

ՀՏԴ 62-501.7

Ինֆորմատիկա

Վոլոդյա Մանասյան

ԱրՊՀ կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի դոցենտ

E-mail manasyan@mail.ru

Անուշ Հարությունյան

ԱրՊՀ կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի դասախոս

E-mail asrian.anush@mail.ru

ՖՈՒՆԿՑԻԱՅԻ ՄԻՆԻՄԱԼԱՅՆՈՂ ԿԵՏԻ ՈՐՈՇՄԱՆ ՃՇՏՈՒԹՅԱՆ ԲԱՐՁՐԱՑՄԱՆ ՄԱՍԻՆ

Սույն հոդվածում փորձ է կատարվում վերլուծել մեկ փոփոխականի ֆունկցիայի մինիմացնող կետի դիրքի ճշգրիտ որոշման հնարավորությունը:

Բանալի բառեր` ֆունկցիա, մինիմիզացնող կետ, գործողությունների սրան, սխալի նվազեցում, ֆիբոնաչչու թվեր, հաշվման ալգորիթմ, ծրագիր:

В. Манасян. А.Арутюнян

О ПОВЫШЕНИИ ТОЧНОСТИ ОПРЕДЕЛЕНИЯ МИНИМИЗИРУЮЩЕЙ ТОЧКИ ФУНКЦИИ

Содержание данной статьи состоит в том, чтобы проанализировать возможности точного определения положения минимизирующей точки функции с одним минимумом, т.е. в уменьшении ошибки, с которой указывается эта точка.

Ключевые слова: функция, минимизирующие точка, план действий, уменьшение ошибки, числа фибоначчи, алгоритм вычислений, программа.

V.Manasyan. A.Haroutyunyan

ON IMPROVING THE ACCURACY OF DETERMINING THE MINIMIZING POINT OF A FUNCTION

The content of this article is to analyze the possibilities of accurately determining the position of the minimizing point of a function with one minimum, i.e. in reducing the error with which this point is indicated.

Key words: function, minimizing point, action plan, error reduction, fibonacci numbers, calculation algorithm, program.

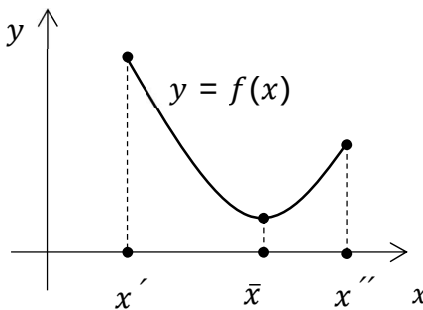
Ինչպես հայտնի է, խնդիրների շատ լայն դաս են կազմում էքստրեմալ խնդիրները, որոնց մեջ պահանջվում է գտնել ֆունկցիաների արժեքները, որոնք ռեալիզացնում են մաքսիմում կամ մինիմում:

Շատ խնդիրների ճշգրիտ լուծումների մոտ լուծումներ գտնելու համար կարելի է մոտարկել ընտրելով ֆունկցիաների անհայտ արժեքները այնպես, որ ստացված սխալը լինի մինիմալ:

Մեկ փոփոխականի մաքսիմումի կամ մինիմումի գտնելու հարցերը բերված են մաթեմատիկական անալիզի տարբեր դասագրքերում և մաթեմատիկայի տարբեր տեղեկատվություններում: Մասնավորապես, [1]-ում բերված են աքսիոմներ լոկալ (ներքին) մաքսիմումի և մինիմումի մասին: Ավանդական եղանակով էքստրեմումները որոշվում են ածանցյալների միջոցով: Բայց միշտ չէ, որ ածանցյալ հաշվելը հարմար է:

Ենթադրենք $f(x)$ ֆունկցիայի մասին հայտնի է, որ այն տված x' կետից մինչև անհայտ $\bar{x}$ կետը նվազում է, իսկ $\bar{x}$ -ից մինչև տված x'' կետը աճում է (նկ.1):

Պահանջվում է վերլուծել f ֆունկցիայի մինիմալացնող կետի դիրքի ճիշտ որոնման հնարավորությունը:



նկ.1

Այս խնդրի նպատակն է մինիմալացնող կետի որոշման ճշտության բարձրացումը, այսինքն՝ սխալի նվազեցումը, ինչով տրված է այս կետը:

Դա հնարավոր կդառնա այս կամ այն ճանապարհով ճիշտ որոշել կամայական ձևով ընտրված

կետերում f ֆունկցիայի արժեքները և համեմատել իրար հետ, երբ խնդրի պայմանները տրվում են f ֆունկցիայի որոշման տիրույթով, այսինքն՝ x' և x'' կետերի միջև L երկարությամբ հատվածով:

Մեզ անհրաժեշտ է գործողությունների P պլան, որը n դիտարկումներից հետո անվերապահորեն կբերի $\bar{x}$ կետի որոշմանը L երկարությամբ հատվածի վրա տված $\varepsilon > 0$ ճշտությամբ: Այդ ճշտությունը, ակնհայտ է, կախված կլինի P -ից, n -ից և L -ից: Այն կնշանակենք՝ $\tau_P(n, L)$: Կհամարենք P_0 պլանը f -ի մինիմումը որոշելու համար L հատվածի վրա օպտիմալ, եթե ցանկացած ուրիշ P պլանի համար

$$\tau_{P_0}(n, L) \leq \tau_P(n, L):$$

Այն կարելի է գրել նաև հետևյալ կերպ՝ $\tau_{P_0}(n, L) = \min_P \tau_P(n, L)$,

որը հանդիսանում է խնդրի բնութագրիչը: Քանի որ այն կախված չէ P -ից, ապա խնդրի համար նպատակահարմար է հետազոտել $\tau(n, L)$ -ի վարքը:

Իրավացի է հետևյալ լեմման [1, 318-319].

Լեմմա. Ինչպիսին էլ լինեն $n \geq 1$ և L թվերը, գոյություն ունի $\bar{x}$ կետի որոնման n քայլանի (կամ n դիտարկումներով) պլան, որը մինիմալացնում է f ֆունկցիայի արժեքը L երկարությամբ հատվածի վրա n քայլերի ընթացքում և օժտված է հետևյալ հատկություններով՝

1)յուրաքանչյուր քայլում դիտարկվում է ինչ-որ x' և x'' հատված,

2)առաջին քայլում հաշվվում է f ֆունկցիայի արժեքը $\frac{U_n}{U_{n+2}}L$ կամ $\frac{U_{n+1}}{U_{n+2}}L$ կետերից որևէ մեկում, որտեղ U_n, U_{n+1}, U_{n+2} -երը հանդիսանում են ֆիբոնաչչու $(1, 1, 2, 3, 5, 8, 13, \dots)$ հաջորդականության համապատասխանաբար $n -$ րդ, $n + 1 -$ րդ, $n + 2 -$ րդ անդամներն են ($U_1 = U_2 = 1, n = 1, 2, 3, \dots$)[2, 87-88],

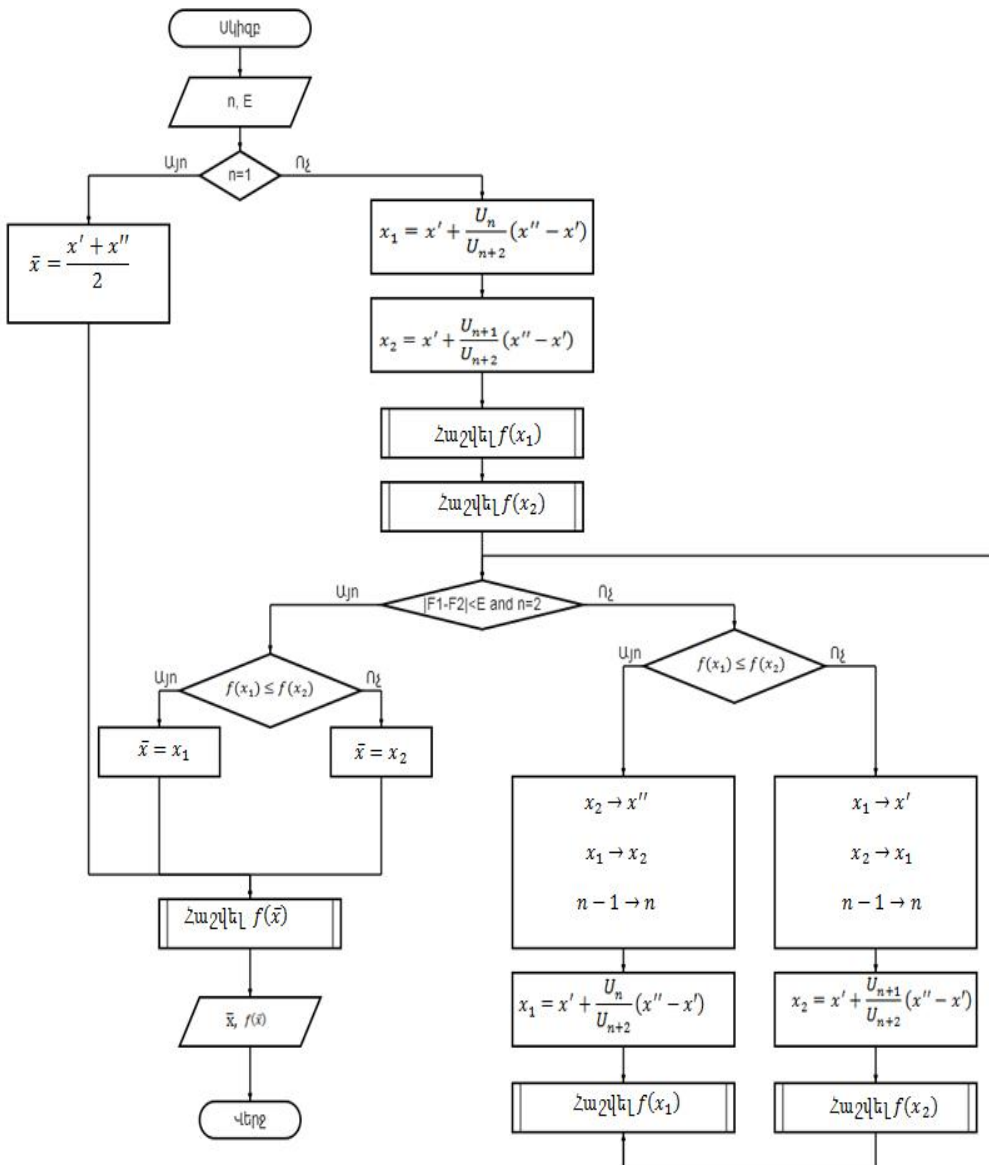
3) $1 < k \leq n$ դեպքում հայտնի են f -ի արժեքը հետևյալ կետերից որևէ մեկում՝

$$x_1 = x' + \frac{U_n}{U_{n+2}}(x'' - x') \quad \text{և} \quad x_2 = x' + \frac{U_{n+1}}{U_{n+2}}(x'' - x'),$$

4) $k -$ րդ քայլում հաշվվում է մյուս կետում ֆունկցիայի արժեքը,

5) $k -$ րդ քայլում կատարվում է $f(x_1)$ և $f(x_2)$ արժեքների համեմատում, եթե $f(x_1) \leq f(x_2)$, ապա $(k + 1) -$ րդ քայլում դիտարկվում է x'_2 հատվածը, իսկ եթե $f(x_1) \geq f(x_2)$, ապա $x_1 x''$ հատվածը:

Որպեսզի համոզվենք, որ $f -$ ը մինիմիզացվող կետը որոշված է L երկարությամբ հատվածի վրա ε -ը չգերազանցող սխալով, անհրաժեշտ է կատարել այնքան n դիտարկումներ, որ $U_{n+1} < \frac{L}{\varepsilon} \leq U_{n+2}$: $x' x''$ հատվածում $\bar{x}$ կետի որոնումը, որը մինիմիզացնում է f ֆունկցիան նկարագրվում է հետևյալ ալգորիթմով (նկ. 2)[3]:



Նկ.2

Օրինակ Եթե տված է $f(x) = \frac{1}{x} + \sqrt{x}$ ֆունկցիան $[1,2]$ հատվածում, ապա

$$f'(x) = -\frac{1}{x^2} + \frac{1}{2\sqrt{x}} = 0$$

$$x^2 = 2\sqrt{x}$$

$$\sqrt{x}(x\sqrt{x} - 2) = 0$$

$$\sqrt{x} = 0$$

$$x = 0 \notin [1, 2]$$

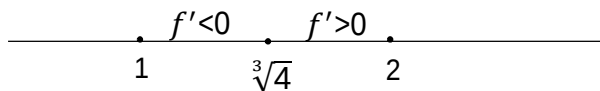
$$x\sqrt{x} - 2 = 0$$

$$x\sqrt{x} = 2$$

$$\sqrt{x^3} = 2$$

$$x^3 = 4$$

$$x = \sqrt[3]{4}$$



$$\sqrt[3]{4} = 1.5874011, \quad \bar{x} = \sqrt[3]{4} \text{ մինիմումի կետ է:}$$

Եթե ֆունկցիան հայտնի է, և չենք օգտագործում ածանցյալը, ապա օգտվում ենք այդ ալգորիթմից՝ ընդունելով $n = 5$ (քայլեր).

Քայլ 1. $n = 5 \neq 1$ անցում Քայլ 4 -ին

Քայլ 2. Հաշվում ենք

$$x_1 = x' + \frac{U_n}{U_{n+2}}(x'' - x') = 1 + \frac{5}{13}(2 - 1) = 1.38461$$

$$x_2 = x' + \frac{U_{n+1}}{U_{n+2}}(x'' - x') = 1 + \frac{8}{13}(2 - 1) = 1.61538$$

Քայլ 3. Հաշվում ենք

$$f(x_1) = \frac{1}{x_1} + \sqrt{x_1} = f(1.38461) = 0.72222 + 1.17670 = 1.89892$$

$$f(x_2) = \frac{1}{x_2} + \sqrt{x_2} = f(1.61538) = 0.61905 + 1.27098 = 1.89003$$

Քայլ 4. $n = 5 \neq 2$

Քայլ 5. $f(x_1) = 1.89892$ և $f(x_2) = 1.89003$, համեմատումը կտա $f(x_1) > f(x_2)$:

Քայլ 6. Կատարենք նշանակումներ

$$x_1 \rightarrow x' = 1.38461$$

$$x_2 \rightarrow x_1 = 1.61538$$

$$n = 4$$

Քայլ 7. Հաշվում ենք

$$x_2 = x' + \frac{U_{n+1}}{U_{n+2}}(x'' - x') = 1.38461 + \frac{5}{8}(2 - 1.38461) = 1.76927$$

Քայլ 8. Հաշվում ենք

$$f(x_2) = \frac{1}{x_2} + \sqrt{x_2} = f(1.76927) = 0.56522 + 1.33012 = 1.89534$$

Քայլ 9. $n = 4 \neq 3$

Քայլ 10. Համեմատում ենք $f(x_1) = 1.89003$ և $f(x_2) = 1.89534$ -ը՝ $f(x_1) \leq f(x_2)$

Քայլ 11. Կատարենք նշանակումներ

$$x_2 \rightarrow x'' = 1.76923$$

$$x_1 \rightarrow x_2 = 1.61538$$

$$n = 3$$

Քայլ 12. Հաշվում ենք

$$x_1 = x' + \frac{U_n}{U_{n+2}}(x'' - x') = 1.38461 + \frac{2}{5}(1.76923 - 1.38461) = 1.53846$$

Քայլ 13. Հաշվում ենք

$$f(x_1) = \frac{1}{x_1} + \sqrt{x_1} = f(1.53846) = 0.65000 + 1.24035 = 1.89035$$

Քայլ 14. $n = 3 \neq 2$

Քայլ 15. Համեմատում ենք $f(x_1) = 1.89035$ և $f(x_2) = 1.89003$ -ը՝ $f(x_1) > f(x_2)$

Քայլ 16. Կատարենք նշանակումներ

$$x_2 \rightarrow x' = 1.53846$$

$$x_2 \rightarrow x_1 = 1.61538$$

$$n = 2$$

Քայլ 17. Հաշվում ենք

$$x_2 = x' + \frac{U_{n+1}}{U_{n+2}}(x'' - x') = 1.53846 + \frac{2}{3}(1.76923 - 1.53846) = 1.69231$$

Քայլ 18. Հաշվում ենք

$$f(x_2) = \frac{1}{x_2} + \sqrt{x_2} = f(1.69231) = 0.59091 + 1.30089 = 1.89170$$

Քայլ 19. $n = 2$

Քայլ 20. Համեմատում ենք $f(x_1) = 1.89003$ և $f(x_2) = 1.89170$ -ը՝ $f(x_1) \leq f(x_2)$

Քայլ 21. Ընդունում ենք $\bar{x} = 1.61538$

$$f_{min} = f(\bar{x}) = 1.89003$$

արմատների համար

$$\bar{x} - \sqrt[3]{4} = 1.61538 - 1.587401 = 0.028$$

$$f_{min} = f(\sqrt[3]{4}) = \frac{1}{\sqrt[3]{4}} + \sqrt[3]{2} = \frac{3}{2}\sqrt[3]{2} = 1.88988:$$

Տարբերությունը $1.89003 - 1.88988 = 0.00015$ է:

Նշենք, որ ավելի լայն ձևով իրացվում է ծրագրավորման մեջ: Ծրագիրը գրված է C++ ծրագրավորման լեզվով և արտածում է քայլ առ քայլ մինիմալացված միջակայքերը և մոտավոր լուծումը ε -ի ճշտությամբ:

```
//minimizacum.cpp
```

```
#include "pch.h"
#include <iostream>
#include<iomanip>
#include<cmath>
```

```
using namespace std;
```

```
double func(double x)
```

```
{
return 1. / x + sqrt(x);
}

int fibonachi(int u) {
if (u == 2 || u == 1)
return 1;
else
return fibonachi(u - 1) + fibonachi(u - 2);
}

double X1(double a, double b,int n)
{
return a + double(fibonachi(n)) / double(fibonachi(n + 2))*(b -
a);
}

double X2(double a, double b,int n)
{
return a + double(fibonachi(n + 1)) / double(fibonachi(n +
2))*(b - a);
}

int main()
{
int n = 5;
double x22, x11, x1 = 1, x2 = 2, F1, F2, _x, epsilon=0.001;
if (n != 1)
{
x11 = X1(x1, x2,n);
x22 = X2(x1, x2,n);
F1 = func(x1);
F2 = func(x2);
cout << "x1=" << x11 << setw(20) << "F1=" << F1 << endl;
cout << "x2=" << x22 << setw(20) << "F2=" << F2 << endl <<
endl;
while ((fabs(F1 - F2) < epsilon) && (n != 2))
{
n--;
if (F1 < F2)
{
x2 = x22;
x22 = x11;
x11 = X1(x1, x2, n);
```

```
F1 = func(x11);
}
else
{
x1 = x11;
x11 = x22;
x22 = X2(x1, x2, n);
F2 = func(x22);
}
cout << "x1=" << x11 << setw(20) << "F1=" << F1 << endl;
cout << "x2=" << x22 << setw(20) << "F2=" << F2 << endl <<
endl;
}
if (F1 < F2)
{
_x = x1;
}
else
{
_x = x2;
}
else
{
_x = (x1 + x2) / 2;
F1 = func(x1);
}
cout << "x=" << _x << endl;
cout << "Fmin(" << _x << ")=" << F1 << endl;
cout << "E=" << fabs(F1 - F2) << endl;
return 0;
}
```



```

Консоль отладки Microsoft Visual Studio
x1=1          F1=2
x2=2          F2=1.91421

x1=1.38462    F1=1.89892
x2=1.61538    F2=1.89003

x1=1.38462    F1=1.89892
x2=1.76923    F2=1.89534

x1=1.53846    F1=1.89035
x2=1.76923    F2=1.89534

x1=1.61538    F1=1.89003
x2=1.76923    F2=1.89534

x1=1.61538    F1=1.89003
x2=1.69231    F2=1.8918

x=1.61538
Fmin(1.61538)=1.89003
E=0.00177092

```

Ակնհայտ է, որ նշված մեթոդը կիրառելի է այն դեպքում, երբ նախօրոք ոչինչ հայտնի չէ ֆունկցիայի մասին կամ էլ ֆունկցիան տրված է, չափազանց հաջող է, դիֆերենցիալ հավասարման մեթոդը կիրառելի չէ:

Գրականություն

1. Корн Г. и Корн Т., Справочник по математике для научных работников и инженеров. Изд. Наука М. 1968, стр-318-319
2. Воробьев Н.Н., Числа Фибоначчи изд. <<Наука>> М., 1969 стр. 87-88, 103-109
3. <https://programforyou.ru/block-diagram-redactor>

Հոդվածը տպագրության է երաշխավորել խմբագրական խորհրդի անդամ, ֆ.մ.գ.թ. Գ.Հ.Սահակյանը: