+1}) & & & & \\ x_{j+1} & f(x_{j+1}) & & & & & \end{array}$$

Разделените разлики, при които крайните възли са различни, пресмятаме рекурентно по познатия начин. За  $p_j(x)$  получаваме

$$\begin{aligned} p_j(x) = & f(x_j) + (x - x_j)f'(x_j) + (x - x_j)^2 f[x_j, x_j, x_{j+1}] + \\ & + (x - x_j)^2 f[x_j, x_j, x_{j+1}, x_{j+1}], \end{aligned}$$

а интерполационния Ермитов кубичен сплайн определяме с

$$s(x) = p_j(x), \quad x \in [x_j, x_{j+1}], \quad j = 1, 2, \dots, k-1.$$

**Забележка.** Така конструираният сплайн не е с максимално възможната гладкост, т.е. не е от класа  $C^2$ . За сметка на това намирането му е сравнително лесно.

#### 4. *B*-сплайни.

В много от алгоритмите, използващи сплайн-функции се прилага представяне по базис от функции, т.нар. *B*-сплайни.

Функцията

$$(x-t)_+^{n-1} = \begin{cases} 0, & x < t, \\ (x-t)^{n-1}, & x \geq t. \end{cases}$$

се нарича *отсечена степенна функция*. При фиксирано  $t$ , тази функция на променливата  $x$  съвпада с константата 0 наляво от  $t$  и с отместената степенна функция  $(x-t)^{n-1}$  надясно от  $t$  по абсцисната ос.

Разделената разлика на отсечената степенна функция  $(x-t)_+^{n-1}$  относно променливата  $x$  в точките  $x_0 < x_1 < \dots < x_n$  наричаме *B-сплайн от степен  $n-1$  с възли  $x_0, x_1, \dots, x_n$*  и ще означаваме с  $B(x_0, x_1, \dots, x_n; t)$ . Така дефинираният *B-сплайн* е функция на  $t$ , но освен това зависи и от избора на възлите, като съгласно определението

$$B(x_0, x_1, \dots, x_n; t) = (x-t)_+^{r-1} [x_0, x_1, \dots, x_n].$$

Важно свойство на *B-сплайните* е следното:

$$\begin{aligned} B(x_0, x_1, \dots, x_n; t) &> 0, & t \in (x_0, x_n), \\ B(x_0, x_1, \dots, x_n; t) &= 0, & t \notin (x_0, x_n). \end{aligned}$$

Интервалът  $(x_0, x_n)$  се нарича *носител* на *B-сплайна*  $B(x_0, x_1, \dots, x_n; t)$ . Извън носителя си *B-сплайнът* е константа 0, а върху носителя приема положителни стойности.

Нека е дадена редицата  $x_1 < x_2 < \dots < x_n < x_{n+1} < \dots < x_{N+n}$ . За всеки  $n+1$  последователни точки от тази редица дефинираме *B-сплайн от степен  $n-1$* :

$$B_{i,n-1}(t) = B(x_i, x_{i+1}, \dots, x_{i+n}; t), \quad i = 1, 2, \dots, N.$$

По този начин получаваме редица от *B-сплайни*, определена от редицата възли. Следните *рекурентни формули на Кокс-де Бор* между *B-сплайните* от последователни степени дават възможност за бързо и ефективно пресмятане на стойностите на *B-сплайните*:

$$\begin{aligned} B_{i,0}(t) &= \begin{cases} \frac{1}{x_{i+1} - x_i}, & t \in [x_i, x_{i+1}), \\ 0, & t \notin [x_i, x_{i+1}), \end{cases} \\ B_{i,k-1}(t) &= \frac{t - x_i}{x_{i+k} - x_i} B_{i,k-2}(t) + \frac{x_{i+k} - t}{x_{i+k} - x_i} B_{i+1,k-2}(t), \end{aligned}$$

където  $i = 1, 2, \dots, N = n - k$  и  $k = 2, 3, \dots, n$ .

В сила е следната теорема.

**Теорема 3.1.** *Всяка сплайн-функция  $s(t)$  от степен  $n - 1$  с възли  $x_{n+1} < x_{n+2} < \dots < x_N$  може да се представи по единствен начин като линейна комбинация на  $B$ -сплайни във вида:*

$$s(t) = \sum_{i=1}^N \alpha_i B_{i,n-1}(t) . \quad (3.3)$$

Представянето (3.3) има някои предимства при компютърната реализация на алгоритми, използващи сплайн-функции. Едно от тях е, че всяка стойност на сплайн-функция от степен  $n - 1$  в точка  $t$  се явява линейна комбинация на стойности в точката  $t$  на само  $n$  на брой  $B$ -сплайна – тези от редицата, в чиито носител попада  $t$ . От друга страна, с рекурентните формули на Кокс-де Бор можем да пресметнем бързо стойностите на  $B$ -сплайните във всяка фиксирана точка  $t$ .

Да разгледаме задачата за намиране на интерполяционен сплайн  $s(t)$  от степен  $n - 1$  с възли  $x_1 < x_2 < \dots < x_n < x_{n+1} < \dots < x_{N+n}$ , удовлетворяващ условията

$$s(t_j) = y_j , \quad j = 1, 2, \dots, N , \quad (3.4)$$

където  $t_1 < t_2 < \dots < t_N$  и  $y_1, y_2, \dots, y_N$  са произволни стойности.

**Теорема 3.2.** *Интерполяционната задача (3.4) има единствено решение при всеки избор на стойностите  $y_1, y_2, \dots, y_N$  тогава и само тогава, когато интерполяционните възли  $t_1 < t_2 < \dots < t_N$  удовлетворяват условията*

$$B_i(t_i) > 0 , \quad i = 1, 2, \dots, N ,$$

т.е.  $i$ -тия интерполяционен възел  $t_i$  принадлежи на носителя на  $i$ -тия  $B$ -сплайн  $B_i$ .

Резултатите от Теорема 3.1 и Теорема 3.2 са основание да търсим решение на интерполяционната задача (3.4) във вид на линейна комбинация на  $B$ -сплайни (3.3). Тогава условията (3.4) ще ни дадат система линейни уравнения за коефициентите  $\alpha_1, \alpha_2, \dots, \alpha_N$  в представянето (3.3). Теорема 3.2 е всъщност необходимо и достатъчно условие получената линейна система да има и то единствено решение.

## 5. Параметрични сплайни.

Описването на криви с произволна форма в CAD/CAE и дизайнерските програми се базира на така наречените *криви на Безиè* – криви, чиито координатни функции са полиноми от определен тип.

Нека са дадени точки  $\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_n$  в равнината. Кривата  $\gamma$ , зададена с векторната функция

$$\mathbf{p}(t) = \sum_{k=0}^n \mathbf{B}_k \binom{n}{k} t^k (1-t)^{n-k}, \quad t \in [0, 1],$$

се нарича *крива на Безиè* от степен  $n$ . Нека  $(x_k, y_k)$  са координатите на контролните точки  $\mathbf{B}_k$ ,  $k = 0, 1, \dots, n$ , а  $x(t)$  и  $y(t)$  са координатните функции на  $\mathbf{p}(t)$ , т.е.

$$\mathbf{p}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}.$$

Тогава имаме

$$\gamma : \begin{cases} x(t) = \sum_{k=0}^n x_k \binom{n}{k} t^k (1-t)^{n-k} \\ y(t) = \sum_{k=0}^n y_k \binom{n}{k} t^k (1-t)^{n-k} \end{cases}, \quad t \in [0, 1]. \quad (3.5)$$

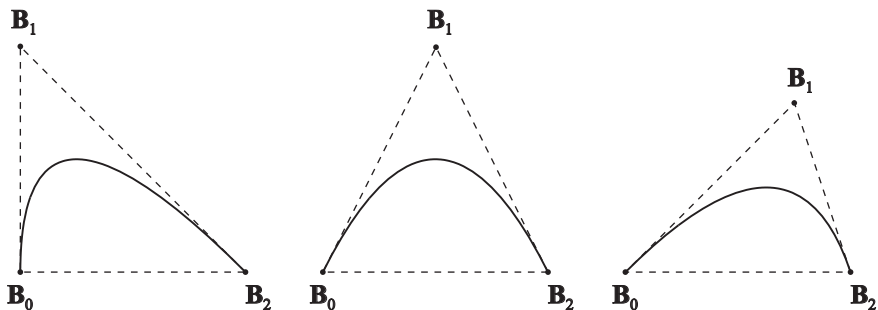
Очевидно, координатните функции  $x(t)$  и  $y(t)$  са алгебрични полиноми от степен  $n$ , записани по специален начин.

Една гладка крива наричаме *параметричен сплайн от степен  $n$* , ако е съставена от няколко криви на Безиè от степен  $n$ . Така например, квадратичният параметричен сплайн е съставен от криви на Безиè от втора степен, кубичният параметричен сплайн е съставен от криви на Безиè от трета степен, и т.н. Това са и случаите с най-голямо приложение в практиката.

В случая на крива на Безиè от втора степен, контролните точки са три и кривата  $\gamma$  се задава с векторната функция

$$\mathbf{p}(t) = \mathbf{B}_0(1-t)^2 + \mathbf{B}_1 2t(1-t) + \mathbf{B}_2 t^2, \quad t \in [0, 1].$$

Да забележим, че  $\mathbf{p}(0) = \mathbf{B}_0$  и  $\mathbf{p}(1) = \mathbf{B}_2$ . Може да се докаже още, че квадратичната крива на Безиè  $\gamma$  изцяло се съдържа в триъгълника  $\mathbf{B}_0\mathbf{B}_1\mathbf{B}_2$ . На фиг. 3.1 е показан видът на кривата  $\gamma$  при фиксирани крайни точки  $\mathbf{B}_0$  и  $\mathbf{B}_2$  и при различни положения на контролната точка  $\mathbf{B}_1$ .



Фигура 3.1

Наименованието *контролни точки* произлиза от факта, че лесно и удобно можем да променяме формата на кривата измествайки положението на тези точки.

Поради сравнително простия вид и гъвкавостта на формата, в дизайнерските компютърни програми се използват кубични криви на Безиє:

$$\mathbf{p}(t) = \mathbf{B}_0(1-t)^3 + \mathbf{B}_13t^2(1-t) + \mathbf{B}_23t(1-t)^2 + \mathbf{B}_3t^3, \quad t \in [0, 1].$$

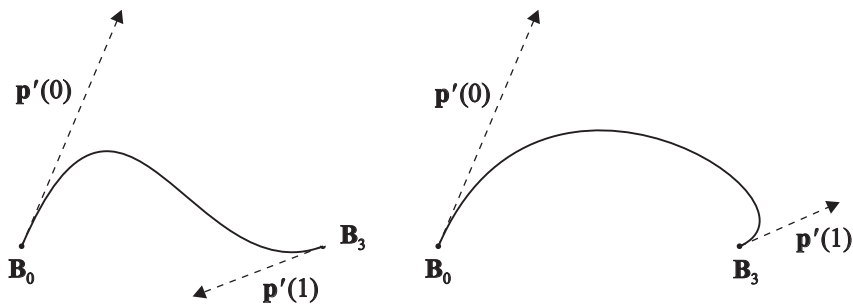
Една кубична крива на Безиє може да бъде определена, ако знаем началната позиция  $\mathbf{p}(0) = \mathbf{B}_0$ , крайната позиция  $\mathbf{p}(1) = \mathbf{B}_3$ , както и допирателните вектори  $\mathbf{p}'(0)$  и  $\mathbf{p}'(1)$  (или което е същото, големините и посоките на допирателните вектори) в тези точки. Ще отбележим, че тази задача е аналог на Ермитовата интерполационна задача за кубични алгебрични полиноми, разгледана в предния параграф във връзка с кубичните сплайни. Решаването ѝ, обаче, не е толкова лесно.

Компютърната програма CorelDraw използва именно кубични криви на Безиє за конструирането на криви със свободна форма. Фиг. 3.2 е илюстрация на горния подход. На нея са показани кривите, които се получават при фиксирани крайни точки  $\mathbf{B}_0$  и  $\mathbf{B}_3$  и при различни допирателни вектори в тези точки.

Освен кривите на Безиє ще споменем два други типа криви, намиращи приложение в CAD/CAE компютърните програми. Нека е дадена редицата  $t_1 < t_2 < \dots < t_r < t_{r+1} < \dots < t_{N+r}$ . За всеки  $r+1$  последователни точки от тази редица дефинираме  $B$ -сплайн от степен  $r-1$ :

$$B_k(t) = B(t_k, t_{k+1}, \dots, t_{k+r}; t), \quad k = 1, 2, \dots, N.$$

Ако в уравненията (3.5) заменим полиномите  $\binom{n}{k}t^k(1-t)^{n-k}$  с  $B$ -сплайните  $B_k(t)$  от степен  $r-1$ , ще получим параметричен сплайн, в



Фигура 3.2

който всяка от координатните функции е линейна комбинация на  $B$ -сплайни:

$$\gamma : \begin{cases} x(t) = \sum_{k=1}^N x_k B_k(t) \\ y(t) = \sum_{k=1}^N y_k B_k(t) \end{cases}, \quad t \in [t_1, t_N].$$

Така дефинираната крива  $\gamma$  се нарича *B-сплайн крива*.

Нека са дадени числата  $w_i > 0$ ,  $i = 1, 2, \dots, N$ , на които ще гледаме като на теглови коефициенти. С помощта на горните  $B$ -сплайни  $B_k(t)$  от степен  $r - 1$  да дефинираме *рационалните B-сплайни*

$$R_k(t) = \frac{w_k B_k(t)}{\sum_{i=1}^N w_i B_i(t)}, \quad k = 1, 2, \dots, N.$$

*NURBS-крива* се нарича крива, определена по следния начин:

$$\gamma : \begin{cases} x(t) = \sum_{k=1}^N x_k R_k(t) \\ y(t) = \sum_{k=1}^N y_k R_k(t) \end{cases}, \quad t \in [t_1, t_N].$$

За сведение, NURBS идва от съкращението на английското наименование за неравномерен рационален  $B$ -сплайн. Тук „неравномерността“ е в смисъл, че мрежата от възли  $t_1 < t_2 < \dots < t_{N+r}$  по която сме дефинирали  $B$ -сплайните и след това рационалните  $B$ -сплайни може да е с различна стъпка между всеки два последователни възела.

**Забележка.** По съвсем същия начин, добавяйки още една координатна функция  $z(t)$  от съответния тип, се дефинират пространствени криви на Безиє,  $B$ -сплайн криви и  $NURBS$ -криви.

Ще подчертаем още веднъж голямото практическо приложение на разгледаните в този параграф параметрични сплайни. В съвременните CAD/CAE програми има създадени пакети, работещи с криви на Безиє, с  $B$ -сплайн криви и с  $NURBS$ -криви. Те се използват за описване и визуализиране на сложни геометрически обекти – както равнинни, така и пространствени.

## §4. МЕТОД НА НАЙ-МАЛКИТЕ КВАДРАТИ

### 1. Най-добри приближения с полиноми.

Нека в резултат на някакви изследвания сме получили данни за функция  $y(x)$  в определени точки, т.е. известни са стойностите

$$y_i = y(x_i), \quad i = 1, 2, \dots, K$$

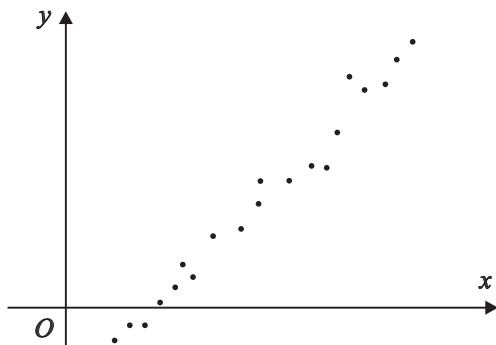
в точките  $x_i, i = 1, 2, \dots, K$ . Нашата цел ще бъде да приближим тези данни със сравнително проста гладка функция. На търсената функция ще позволим да се различава от регистрираните стойности  $y_i$  за сметка на простия ѝ вид. Този процес се нарича най-общо *изглаждане на данни*.

Да предположим, че данните от измерванията, които сме извършили са както на фиг. 4.1. От разположението им може да се направи заключението, че всъщност имаме работа с линейна зависимост на стойностите от аргумента.

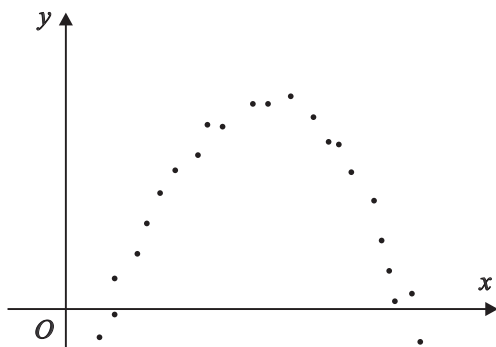
В такъв случай, вместо да се интерполира с полином от твърде висока степен или със сплайн-функция (което изисква не малко изчисления), много по-удобно ще бъде да намерим линейна функция, която има поведението на данните от фигурата. Геометрично можем да прекараме колкото си искаме прави, които визуално добре да изглаждат данните. Методът който ще разгледаме ще ни даде алгоритъм, по който ще намерим тази права, която в някакъв смисъл най-добре приближава нашите данни.

Ако имаме данни както на фиг. 4.2, естествено е да потърсим „най-добрия“ полином от втора степен. Така поведението на данните може да ни подсказва от какъв вид е търсената зависимост. Между другото, много от непрекъснатите процеси в практиката протичат по линейна или квадратична, т.е. полиномиална зависимост.





Фигура 4.1



Фигура 4.2

По-общо, разглеждаме задачата за намирането на алгебричен полином  $p(x)$  от степен  $n$ , приближаващ „най-добре“ в определен смисъл данните

$$\begin{array}{c|c|c|c} x_1 & x_2 & \dots & x_K \\ \hline y_1 & y_2 & \dots & y_K \end{array}. \quad (4.1)$$

Изразът

$$\sum_{i=1}^K (p(x_i) - y_i)^2 \quad (4.2)$$

представлява една обобщена грешка от изглаждането, отчитащ грешката която ще направим във всяка една от точките  $x_i$ ,  $i = 1, 2, \dots, K$ . Тогава да разгледаме задачата за намиране на полином  $p^*(x) \in \mathcal{P}_n$ , за

който изразът (4.2) приема възможно най-малка стойност. Въпросът за съществуване и единственост на такъв полином е решен положително от следващата теорема.

**Теорема 4.1.** *Нека  $n$  е произволно естествено число. При всеки избор на данните (4.1), съществува и то единствен полином  $p^*(x) \in \mathcal{P}_n$ , такъв че*

$$\sum_{i=1}^K (p^*(x_i) - y_i)^2 = \min_{p \in \mathcal{P}_n} \sum_{i=1}^K (p(x_i) - y_i)^2 .$$

Полиномът  $p^*(x)$  от Теорема 4.1 се нарича *полином на най-добро приближение по метода на най-малките квадрати от степен  $n$  за данните (4.1)*, а сега ще разясним и самия *метод на най-малките квадрати* за намирането на  $p^*(x)$ .

Да запишем  $p(x) \in \mathcal{P}_n$  с неопределени коефициенти във вида

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 .$$

Изразът (4.2) може да се разглежда като функция на коефициентите  $a_0, a_1, \dots, a_n$ , определящи еднозначно полинома, т.е.

$$F(a_0, a_1, \dots, a_n) = \sum_{i=1}^K (a_n x_i^n + a_{n-1} x_i^{n-1} + \dots + a_0 - y_i)^2 . \quad (4.3)$$

По такъв начин задачата за намиране на полинома на най-добро приближение се свежда до задача за намиране на минимум на функция на няколко променливи. От необходимите условия за екстремум следва, че коефициентите на полинома на най-добро приближение  $p^*(x)$  трябва да удовлетворяват системата уравнения:

$$\left\{ \begin{array}{l} \frac{\partial F}{\partial a_n} (a_0, a_1, \dots, a_n) = 0 \\ \frac{\partial F}{\partial a_{n-1}} (a_0, a_1, \dots, a_n) = 0 \\ \dots \dots \dots \\ \frac{\partial F}{\partial a_0} (a_0, a_1, \dots, a_n) = 0 \end{array} \right. . \quad (4.4)$$

Тъй като за  $j = n, n-1, \dots, 0$  имаме

$$\begin{aligned}\frac{\partial F}{\partial a_j} &= \sum_{i=1}^K (a_n x_i^n + a_{n-1} x_i^{n-1} + \dots + a_0 - y_i) x_i^j \\ &= a_n \sum_{i=1}^K x_i^{n+j} + a_{n-1} \sum_{i=1}^K x_i^{n+j-1} + \dots + a_0 \sum_{i=1}^K x_i^j - \sum_{i=1}^K x_i^j y_i,\end{aligned}$$

то след внимателно преобразуване системата (4.4) приема вида

$$\left\{ \begin{aligned} a_n \sum_{i=1}^K x_i^{2n} + a_{n-1} \sum_{i=1}^K x_i^{2n-1} + \dots + a_0 \sum_{i=1}^K x_i^n &= \sum_{i=1}^K x_i^n y_i \\ a_n \sum_{i=1}^K x_i^{2n-1} + a_{n-1} \sum_{i=1}^K x_i^{2n-2} + \dots + a_0 \sum_{i=1}^K x_i^{n-1} &= \sum_{i=1}^K x_i^{n-1} y_i \\ \dots &\dots \\ a_n \sum_{i=1}^K x_i^n + a_{n-1} \sum_{i=1}^K x_i^{n-1} + \dots + a_0 K &= \sum_{i=1}^K y_i \end{aligned} \right. ,$$

с неизвестни  $a_0, a_1, \dots, a_n$ .

Може да се докаже, че детерминантата на тази линейна система е различна от нула, стига всички точки  $x_1, x_2, \dots, x_K$  да са различни. Следователно, например по формулите на Крамер, можем да намерим единственото решение на тази система, което ни дава коефициентите на полинома на най-добро приближение.

**Забележка.** В общия случай, решението на системата (4.4) ни дава локален екстремум на функцията  $F(a_0, a_1, \dots, a_n)$ . Поради специалния вид на функцията  $F$  обаче, в този случай единственото решение на системата (4.4) е точка на минимум за функцията  $F$ , която освен това се явява и глобален минимум.

За удобство ще напишем горната система в някои частни случаи.

При  $n = 1$  коефициентите на полинома на най-добро приближение от първа степен  $p^*(x) = a_1 x + a_0$  намираме от системата:

$$\left\{ \begin{aligned} a_1 \sum_{i=1}^K x_i^2 + a_0 \sum_{i=1}^K x_i &= \sum_{i=1}^K x_i y_i \\ a_1 \sum_{i=1}^K x_i + a_0 K &= \sum_{i=1}^K y_i \end{aligned} \right. \quad (4.5)$$

При  $n = 2$  коефициентите на полинома на най-добро приближение

от втора степен  $p^*(x) = a_2x^2 + a_1x + a_0$  намираме от системата:

$$\begin{cases} a_2 \sum_{i=1}^K x_i^4 + a_1 \sum_{i=1}^K x_i^3 + a_0 \sum_{i=1}^K x_i^2 = \sum_{i=1}^K x_i^2 y_i \\ a_2 \sum_{i=1}^K x_i^3 + a_1 \sum_{i=1}^K x_i^2 + a_0 \sum_{i=1}^K x_i = \sum_{i=1}^K x_i y_i \\ a_2 \sum_{i=1}^K x_i^2 + a_1 \sum_{i=1}^K x_i + a_0 K = \sum_{i=1}^K y_i \end{cases} \quad (4.6)$$

**Пример 4.1.** Да се намери по метода на най-малките квадрати полиномът на най-добро приближение от втора степен за данните:

$x_i$	-4	-2	0	1	2	3
$y_i$	-8	-2	3	3	-1	-7

В случая имаме  $n = 2$  и  $K = 6$ . Пресмятаме сумите:

$$\begin{aligned} \sum_{i=1}^6 x_i^4 &= (-4)^4 + (-2)^4 + 0^4 + 1^4 + 2^4 + 3^4 = 370, \\ \sum_{i=1}^6 x_i^3 &= (-4)^3 + (-2)^3 + 0^3 + 1^3 + 2^3 + 3^3 = -36, \\ \sum_{i=1}^6 x_i^2 &= (-4)^2 + (-2)^2 + 0^2 + 1^2 + 2^2 + 3^2 = 34, \\ \sum_{i=1}^6 x_i &= (-4) + (-2) + 0 + 1 + 2 + 3 = 0, \\ \sum_{i=1}^6 x_i^2 y_i &= (-4)^2 \cdot (-8) + (-2)^2 \cdot (-2) + 0^2 \cdot 3 + 1^2 \cdot 3 + 2^2 \cdot (-1) + 3^2 \cdot (-7) \\ &= -200, \\ \sum_{i=1}^6 x_i y_i &= (-4) \cdot (-8) + (-2) \cdot (-2) + 0 \cdot 3 + 1 \cdot 3 + 2 \cdot (-1) + 3 \cdot (-7) = 16, \\ \sum_{i=1}^6 y_i &= (-8) + (-2) + 3 + 3 + (-1) + (-7) = -12. \end{aligned}$$

Системата (4.6) има вида

$$\begin{cases} 370a_2 - 36a_1 + 34a_0 = -200 \\ -36a_2 + 34a_1 + 0a_0 = 16 \\ 34a_2 + 0a_1 + 6a_0 = -12 \end{cases} .$$

Оттук намираме  $a_2 = -\frac{1467}{1775}$ ,  $a_1 = -\frac{718}{1775}$ ,  $a_0 = \frac{4763}{1775}$ . Следователно търсеният полином е

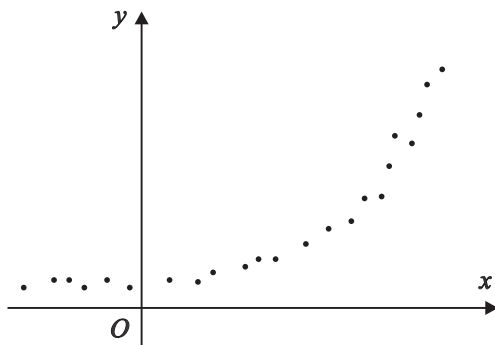
$$p^*(x) = -\frac{1467}{1775}x^2 - \frac{718}{1775}x + \frac{4763}{1775} . \quad \square$$

## 2. Модификации.

Да предположим сега, че имаме данните

$$\begin{array}{c|c|c|c} x_1 & x_2 & \dots & x_K \\ \hline z_1 & z_2 & \dots & z_K \end{array} , \quad (4.7)$$

показани на фиг. 4.3.



Фигура 4.3

Поведението на тези данни ни навежда на мисълта, че ще е подходящо да ги изглаждаме с експоненциална функция от определен тип. Ще скицираме накратко един метод за такова изглаждане, базиран на метода на най-малките квадрати.

Да логаритмуваме стойностите:  $y_j = \ln z_j$ ,  $j = 1, 2, \dots, K$ . По метода на най-малките квадрати да намерим полинома на най-добро приближение от първа степен  $p^*(x) = ax + b$  за данните

$$\begin{array}{c|c|c|c} x_1 & x_2 & \dots & x_K \\ \hline y_1 & y_2 & \dots & y_K \end{array} . \quad (4.8)$$

Товагава данните (4.7) изглаждаме с функцията

$$\varphi(x) = e^{ax+b}.$$

Ясно е, че изглаждане за (4.7) с функция от вида  $\psi(x) = e^{ax^2+bx+c}$  ще получим, ако за (4.8) по метода на най-малките квадрати намерим полинома на най-добро приближение от втора степен  $p^*(x) = ax^2+bx+c$ .

При периодични данни (4.7), чието поведение напомня синусоида, можем по подобен начин да изгладим данните (4.8), при  $y_j = \arcsin z_j$ ,  $j = 1, 2, \dots, K$ , с полином от първа степен  $p^*(x) = ax + b$ . Така получаваме изглаждаща функция  $\varphi(x) = \sin(ax + b)$ . Тук може да се наложи първо да нормираме по подходящ начин стойностите в (4.7).

Тази схема за изглаждане на данни можем да приложим и като използваме функции от други класове.

### 3. Приближено решаване на преопределени системи.

Да разгледаме система от  $k$  линейни уравнения с  $n$  неизвестни

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n = b_k \end{cases}, \quad (4.9)$$

където коефициентите  $a_{ij}$ ,  $b_i$  за  $i = 1, 2, \dots, k$ ,  $j = 1, 2, \dots, n$  са дадени. При стандартните матрични означения

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kn} \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix},$$

тази система можем да запишем във вида

$$AX = B. \quad (4.10)$$

Нататък с  $A'$  ще бележим транспонираната на матрицата  $A$ .

Системата (4.9) се нарича *преопределена*, ако броят на линейно независимите уравнения в тази система е по-голям от броя на неизвестните  $n$ . Нататък ще считаме, че всички  $k$  уравнения в (4.9) са линейно независими.

Както е известно, такава система няма решение. Въпреки това, ние ще видим как може да се решават преопределени системи, макар и приближено. За приближено решение на (4.9) може да ни служи произволна

$n$ -торка числа. Така обаче може да се окаже, че при всички уравнения едната страна е много по-голяма от другата, т.е. уравненията се удовлетворяват с голяма грешка. Следователно, ще искаме нашето приближено решение да удовлетворява с възможно най-малка грешка системата. Естествено възниква въпросът как да дефинираме тази грешка и кое ще бъде „най-доброто“ приближено решение на системата.

Ще използваме отново методът на най-малките квадрати. Да разгледаме функцията

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^k (a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - b_i)^2. \quad (4.11)$$

Това е точно сумата от квадратите на грешките, които ще направим във всяко едно уравнение на системата, ако за приближено решение изберем  $(x_1, x_2, \dots, x_n)$ . Минимизирането на тази обобщена грешка води до задача за минимизиране на функция на няколко променливи. От необходимото условие за екстремум следва, че в точката на минимум на  $F$  имаме:

$$\left\{ \begin{array}{l} \frac{\partial F}{\partial x_1}(x_1, x_2, \dots, x_n) = 0 \\ \frac{\partial F}{\partial x_2}(x_1, x_2, \dots, x_n) = 0 \\ \dots\dots\dots \\ \frac{\partial F}{\partial x_n}(x_1, x_2, \dots, x_n) = 0 \end{array} \right. . \quad (4.12)$$

Тъй като за  $j = 1, 2, \dots, n$  имаме

$$\begin{aligned} \frac{\partial F}{\partial x_j} &= \sum_{i=1}^k (a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - b_i)a_{ij} = \\ &= x_1 \sum_{i=1}^k a_{i1}a_{ij} + x_2 \sum_{i=1}^k a_{i2}a_{ij} + \dots + x_n \sum_{i=1}^k a_{in}a_{ij} - \sum_{i=1}^k b_i a_{ij}, \end{aligned}$$





Системата (4.13) има вида

$$\begin{cases} 23x - 5y = 35 \\ -5x + 4y = -13 \end{cases} ,$$

откъдето намираме  $x = \frac{75}{67}$ ,  $y = -\frac{125}{67}$ . Тогава най-доброто приближено решение по метода на най-малките квадрати на системата (4.14) е

$$x = \frac{75}{67} \approx 1,119 , \quad y = -\frac{125}{67} \approx -1,866 . \quad \square$$

## §5. СРЕДНО-КВАДРАТИЧНИ ПРИБЛИЖЕНИЯ

В предишния параграф показахме как с метода на най-малките квадрати можем да приближаваме дискретни данни. Подобен подход ще приложим и за решаването на следната задача: дадена е функция  $f(x)$ ,  $x \in [a, b]$ ; търсим функция  $\varphi(x)$  от определен клас, която е „най-близо“ относно средно-квадратичното разстояние до дадената  $f(x)$  за  $[a, b]$ .

Да разгледаме по-подробно тази задача, когато търсим полином от фиксирана степен  $n$ , т.е. измежду полиномите  $p(x) \in \mathcal{P}_n$  търсим такъв, за който се достига

$$\min_{p \in \mathcal{P}_n} \|f - p\|_2 = \min_{p \in \mathcal{P}_n} \sqrt{\int_a^b |f(x) - p(x)|^2 dx} . \quad (5.1)$$

**Теорема 5.1.** Нека  $n$  е произволно естествено число. За всяка функция  $f \in L_2[a, b]$  съществува и то единствен полином  $p^* \in \mathcal{P}_n$ , такъв че

$$\|f - p^*\|_2 = \min_{p \in \mathcal{P}_n} \|f - p\|_2 .$$

Полиномът  $p^*(x)$  от Теорема 5.1 се нарича *полином на най-добро средноквадратично приближение от степен  $n$  за функцията  $f(x)$  в интервала  $[a, b]$* .

Лесно се съобразява, че ако за един полином от  $\mathcal{P}_n$  се достига минимумът в (5.1), то за него се достига и минимумът в

$$\|f - p\|_2^2 = \int_a^b [f(x) - p(x)]^2 dx , \quad (5.2)$$

и обратно. Записвайки  $p(x) \in \mathcal{P}_n$  с неопределени коефициенти във вида

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 ,$$

изразът (5.2) може да се разглежда като функция на коефициентите  $a_0, a_1, \dots, a_n$ :

$$F(a_0, a_1, \dots, a_n) = \int_a^b [f(x) - a_n x^n - a_{n-1} x^{n-1} - \dots - a_0]^2 dx . \quad (5.3)$$

Така задачата за намиране на полином на най-добро средноквадратично приближение се свежда до задача за намиране на минимум на функция на няколко променливи. От необходимите условия за екстремум следва, че коефициентите на полинома на най-добро приближение  $p^*(x)$  трябва да удовлетворяват системата уравнения:

$$\left\{ \begin{array}{l} \frac{\partial F}{\partial a_n} (a_0, a_1, \dots, a_n) = 0 \\ \frac{\partial F}{\partial a_{n-1}} (a_0, a_1, \dots, a_n) = 0 \\ \dots \dots \dots \\ \frac{\partial F}{\partial a_0} (a_0, a_1, \dots, a_n) = 0 \end{array} \right. . \quad (5.4)$$

Оттук получаваме следната линейна система:

$$\left\{ \begin{array}{l} a_n \int_a^b x^{2n} dx + a_{n-1} \int_a^b x^{2n-1} dx + \dots + a_0 \int_a^b x^n dx = \int_a^b x^n f(x) dx \\ a_n \int_a^b x^{2n-1} dx + a_{n-1} \int_a^b x^{2n-2} dx + \dots + a_0 \int_a^b x^{n-1} dx = \int_a^b x^{n-1} f(x) dx \\ \dots \dots \dots \\ a_n \int_a^b x^n dx + a_{n-1} \int_a^b x^{n-1} dx + \dots + a_0 \int_a^b dx = \int_a^b f(x) dx \end{array} \right.$$

с неизвестни  $a_0, a_1, \dots, a_n$ .

Може да се докаже, че детерминантата на тази система е различна от нула. Тогава от теоремата на Крамер следва, че тази система има единствено решение. Това единственото решение на на тази система, което ни дава коефициентите на полинома на най-добро средноквадратично приближение.

**Забележка.** В общия случай, решението на системата (5.4) ни дава локален екстремум на функцията  $F(a_0, a_1, \dots, a_n)$ . Поради специалния вид на функцията  $F$ , обаче, в този случай единственото решение на системата (5.4) е точка на минимум за функцията  $F$ , който при това се явява и глобален минимум.

При  $n = 1$  коефициентите на полинома на най-добро приближение от първа степен  $p^*(x) = a_1x + a_0$  намираме от системата:

$$\begin{cases} a_1 \int_a^b x^2 dx + a_0 \int_a^b x dx = \int_a^b x f(x) dx \\ a_1 \int_a^b x dx + a_0 \int_a^b dx = \int_a^b f(x) dx \end{cases} \quad (5.5)$$

При  $n = 2$  коефициентите на полинома на най-добро приближение от втора степен  $p^*(x) = a_2x^2 + a_1x + a_0$  намираме от системата:

$$\begin{cases} a_2 \int_a^b x^4 dx + a_1 \int_a^b x^3 dx + a_0 \int_a^b x^2 dx = \int_a^b x^2 f(x) dx \\ a_2 \int_a^b x^3 dx + a_1 \int_a^b x^2 dx + a_0 \int_a^b x dx = \int_a^b x f(x) dx \\ a_2 \int_a^b x^2 dx + a_1 \int_a^b x dx + a_0 \int_a^b dx = \int_a^b f(x) dx \end{cases} \quad (5.6)$$

**Пример 5.1.** Да се намери полиномът на най-добро средноквадратично приближение от първа степен  $p^*(x) = a_1x + a_0$  за функцията  $f(x) = e^x$  в интервала  $[0, 1]$ .

В случая  $n = 1$ . Пресмятаме необходимите интеграли:

$$\begin{aligned} \int_0^1 x^2 dx &= \frac{x^3}{3} \Big|_0^1 = \frac{1}{3}, \\ \int_0^1 x dx &= \frac{x^2}{2} \Big|_0^1 = \frac{1}{2}, \\ \int_0^1 dx &= 1 - 0 = 1, \\ \int_0^1 x e^x dx &= \int_0^1 x de^x = x e^x \Big|_0^1 - \int_0^1 e^x dx = e - 0 - e^x \Big|_0^1 = e - e + 1 = 1, \\ \int_0^1 e^x dx &= e^x \Big|_0^1 = e - 1. \end{aligned}$$

От системата (5.5), имаща вида

$$\begin{cases} \frac{1}{3} a_1 + \frac{1}{2} a_0 = 1 \\ \frac{1}{2} a_1 + a_0 = e - 1 \end{cases},$$

намираме  $a_1 = 18 - 6e$ ,  $a_0 = 4e - 10$ . Следователно търсеният полином на най-добро средноквадратично приближение е

$$p^*(x) = (18 - 6e)x + 4e - 10. \quad \square$$

## §6. РАВНОМЕРНИ ПРИБЛИЖЕНИЯ

В този параграф ще разгледаме някои въпроси за приближаване на функции с полиноми, като грешката от приближаването ще мерим с равномерното разстояние. Да припомним, че равномерно разстояние между непрекъснатите в  $[a, b]$  функции  $f$  и  $f^*$  дефинирахме по следния начин:

$$\|f - f^*\| := \max_{x \in [a, b]} |f(x) - f^*(x)|.$$

**Теорема 6.1** (Борел). *Нека  $f$  е непрекъснатата в интервала  $[a, b]$  функция. За всяко  $n = 1, 2, \dots$ , съществува полином  $p^*$  от степен  $n$ , такъв че*

$$\|f - p^*\| = \min_{p \in \mathcal{P}_n} \|f - p\| = \min_{p \in \mathcal{P}_n} \max_{x \in [a, b]} |f(x) - p(x)|. \quad (6.1)$$

Полиномът  $p^* \in \mathcal{P}_n$  от Теорема 6.1, за който се достига минимумът в (6.1), се нарича *полином на най-добро равномерно приближение от степен  $n$  за функцията  $f$  в интервала  $[a, b]$* .

Докато Теорема 6.1 решава въпроса за съществуването на полином на най-добро равномерно приближение, характеризация на този полином се дава от следващата теорема за алтернанса.

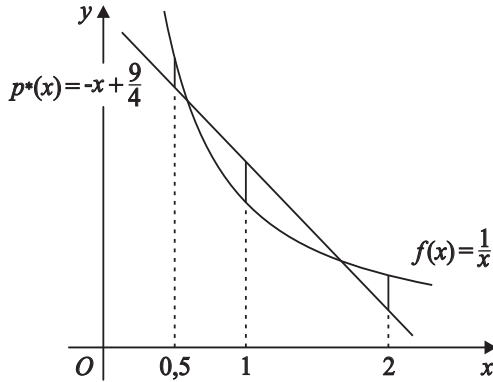
**Теорема 6.2** (Чебишов). *Нека  $f$  е непрекъснатата в интервала  $[a, b]$  функция. Полиномът  $p^* \in \mathcal{P}_n$  е полином на най-добро равномерно приближение от степен  $n$  за функцията  $f$  в  $[a, b]$  тогава и само тогава, когато съществуват  $n + 2$  точки  $\{x_k\}_{k=0}^{n+1}$ ,  $a \leq x_0 < x_1 < \dots < x_{n+1} \leq b$ , такива че*

$$f(x_i) - p^*(x_i) = (-1)^i \sigma \|f - p^*\|, \quad i = 0, 1, \dots, n + 1, \quad (6.2)$$

където  $\sigma = 1$  или  $\sigma = -1$ .

Следователно, за да намерим полином на най-добро равномерно приближение от фиксирана степен  $n$  за дадена функция, е достатъчно да намерим полином, за който грешката  $f(x) - p^*(x)$  приема алтернативно в  $n + 2$  точки минимум или максимум, равен по абсолютна стойност на равномерното разстояние  $\|f - p^*\|$ . За малки  $n$  със средствата на математическия анализ не е трудно да намерим точките на алтернанс и полинома на най-добро равномерно приближение.

**Пример 6.1.** Да се намери полиномът на най-добро равномерно приближение от първа степен  $p^*(x) = ax + b$  за функцията  $f(x) = \frac{1}{x}$  в интервала  $[\frac{1}{2}, 2]$ .



Фигура 6.1

Нека  $E = \|f - p^*\| = \max_{x \in [1/2, 2]} |\frac{1}{x} - ax - b|$ . В случая имаме  $n = 1$  и съгласно Теоремата на Чебишов трябва да намерим полином с три точки на алтернанс  $\frac{1}{2} \leq x_0 < x_1 < x_2 \leq 2$ . Лесно се съобразява, че поведението на  $p^*(x)$  трябва да е като на фиг. 6.1. Поради монотонността на разликата  $f(x) - p^*(x)$  в краищата на разглеждания интервал е ясно, че  $x_0 = \frac{1}{2}$  и  $x_2 = 2$ . Тогава условията за алтернанс (6.2) приемат вида:

$$f\left(\frac{1}{2}\right) - p^*\left(\frac{1}{2}\right) = 2 - \frac{a}{2} - b = E ,$$

$$f(x_1) - p^*(x_1) = \frac{1}{x_1} - ax_1 - b = -E ,$$

$$f(2) - p^*(2) = \frac{1}{2} - 2a - b = E .$$

С почленно изваждане на третото уравнение от първото получаваме  $\frac{3}{2} + \frac{3a}{2} = 0$ , откъдето  $a = -1$ .

Разликата  $r(x) := f(x) - p^*(x) = \frac{1}{x} - ax - b$  има локален екстремум в някаква точка  $x_1$ . В такъв случай,  $r'(x_1) = 0$  и  $x_1$  е корен на уравнението  $r'(x) = 0$ , т.е. на  $\frac{-1}{x^2} + 1 = 0$ . Оттук,  $x_1 = 1$ .

Заместваме в първите две уравнения на системата:

$$\begin{aligned}\frac{5}{2} - b &= E \\ 2 - b &= -E,\end{aligned}$$

откъдето намираме  $b = \frac{9}{4}$  и  $E = \frac{1}{4}$ .

Следователно търсеният полином е

$$p^*(x) = -x + \frac{9}{4}. \quad \square$$

За приближено намиране на полином на най-добро равномерно приближение се използват *алгоритмите на Ремез*. Всеки от тях е итерационен алгоритъм, в резултат на който се получава редица от полиноми от степен  $n$ :  $p_0, p_1, \dots, p_k, \dots$ , равномерно сходяща в интервала  $[a, b]$  към  $p^*$ .

Ще отбележим още няколко свойства на полинома на най-добро равномерно приближение.

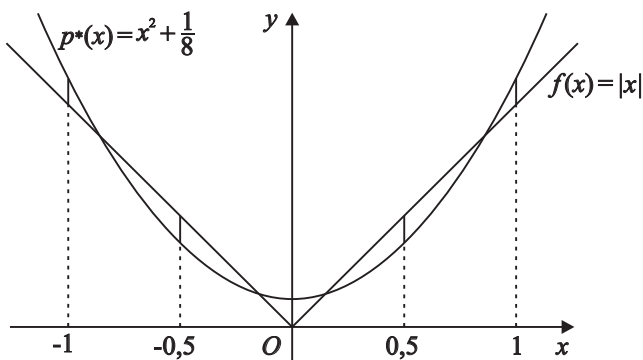
**Теорема 6.3.** Нека  $f$  е непрекъснатата в интервала  $[a, b]$  функция. Полиномът на най-добро равномерно приближение от степен  $n$  за функцията  $f$  в интервала  $[a, b]$  е единствен.

**Теорема 6.4.** Нека  $f$  е непрекъснатата в интервала  $[-a, a]$  четна (нечетна) функция. Тогава полиномът на най-добро равномерно приближение от степен  $n$  за функцията  $f$  в интервала  $[-a, a]$  също е четен (съотв., нечетен).

**Пример 6.2.** Да се намери полиномът на най-добро равномерно приближение от трета степен  $p^*(x)$  за функцията  $f(x) = |x|$  в интервала  $[-1, 1]$ .

В този случай имаме  $n = 3$ . Нека  $E = \|f - p^*\| = \max_{x \in [-1, 1]} ||x| - p^*(x)|$ .

Съгласно Теорема 6.4,  $p^*(x)$  трябва да е четен, тъй като функцията  $f(x) = |x|$  е четна в  $[-1, 1]$ , т.е. в представянето на  $p^*(x)$  ще участват само четните степени на  $x$ . Тогава търсеният полином ще има вида  $p^*(x) = ax^2 + b$ . Съгласно Теоремата на Чебишов  $p^*(x)$  ще има и пет точки на алтернанс  $-1 \leq x_0 < x_1 < x_2 < x_3 < x_4 \leq 1$ . Лесно се съобразява, че поведението на  $p^*(x)$  трябва да е като на фиг. 6.2. Поради симетрията и монотонността на разликата  $f(x) - p^*(x)$  в краищата на разглеждания интервал е ясно, че  $x_0 = -1$ ,  $x_2 = 0$ ,  $x_4 = 1$  и  $x_3 = -x_1$ .



Фигура 6.2

Тогава условията за алтернанс (6.2) приемат вида:

$$\begin{aligned} f(0) - p^*(0) &= -b = -E, \\ f(x_3) - p^*(x_3) &= x_3 - ax_3^2 - b = E, \\ f(1) - p^*(1) &= 1 - a - b = -E. \end{aligned}$$

Пропускаме условията в точките  $x_0 = -1$  и  $x_1$ , тъй като те ще доведат до същите уравнения, както третото и второто. С почленно изваждане на третото уравнение от първото получаваме  $a = 1$ .

Разликата  $r(x) := f(x) - p^*(x) = |x| - ax^2 - b$  има локален екстремум в точката  $x_3$ . Тогава  $r'(x_3) = 0$  и  $x_3$  е корен на уравнението  $r'(x) = 0$ , т.е. за  $x > 0$ , на  $1 - 2x = 0$ . Оттук,  $x_3 = -\frac{1}{2}$ .

Заместваме в първите две уравнения на системата:

$$\begin{aligned} b &= E \\ \frac{1}{4} - b &= E, \end{aligned}$$

откъдето намираме  $b = E = \frac{1}{8}$ . Търсеният полином на най-добро равномерно приближение е

$$p^*(x) = x^2 + \frac{1}{8}. \quad \square$$

Съвсем естествен е въпросът: можем ли да приближим една непрекъсната функция с произволна точност относно равномерното разстояние? Отговорът се съдържа в следващата теорема.

**Теорема 6.5.** Нека  $f$  е непрекъсната в интервала  $[a, b]$  функция. За всяко  $\varepsilon > 0$  съществува полином  $p$ , такъв че

$$\|f - p\| < \varepsilon .$$

Следователно, с алгебрични полиноми можем да приближим колкото си искаме добре дадена непрекъсната функция.

**Пример 6.3.** Нека функцията  $f$  е непрекъсната в интервала  $[0, 1]$ . Да разгледаме редицата от полиноми, дефинирани по следния начин:

$$B_n(f; x) = \sum_{k=0}^n f\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k}, \quad x \in [0, 1] .$$

Полиномите  $B_n(f; x)$  се наричат *полиноми на Бернщайн*. Те са линейни комбинации на полиномите  $\binom{n}{k} x^k (1-x)^{n-k}$ , които вече използвахме в § 3 при конструирането на кривите на Безиє.

Може да се докаже, че

$$\|f - B_n(f; \cdot)\| = O\left(\frac{1}{\sqrt{n}}\right) .$$

Доказателството на Теорема 6.5 се основава на този факт и на други свойства на полиномите на Бернщайн. Следователно за достатъчно голямо  $n$ , с полином на Бернщайн можем да постигнем каквато си искаме предварително зададена точност  $\varepsilon$  спрямо равномерното разстояние в  $[0, 1]$ . Тогава редицата от полиноми  $B_n(f; x)$ ,  $n = 0, 1, \dots$ , е равномерно сходяща в  $[0, 1]$  към функцията  $f(x)$ , а следователно и

$$\lim_{n \rightarrow \infty} B_n(f; x) = f(x), \quad x \in [0, 1] . \quad \square$$

Използвайки линейна трансформация, лесно можем да напишем как ще изглеждат полиномите на Бернщайн за произволен интервал  $[a, b]$ .

## §7. ПРИБИЖЕНО РЕШАВАНЕ НА НЕЛИНЕЙНИ УРАВНЕНИЯ

### 1. Нелинейни уравнения.

Намирането на точно реално решение (реален корен) на нелинейно уравнение е в общия случай трудна задача. Поради това, че в практиката се работи не с точни стойности на величини, а с техни приближения,



и при решаването на нелинейни уравнения можем да потърсим достатъчно добро приближение на корена. Действително, има разработени методи за приближено решаване на нелинейни уравнения и в този параграф ще разгледаме някои от тях.

Задачата за намиране на реално решение  $\xi$  на дадено *нелинейно уравнение*

$$f(x) = 0, \quad (7.1)$$

т.е. уравнение, в което функцията  $f(x)$  не е от вида  $ax + b$ , се разделя на две части.

1) Локализиране на корена. Това означава, използвайки вида на функцията  $f(x)$ , графиката ѝ, изследвания със средствата на математическия анализ и др., да намерим относително „малък“ интервал  $[a, b]$ , който съдържа корена  $\xi$  на уравнението (7.1).

2) Уточняване на корена. Включва избор на начално приближение  $x_0$  и построяване на редица  $x_0, x_1, \dots, x_n, \dots$ , обикновено в резултат на *итерационен процес*, такава че

$$\lim_{n \rightarrow \infty} x_n = \xi.$$

На практика се пресмятат краен брой последователни приближения и при предварително избрана *точност*  $\varepsilon$ , пресмятанията се преустановяват при първото  $n$ , за което се изпълни условието

$$|x_n - x_{n-1}| < \varepsilon.$$

Тогава за приближено решение на уравнението се избира последното намерено приближение, т.е.  $\xi \approx x_n$ .

За сравняване на скоростта на сходимост при различните методи се използва следната характеристика. Казваме, че итерационният процес, водещ до редицата от последователни приближения  $x_0, x_1, \dots, x_n, \dots$  на корена  $\xi$  на уравнението (7.1), има *ред на сходимост*  $p$  ( $p > 1$ ), ако съществуват числа  $q \in (0, 1)$  и  $C > 0$ , такива че

$$|\xi - x_n| \leq Cq^{p^n}, \quad n = 0, 1, 2, \dots$$

## 2. Метод на разполовяването.

Разглеждаме уравнението

$$f(x) = 0,$$

където функцията  $f(x)$  е непрекъсната в интервала  $[a, b]$  и  $f(a)f(b) < 0$ . Тези условия гарантират съществуването на корен в  $[a, b]$ , който може да не е единствен. Ще предположиме, че в резултат на локализиране сме намерили достатъчно малък интервал  $[a, b]$ , в който даденото уравнение има единствен корен  $\xi$ .

Пресмятаме стойността на функцията  $f$  в средната точка на интервала  $[a, b]$ , именно  $x = \frac{a+b}{2}$ . Тогава, поради факта, че  $f$  приема стойности с различни знаци в краищата  $a$  и  $b$  на интервала, имаме следните възможни случаи.

1) Ако  $f\left(\frac{a+b}{2}\right) = 0$ , то ние сме намерили точно корена  $\xi = \frac{a+b}{2}$ .

2) Ако  $f(a)f\left(\frac{a+b}{2}\right) < 0$ , то  $\xi \in \left(a, \frac{a+b}{2}\right)$  и полагаме

$$a_1 = a, \quad b_1 = \frac{a+b}{2}.$$

3) Ако  $f\left(\frac{a+b}{2}\right)f(b) < 0$ , то  $\xi \in \left(\frac{a+b}{2}, b\right)$ . Тук полагаме

$$a_1 = \frac{a+b}{2}, \quad b_1 = b.$$

Така в случай, че не сме намерили точната стойност на корена получаваме интервал  $[a_1, b_1]$ , съдържащ корена  $\xi$  и два пъти по-малък от началния. В този интервал отново  $f$  е непрекъсната и сменя знаците си в краищата му. Повтаряме тази процедура многократно и получаваме редица от вложени интервали

$$[a, b] \supset [a_1, b_1] \supset [a_2, b_2] \supset \dots \supset [a_n, b_n] \supset \dots,$$

всеки от които е два пъти по-малък от предходния и съдържа корена  $\xi$ . След  $n$ -тото разполовяване на началния интервал получаваме интервал с дължина

$$b_n - a_n = \frac{b_{n-1} - a_{n-1}}{2} = \frac{b_{n-2} - a_{n-2}}{4} = \dots = \frac{b - a}{2^n},$$

съдържащ корена. Следователно при

$$n > \log_2 \frac{b-a}{\varepsilon} \iff \frac{b-a}{2^n} < \varepsilon,$$

избирайки  $x_n$  да бъде произволно число от интервала  $[a_n, b_n]$ , ще имаме

$$|\xi - x_n| \leq b_n - a_n = \frac{b-a}{2^n} < \varepsilon$$

и  $\xi \approx x_n$  с грешка по-малка от  $\varepsilon$ .

Методът на разполовяване е лесен за програмиране и винаги сходящ при наложените условия за функцията  $f$ . За разлика от следващите методи, които ще изложим, той е сравнително бавно-сходящ. Както показахме по-горе, методът на разполовяване има скорост на сходимост на геометрична прогресия с частно  $q = \frac{1}{2}$ .

**Пример 7.1.** С метода на разполовяването да се намери приближено с точност  $\varepsilon = 0,001$  положителният корен на уравнението

$$(x+1)^2 - \operatorname{arctg} x - 4 = 0.$$

Нека  $f(x) = (x+1)^2 - \operatorname{arctg} x - 4$ . Понеже

$$f(1) = 4 - \frac{\pi}{4} - 4 = -\frac{\pi}{4} < 0 \quad \text{и} \quad f(2) = 9 - \operatorname{arctg} 2 - 4 > 9 - \frac{\pi}{2} > 0,$$

функцията  $f(x)$  си сменя знака в интервала  $[1, 2]$ . От вида на производната  $f'(x) = 2(x+1) - \frac{1}{1+x^2}$  лесно се съобразява, че  $f'(x) > 0$  при  $x \in [0, \infty)$  и тогава  $f(x)$  е строго растяща в  $[0, \infty)$ . От казаното следва, че  $f(x)$  има единствен положителен корен и той е в интервала  $[1, 2]$ .

Пресмятанията показват

$$\begin{array}{ll} [a_1, b_1] = [1; 1,5], & [a_6, b_6] = [1,20313; 1,21875], \\ [a_2, b_2] = [1; 1,25], & [a_7, b_7] = [1,20313; 1,21094], \\ [a_3, b_3] = [1,125; 1,25], & [a_8, b_8] = [1,20703; 1,21094], \\ [a_4, b_4] = [1,1875; 1,25], & [a_9, b_9] = [1,20898; 1,21094], \\ [a_5, b_5] = [1,1875; 1,21875], & [a_{10}, b_{10}] = [1,20898; 1,20996], \end{array}$$

като  $b_{10} - a_{10} = 1,20996 - 1,20898 = 0,00098 < 0,001$ . За приближение на корена можем да изберем кое да е число от последния интервал  $[a_{10}, b_{10}] = [1,20898; 1,20996]$ . Обикновено се избира средата, т.е.

$$\xi \approx \frac{a_{10} + b_{10}}{2} = \frac{1,20898 + 1,20996}{2} = 1,20947.$$

Бяха необходими 10 итерации с метода на разполовяването, за да получим приближение на корена с точност  $\varepsilon = 0,001$  след като сме го локализирали в интервал  $[a, b] = [1, 2]$  с дължина  $b - a = 1$ .  $\square$

### 3. Метод на хордите.

Нека за функцията  $f$  в уравнението

$$f(x) = 0$$

са изпълнени условията:

- (i)  $f(a)f(b) < 0$ ;
- (ii)  $f \in C^2[a, b]$ , т.е.  $f$  има непрекъснатата втора производна в  $[a, b]$ ;
- (iii)  $f'$  и  $f''$  имат постоянен знак в  $[a, b]$ .

Тогава уравнението  $f(x) = 0$  има точно един корен  $\xi$  в интервала  $[a, b]$ .

Условията (i)-(iii) определят четири възможни случая, според това дали функцията  $f$  е растяща ( $f'(x) > 0$ ) или намаляваща ( $f'(x) < 0$ ), изпъкнала ( $f''(x) > 0$ ) или вдлъбната ( $f''(x) < 0$ ). Именно,

- 1)  $f'(x) > 0$  и  $f''(x) > 0$ , в който случай  $f(a) < 0$  и  $f(b) > 0$ ;
- 2)  $f'(x) > 0$  и  $f''(x) < 0$ , в който случай  $f(a) < 0$  и  $f(b) > 0$ ;
- 3)  $f'(x) < 0$  и  $f''(x) > 0$ , в който случай  $f(a) > 0$  и  $f(b) < 0$ ;
- 4)  $f'(x) < 0$  и  $f''(x) < 0$ , в който случай  $f(a) > 0$  и  $f(b) < 0$ .

За начално приближение  $x_0$  избираме този от краищата на интервала  $[a, b]$ , в който имаме  $ff'' < 0$ , т.е.

- при  $f(a)f''(a) < 0$ , полагаме  $x_0 = a$  и следващите приближения пресмятаме по рекурентните формули

$$x_n = x_{n-1} - \frac{b - x_{n-1}}{f(b) - f(x_{n-1})} f(x_{n-1}), \quad n = 1, 2, \dots ;$$

- при  $f(b)f''(b) < 0$ , полагаме  $x_0 = b$  и последователните приближения пресмятаме по формулите

$$x_n = x_{n-1} - \frac{x_{n-1} - a}{f(x_{n-1}) - f(a)} f(x_{n-1}), \quad n = 1, 2, \dots .$$

Може да се докаже, че методът на хордите е сходящ със скорост на геометрична прогресия. Наименованието на този метод идва от геометричната интерпретация на последователните приближения – всяко  $x_n$  се явява пресечната точка на абсцисата с хордата, свързваща точките:  $(b, f(b))$  и  $(x_{n-1}, f(x_{n-1}))$  при  $x_0 = a$ , съответно  $(a, f(a))$  и  $(x_{n-1}, f(x_{n-1}))$  при  $x_0 = b$ .

**Пример 7.2.** С метода на хордите да се намери приближено с точност  $\varepsilon = 0,001$  положителният корен на уравнението

$$(x + 1)^2 - \operatorname{arctg} x - 4 = 0 .$$

Нека  $f(x) = (x + 1)^2 - \operatorname{arctg} x - 4$ . Както в Пример 7.2 локализираме корена  $\xi \in [a, b] = [1, 2]$ . За производните имаме

$$f'(x) = 2(x + 1) - \frac{1}{1 + x^2} > 0 , \quad f''(x) = 2 + \frac{2x}{(1 + x^2)^2} > 0 , \quad x \in [0, \infty) .$$

Функцията  $f(x)$  удовлетворява условията (i)-(iii) за интервала  $[1, 2]$ . По-неже  $f(a)f''(a) = f(1)f''(1) < 0$ , избираме  $x_0 = 1$  и последователните приближения пресмятаме по формулите

$$x_n = x_{n-1} - \frac{2 - x_{n-1}}{f(2) - f(x_{n-1})} f(x_{n-1}) , \quad n = 1, 2, \dots .$$

Имаме

$x_1 = 1,16788$  ,  $x_2 = 1,20132$  ,  $x_3 = 1,20759$  ,  $x_4 = 1,20875$  ,  $x_5 = 1,20896$  ,  
като  $x_5 - x_4 = 1,20896 - 1,20875 = 0,00021 < 0,001$ . За приближение на корена избираме

$$\xi \approx x_5 = 1,20896 .$$

С метода на хордите бяха необходими 5 итерации, за да получим приближение на корена с точност  $\varepsilon = 0,001$ , след като сме го локализирали в интервала  $[a, b] = [1, 2]$  с дължина  $b - a = 1$ .  $\square$

#### 4. Метод на Нютон.

Нека за функцията  $f$  в уравнението

$$f(x) = 0 ,$$

както и при метода на хордите, са изпълнени условията:

- (i)  $f(a)f(b) < 0$ ;
- (ii)  $f \in C^2[a, b]$ , т.е.  $f$  има непрекъсната втора производна в  $[a, b]$ ;
- (iii)  $f'$  и  $f''$  имат постоянен знак в  $[a, b]$ .

Тогава уравнението  $f(x) = 0$  има точно един корен  $\xi$  в интервала  $[a, b]$  и отново можем да разгледаме четири случая в зависимост от знаците на производните  $f'(x)$  и  $f''(x)$ .

Тук за начално приближение  $x_0$  избираме този от краищата на интервала  $[a, b]$ , в който имаме  $f f'' > 0$ , т.е.

- при  $f(a)f''(a) > 0$ , полагаме  $x_0 = a$ ,
- при  $f(b)f''(b) > 0$ , полагаме  $x_0 = b$ ,

като и в двата случая следващите последователни приближения пресмятаме по формулите

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \quad n = 1, 2, \dots$$

Може да се докаже, че така описаният *метод на Нютон* има ред на сходимост  $p = 2$  (квадратична сходимост). Този метод понякога се нарича и *метод на допирателните*. Последното наименование идва от геометричната интерпретация на последователните приближения  $x_n$ ,  $n = 1, 2, \dots$  – всяко  $x_n$  се явява абсциса на пресечната точка на допирателната към графиката на функцията  $f(x)$  в точката  $(x_{n-1}, f(x_{n-1}))$  с абсцисната ос.

**Пример 7.3.** С метода на Нютон да се намери приближено с точност  $\varepsilon = 0,001$  положителният корен на уравнението

$$(x + 1)^2 - \operatorname{arctg} x - 4 = 0.$$

Нека  $f(x) = (x + 1)^2 - \operatorname{arctg} x - 4$ . Както в Пример 7.2 локализираме корена  $\xi \in [a, b] = [1, 2]$ . За производните имаме

$$f'(x) = 2(x + 1) - \frac{1}{1 + x^2} > 0, \quad f''(x) = 2 + \frac{2x}{(1 + x^2)^2} > 0, \quad x \in [0, \infty).$$

функцията  $f(x)$  удовлетворява условията (i)-(iii) за интервала  $[1, 2]$ . Понеже  $f(b)f''(b) = f(2)f''(2) > 0$  избираме  $x_0 = 2$  и последователните приближения пресмятаме по формулите

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \quad n = 1, 2, \dots$$

Имаме

$$x_1 = 1,32882, \quad x_2 = 1,21296, \quad x_3 = 1,20902, \quad x_4 = 1,20901,$$

като  $|x_4 - x_3| = |1,20901 - 1,20902| = 0,00001 < 0,001$ . За приближение на корена избираме

$$\xi \approx x_4 = 1,20901.$$

Да отбележим, че с метода на Нютон бяха необходими само 4 итерации, за да получим приближение на корена с точност 0,00001 след локализирането му в интервала  $[a, b] = [1, 2]$  с дължина  $b - a = 1$ .  $\square$

## 5. Метод на простата итерация.

Нека е дадено уравнението

$$f(x) = 0, \quad (7.2)$$

което с преобразования довеждаме до еквивалентното уравнение

$$x = \varphi(x). \quad (7.3)$$

Тогава  $\xi$  е корен на (7.2) тогава и само тогава, когато  $\xi$  е корен на (7.3). Ще отбележим, че такава трансформация може да извършим по много начини. Например, ако  $g$  е обратима функция на една променлива и  $g^{-1}$  е нейната обратна, от (7.2) получаваме  $g(x) = f(x) + g(x)$ , откъдето  $x = g^{-1}(f(x) + g(x))$ , т.е. уравнение от вида (7.3).

Да дефинираме итерационен процес при дадено начално приближение  $x_0$  с рекурентните формули

$$x_n = \varphi(x_{n-1}), \quad n = 1, 2, \dots \quad (7.4)$$

Нека  $\varphi(x)$  е непрекъснатата функция и редицата от последователните приближения  $x_0, x_1, \dots, x_n, \dots$  е сходяща, с граница

$$\xi = \lim_{n \rightarrow \infty} x_n.$$

С граничен преход при  $n \rightarrow \infty$  в равенството (7.4) получаваме, че

$$\xi = \varphi(\xi),$$

т.е.  $\xi$  е корен на уравнението (7.3), а следователно и на еквивалентното му (7.2). В такъв случай, при зададена точност  $\varepsilon$  от някой номер на индекса нататък всички последователни приближения ще удовлетворяват неравенството

$$|\xi - x_n| < \varepsilon.$$

Описаният метод за решаване на дадено нелинейно уравнение се нарича *метод на простата итерация*, а функцията  $\varphi(x)$  – *итерираща функция*. Идеята на метода е приложима и към други задачи от областта на числените методи – приближено решаване на системи линейни уравнения, системи нелинейни уравнения и др.

Остана да изясним въпроса при какви условия итерационният процес (7.4) е сходящ. Следващата теорема дава едно достатъчно условие за това.

**Теорема 7.1.** Нека  $\varphi : [a, b] \rightarrow [a, b]$  и  $\varphi$  има непрекъсната производна в  $[a, b]$ . Нека  $q$  е число,  $0 < q < 1$ , такова че

$$|\varphi'(x)| \leq q \quad \text{за всяко } x \in [a, b].$$

Тогава итерационният процес (7.4) е сходящ към единствения корен на уравнението (7.3), при това за произволно начално приближение  $x_0$  от интервала  $[a, b]$ .

**Доказателство.** Да приложим Теоремата за крайните нараствания за функцията  $\varphi(x)$  в интервала  $[x_{n-1}, x_n]$  с краища две последователни приближения, получени по формулите (7.4). Тогава съществува точка  $\eta_{n-1} \in (x_{n-1}, x_n) \subset [a, b]$ , така че

$$x_n - x_{n-1} = \varphi(x_{n-1}) - \varphi(x_{n-2}) = (x_{n-1} - x_{n-2})\varphi'(\eta_{n-1}).$$

От неравенството

$$|x_n - x_{n-1}| = |x_{n-1} - x_{n-2}| |\varphi'(\eta_{n-1})| \leq |x_{n-1} - x_{n-2}| q,$$

приложено последователно получаваме

$$|x_n - x_{n-1}| \leq |x_{n-1} - x_{n-2}| q \leq |x_{n-2} - x_{n-3}| q^2 \leq \dots \leq |x_1 - x_0| q^n,$$

т.е.

$$|x_n - x_{n-1}| \leq |x_1 - x_0| q^n \leq (b - a) q^n, \quad n = 1, 2, \dots$$

Разглеждаме числовия ред

$$x_0 + (x_1 - x_0) + \dots + (x_n - x_{n-1}) + \dots$$

Той е сходящ, защото се мажорира от сумата на безкрайна геометрична прогресия

$$\sum_{n=0}^{\infty} (b - a) q^n$$

с частно  $q$ ,  $0 < q < 1$ . Следователно и редицата от парциалните му суми, а именно  $x_0, x_1, \dots, x_n, \dots$ , е сходяща към някакво число  $\xi$ . Както вече показахме, с граничен преход в (7.4) получаваме, че  $\xi$  е корен на уравнението (7.3). Единствеността на корена  $\xi$  се доказва лесно.  $\square$

**Пример 7.4.** Уравнението  $x + \ln x = 0$  има единствен корен  $\xi \in [\frac{1}{4}, \frac{3}{4}]$ . Дефинираме итерационни процеси с рекурентните формули:

$$1) x_n = -\ln x_{n-1}; \quad 2) x_n = e^{-x_{n-1}}; \quad 3) x_n = \frac{1}{2}(x_{n-1} + e^{-x_{n-1}}).$$



Коя от формулите трябва да използваме?

1) В този случай  $\varphi(x) = -\ln x$ ,  $\varphi'(x) = -\frac{1}{x}$ ,  $|\varphi'(x)| = \frac{1}{x} \geq \frac{4}{3} > 1$  за всяко  $x \in [\frac{1}{4}, \frac{3}{4}]$ . Следователно нищо не ни гарантира, че итерационният процес ще бъде сходящ.

2) Сега  $\varphi(x) = e^{-x}$ ,  $\varphi'(x) = -e^{-x}$ ,  $|\varphi'(x)| = e^{-x} \leq e^{-1/4} \approx 0,779 < 1$  за всяко  $x \in [\frac{1}{4}, \frac{3}{4}]$ . Следователно итерационният процес ще бъде сходящ при произволен избор на начално приближение от  $[\frac{1}{4}, \frac{3}{4}]$ .

3) Тук  $\varphi(x) = \frac{1}{2}(x + e^{-x})$ ,  $\varphi'(x) = \frac{1}{2}(1 - e^{-x})$ ,  $|\varphi'(x)| = \frac{1}{2}(1 - e^{-x}) \leq \frac{1}{2}(1 - e^{-3/4}) \approx 0,528 < 1$  за всяко  $x \in [\frac{1}{4}, \frac{3}{4}]$ . Итерационният процес ще бъде сходящ при произволен избор на начално приближение от  $[\frac{1}{4}, \frac{3}{4}]$ .

Можем да решим приближено с метода на простата итерация даденото уравнение, като използваме произволно начално приближение  $x_0 \in [\frac{1}{4}, \frac{3}{4}]$  както с формулите 2), така и с 3). Тъй като мажорантата на  $|\varphi'(x)|$  в последния случай е по-малка, използването на формулите 3) ще ни даде по-бърза сходимост на итерационния процес към корена на уравнението.  $\square$

**Пример 7.5.** С метода на простата итерация да се намери приближено с точност  $\varepsilon = 0,001$  положителният корен на уравнението

$$(x + 1)^2 - \operatorname{arctg} x - 4 = 0 .$$

Както в Пример 7.2 локализираме корена  $\xi \in [a, b] = [1, 2]$ . Да преобразуваме даденото уравнение в еквивалентното при  $x > 0$  уравнение

$$x = \sqrt{4 + \operatorname{arctg} x} - 1 .$$

Да положим

$$\varphi(x) = \sqrt{4 + \operatorname{arctg} x} - 1$$

и да разгледаме итерационния процес, дефиниран с равенствата

$$x_n = \varphi(x_{n-1}) = \sqrt{4 + \operatorname{arctg} x_{n-1}} - 1 , \quad n = 1, 2, \dots ,$$

където началното приближение  $x_0$  е произволно число от интервала  $[1, 2]$ . Пресмятаме производната

$$\varphi'(x) = \frac{1}{2(1+x^2)\sqrt{4+\operatorname{arctg} x}} .$$

Понеже множителите  $1 + x^2$  и  $\sqrt{4 + \arctg x}$  са растящи в  $[1, 2]$ , то  $\varphi'(x)$  е намаляваща в  $[1, 2]$ . Оттук

$$0 < \varphi'(x) \leq \varphi'(1) = \frac{1}{4\sqrt{4 + \pi/4}} \approx 0,114$$

и условието за итериращата функция от Теорема 7.1 е изпълнено. Имаме

$$|\varphi'(x)| < q = 0,12 < 1 .$$

Следователно итерационният процес е сходящ при произволно начално приближение  $x_0 \in [1, 2]$ .

Ето резултатите от изчисленията при начални приближения  $x_0 = 1$ ,  $x_0 = 1,5$  и  $x_0 = 2$ .

$x_0 = 1$	$x_0 = 1,5$	$x_0 = 2$
$x_1 = 1,18756$	$x_1 = 1,23222$	$x_1 = 1,2599$
$x_2 = 1,20702$	$x_2 = 1,21112$	$x_2 = 1,21357$
$x_3 = 1,20883$	$x_3 = 1,20921$	$x_3 = 1,20943$
$x_4 = 1,209$	$x_4 = 1,20903$	$x_4 = 1,20905$

□

## §8. ПРИБИЖЕНО РЕШАВАНЕ НА СИСТЕМИ НЕЛИНЕЙНИ УРАВНЕНИЯ

Да разгледаме система от две уравнения с две неизвестни  $x$  и  $y$

$$\begin{cases} f(x, y) = 0 , \\ g(x, y) = 0 , \end{cases} \quad (8.1)$$

където функциите  $f(x, y)$  и  $g(x, y)$  не са линейни.

Намирането на точно решение  $(\xi, \eta)$  на такава система може да ни се удаде само в някои много специални случаи. А какво да правим, когато не можем да намерим точно решение? Тогава да опитаме да намерим достатъчно добро приближено решение.

Ще изложим едно обобщение на метода на Нютон от предишния параграф за приближено решаване на системи от вида (8.1). Именно, при начално приближение  $(x_0, y_0)$ , избрано достатъчно близо до решението  $(\xi, \eta)$  на системата (8.1), ще построим редица от последователни

приближения  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \dots$ , така че

$$\lim_{n \rightarrow \infty} (x_n, y_n) = (\xi, \eta) .$$

Да предположим, че сме намерили  $(n-1)$ -то приближение  $(x_{n-1}, y_{n-1})$  и то е достатъчно близо до точното решение  $(\xi, \eta)$  на (8.1). Полагаме за удобство

$$\begin{aligned} f_{n-1} &= f(x_{n-1}, y_{n-1}) , & g_{n-1} &= g(x_{n-1}, y_{n-1}) , \\ \frac{\partial f_{n-1}}{\partial x} &= \frac{\partial f(x_{n-1}, y_{n-1})}{\partial x} , & \frac{\partial g_{n-1}}{\partial x} &= \frac{\partial g(x_{n-1}, y_{n-1})}{\partial x} , \\ \frac{\partial f_{n-1}}{\partial y} &= \frac{\partial f(x_{n-1}, y_{n-1})}{\partial y} , & \frac{\partial g_{n-1}}{\partial y} &= \frac{\partial g(x_{n-1}, y_{n-1})}{\partial y} \end{aligned}$$

и развиваме на функциите  $f(x, y)$  и  $g(x, y)$  около точката  $(x_{n-1}, y_{n-1})$  по формулата на Тейлор:

$$\left\{ \begin{aligned} f(x, y) &= f_{n-1} + \frac{\partial f_{n-1}}{\partial x} (x - x_{n-1}) + \frac{\partial f_{n-1}}{\partial y} (y - y_{n-1}) + R_{n-1}(f) , \\ g(x, y) &= g_{n-1} + \frac{\partial g_{n-1}}{\partial x} (x - x_{n-1}) + \frac{\partial g_{n-1}}{\partial y} (y - y_{n-1}) + R_{n-1}(g) . \end{aligned} \right.$$

Пренебрегваме остатъците  $R_{n-1}(f)$  и  $R_{n-1}(g)$  и от системата нелинейни уравнения (8.1) получаваме линейната система

$$\left\{ \begin{aligned} f_{n-1} + \frac{\partial f_{n-1}}{\partial x} (x - x_{n-1}) + \frac{\partial f_{n-1}}{\partial y} (y - y_{n-1}) &= 0 , \\ g_{n-1} + \frac{\partial g_{n-1}}{\partial x} (x - x_{n-1}) + \frac{\partial g_{n-1}}{\partial y} (y - y_{n-1}) &= 0 . \end{aligned} \right. \quad (8.2)$$

Избираме следващото приближение  $(x_n, y_n)$  на  $(\xi, \eta)$  да бъде решението на линейната система (8.2) и така за всяко  $n = 1, 2, \dots$ . По такъв начин приближеното решаване на една нелинейна система от вида (8.1) свеждаме до многократно решаване на линейна система от вида (8.2).

При зададена точност  $\varepsilon > 0$  изчисленията продължават докато две последователни приближения се окажат на разстояние, по-малко от  $\varepsilon$ . Ако използваме Евклидовото разстояние в равнината, условието ще има вида

$$\sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2} < \varepsilon .$$

Тогава за приближено решение на системата (8.1) с описания метод посочваме последното пресметнато приближение  $(x_n, y_n)$ , т.е.

$$(\xi, \eta) \approx (x_n, y_n) \iff \xi \approx x_n \text{ и } \eta \approx y_n .$$



решение  $\Xi$ , построяваме редица  $X^{(n)} = (x_1^{(n)}, x_2^{(n)}, \dots, x_k^{(n)})$ ,  $n = 1, 2, \dots$  така че

$$\lim_{n \rightarrow \infty} X^{(n)} = \Xi .$$

Ако означим

$$\Phi(X) = \begin{pmatrix} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \dots & \frac{\partial f_1(X)}{\partial x_k} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \dots & \frac{\partial f_2(X)}{\partial x_k} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_k(X)}{\partial x_1} & \frac{\partial f_k(X)}{\partial x_2} & \dots & \frac{\partial f_k(X)}{\partial x_k} \end{pmatrix}, \quad F(X) = \begin{pmatrix} f_1(X) \\ f_2(X) \\ \dots \\ f_k(X) \end{pmatrix},$$

$$\Phi_{n-1} = \Phi(X^{(n-1)}), \quad F_{n-1} = F(X^{(n-1)})$$

и  $\Phi_{n-1}^{-1}$  е обратната на матрицата  $\Phi_{n-1}$ , то получаваме

$$X^{(n)} = X^{(n-1)} - \Phi_{n-1}^{-1} \cdot F_{n-1}, \quad n = 1, 2, \dots \quad (8.5)$$

Въпросът със сходимостта на итерационния процес при метода на Нютон за нелинейни системи стои по-сложно. Няма общи и удобни за проверяване условия, които да ни гарантират сходимост на редицата от последователните приближения към решението на системата. На практика се процедурира по следния начин: програмата, реализираща метода, се стартира с някакво начално приближение и ако редицата от последователни приближения се окаже разходяща, сменя се началното приближение и програмата се стартира отново; продължава се в същия дух докато се попадне на подходящо начално приближение, за което процесът е сходящ. За добрия избор на начално приближение може да използваме средствата на математическия анализ, програма за визуализация на графики на функции на две променливи и други съображения.

**Пример 8.1.** С метода на Нютон да се намери приближено с точност  $\varepsilon = 0,001$  решението  $(\xi; \eta)$  на системата

$$\begin{cases} x^2 - 4x + y^2 = 0, \\ x^2 + y^2 + 6x - 2y - 6 = 0, \end{cases}$$

като за начално приближение е избрано  $X_0 = (x_0; y_0) = (1; 1)$ .

За  $X = (x; y)$  имаме

$$F(X) = F(x, y) = \begin{pmatrix} f(x, y) \\ g(x, y) \end{pmatrix} = \begin{pmatrix} x^2 - 4x + y^2 \\ x^2 + y^2 + 6x - 2y - 6 \end{pmatrix},$$

$$\Phi(X) = \Phi(x, y) = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} = \begin{pmatrix} 2x - 4 & 2y \\ 2x + 6 & 2y - 2 \end{pmatrix}.$$

При първата итерация намираме  $X_1$ :

$$F_0 = F(1, 1) = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \quad \Phi_0 = \Phi(1, 1) = \begin{pmatrix} -2 & 2 \\ 8 & 0 \end{pmatrix}, \quad \Phi_0^{-1} = \begin{pmatrix} 0 & 0,125 \\ 0,5 & 0,125 \end{pmatrix},$$

$$X_1 = X_0 - \Phi_0^{-1} \cdot F_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 & 0,125 \\ 0,5 & 0,125 \end{pmatrix} \cdot \begin{pmatrix} -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Аналогично за  $X_2$  имаме:

$$F_1 = F(1, 2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \Phi_1 = \Phi(1, 2) = \begin{pmatrix} -2 & 4 \\ 8 & 2 \end{pmatrix}, \quad \Phi_1^{-1} = \begin{pmatrix} -0,056 & 0,111 \\ 0,222 & 0,056 \end{pmatrix},$$

$$X_2 = X_1 - \Phi_1^{-1} \cdot F_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} -0,056 & 0,111 \\ 0,222 & 0,056 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0,944 \\ 1,722 \end{pmatrix};$$

за  $X_3$ :

$$F_2 = F(0,944, 1,722) = \begin{pmatrix} 0,080 \\ 0,080 \end{pmatrix},$$

$$\Phi_2 = \Phi(0,944, 1,722) = \begin{pmatrix} -2,111 & 3,444 \\ 7,889 & 1,444 \end{pmatrix}, \quad \Phi_2^{-1} = \begin{pmatrix} -0,048 & 0,114 \\ 0,261 & 0,070 \end{pmatrix},$$

$$X_3 = X_2 - \Phi_2^{-1} \cdot F_2 = \begin{pmatrix} 0,944 \\ 1,722 \end{pmatrix} - \begin{pmatrix} -0,048 & 0,114 \\ 0,261 & 0,070 \end{pmatrix} \cdot \begin{pmatrix} 0,080 \\ 0,080 \end{pmatrix} = \begin{pmatrix} 0,940 \\ 1,696 \end{pmatrix};$$

за  $X_4$ :

$$F_3 = F(0,940, 1,696) = \begin{pmatrix} 0,001 \\ 0,001 \end{pmatrix},$$

$$\Phi_3 = \Phi(0,944, 1,722) = \begin{pmatrix} -2,122 & 3,391 \\ 7,878 & 1,391 \end{pmatrix}, \quad \Phi_3^{-1} = \begin{pmatrix} -0,047 & 0,114 \\ 0,266 & 0,072 \end{pmatrix},$$

$$X_4 = X_3 - \Phi_3^{-1} \cdot F_3 = \begin{pmatrix} 0,940 \\ 1,696 \end{pmatrix} - \begin{pmatrix} -0,047 & 0,114 \\ 0,266 & 0,072 \end{pmatrix} \cdot \begin{pmatrix} 0,001 \\ 0,001 \end{pmatrix} = \begin{pmatrix} 0,939 \\ 1,695 \end{pmatrix}.$$

Ако с  $\|X - Y\|$  означим евклидовото разстояние между точките  $X$  и  $Y$  в равнината, за разстоянията между последователните приближения имаме:  $\|X_1 - X_0\| = 1$ ,  $\|X_2 - X_1\| = 0,283$ ,  $\|X_3 - X_2\| = 0,027$ ,  $\|X_4 - X_3\| = 0,0003$ . Следователно точност  $\varepsilon = 0,001$  достигаме след четвъртата итерация и можем да приемем, че  $(\xi; \eta) \approx X_4$ , т.е.

$$(\xi; \eta) \approx (0,939; 1,695) .$$

□