ALGEBRAIK KO'PHADLAR UCHUN LOBACHEVSKIY USULI TENGLAMALARNI TAQRIBIY ILDIZLARINI ANIQLASHDA DASTURLASH TILLARIDAN FOYDALANISH

[1]Musurmonqulova M.M., [2]Jomg'irova N.A

[1,2]Termiz davlat universiteti Kompyuter ilmlari va dasturlash texnologiyalari ta'lim yo'nalishi 2-kurs talabalari

Annotatsiya

Ushbu ilmiy maqolada algebraik tenglamalarning taqribiy ildizlarini aniqlashda qo'llaniladigan **Lobachevskiy usuli**ning mohiyati va amaliy qo'llanilishi tahlil qilinadi. Asosiy e'tibor bu usulning algoritmini ishlab chiqish va uni zamonaviy dasturlash tillari, xususan **Python, C++ yoki MATLAB** asosida dasturlash orqali kompyuterda real masalalarga qo'llash imkoniyatlariga qaratilgan. Ish davomida turli darajadagi ko'phadlar uchun tajriba o'tkazilib, natijalar grafik tarzda aks ettiriladi. Shuningdek, Lobachevskiy usulining aniqlik darajasi, konvergentsiya shartlari va boshqa sonli metodlar bilan taqqoslanadi. Tadqiqot natijalari talabalar, dasturchilar va matematik modelchilar uchun algebraik tenglamalarni samarali yechishda foydali bo'lishi mumkin.

Abstract

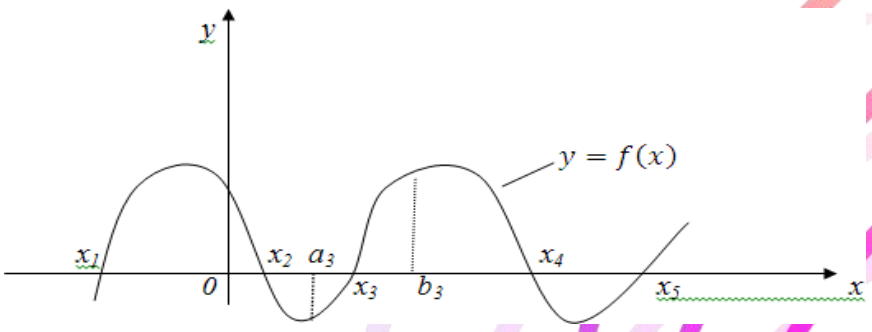
This scientific article analyzes the essence and practical application of the Lobachevsky method used to determine the approximate roots of algebraic equations. The main attention is paid to the development of an algorithm for this method and its application to real problems on a computer using modern programming languages, in particular, Python, C++ or MATLAB. During the work, experiments are conducted for polynomials of different degrees, and the results are displayed graphically. Also, the accuracy of the Lobachevsky method, convergence conditions, and comparison with other numerical methods are discussed. The results of the research can be useful for students, programmers, and mathematical modelers in effectively solving algebraic equations.

Аннотация

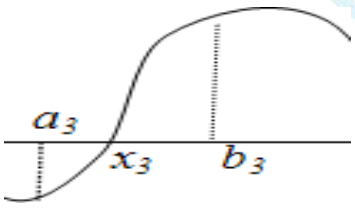
В данной научной статье анализируется сущность и практическое применение метода Лобачевского, который используется для определения приближенных корней алгебраических уравнений. Основное внимание уделяется разработке алгоритма для этого метода и его применению к решению

реальных задач на компьютере посредством программирования на современных языках программирования, в частности Python, C++ или MATLAB. В ходе работы проводятся эксперименты для полиномов разных степеней, а результаты отображаются графически. Обсуждаются также уровень точности, условия сходимости и сравнение метода Лобачевского с другими численными методами. Результаты исследования могут быть полезны студентам, программистам и специалистам по математическому моделированию для эффективного решения алгебраических уравнений.

Tenglamaning ildizi mavjudligi sharti, agar biror [*a,b*] oraliqda *y = f(x)* funksiya uzluksiz bo'lib, *f(a) · f(b) < 0* bo'lsa, shu oraliqda *f(x)=0* tenglamaning kamida bitta ildizi mavjud bo'ladi.



Agar biror [*a,b*] oraliqda *y=f(x)* funksiya uzluksiz bo'lib, birinchi tartibli uzluksiz xosilaga ega bo'lsa va *f(a) · f(b) < 0* , *f ' (x)* ([*a,b*] da ishorasi o'zgarmasa) shartlar bajarilsa, *f(x)=0* tenglama shu oraliqda yagona xaqiqiy ildizga ega bo'ladi.



Lobachevskiy usuli boyicha $P_n(x)=0$ tenglamaning taqribiy ildizlarini topish:

Bizga quyidagi tenglama berilgan

$P_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n = 0$ Va

Bu

tenglamanning n ta ildizi mavjud bo'lsa. Unung taqribiy ildizlarini **Lobachevskiy usuli** bo'yicha quyidagi **algoritm** bilan topish mumkin:

1) Quyidagi ifoda yordamida uning koeffitsentlarini hisoblash formulasini aniqlash:

$$a_k^s = (-1)^k \left(\left(a_k^{(S-1)}\right)^2 + \sum_{i=1}^{k} (-1)^i 2 a_{k-i}^{(S-1)} a_{k+i}^{(S-1)} \right)$$

$(S - usul\ qadami, k = 0,1,2,\dots,n.)$

Odatda, yuqoridagi ifodadan $a_k^{(S)}$larni hisoblash formulasi quyidagi ko'rinishda davom etadi:

$$a_0^{(s)} = \left(a_0^{(S-1)}\right)^2$$

$$a_1^{(s)} = -\left(a_1^{(S-1)}\right)^2 + 2a_0^{(S-1)} a_2^{(S-1)};$$

$$a_2^{(s)} = \left(a_2^{(S-1)}\right)^2 - 2a_1^{(S-1)} a_3^{(S-1)} + 2a_0^{(S-1)} 2a_4^{(S-1)};$$

$$a_n^{(S)} = \dots$$

2) Har doim S qadam 1 raqamidan boshlanadi. 0-qadamda (S=0 bo'lganda $a_k^{(0)}$lar sifatida berilgan tenglamaning koeffitsentlarini o'zini olamiz:

$$a_k^{(0)} = a_k \quad (k = 0,1,2,3,\dots,n)$$

3) S ning keyingi qadami uchun yuqorida aniqlangan $a_k^{(S)}$ larni hisoblash formulasi yordamida lardan foydalanib, ularning qiymatlarini hisoblaymiz.

4) Topilgan yangi koeffitsentlardan foydalanib, tenglamaning ildizlarini topishda ishlatiladigan quyidagi umumiy formula yordamida

$$a_k^{(S)} = \sqrt[2^S]{-a_k^{(S)} / a_{k-1}^{(S)}}, \quad k = 1,2,3,\dots,n.$$

unung taqribiy ildizlarini

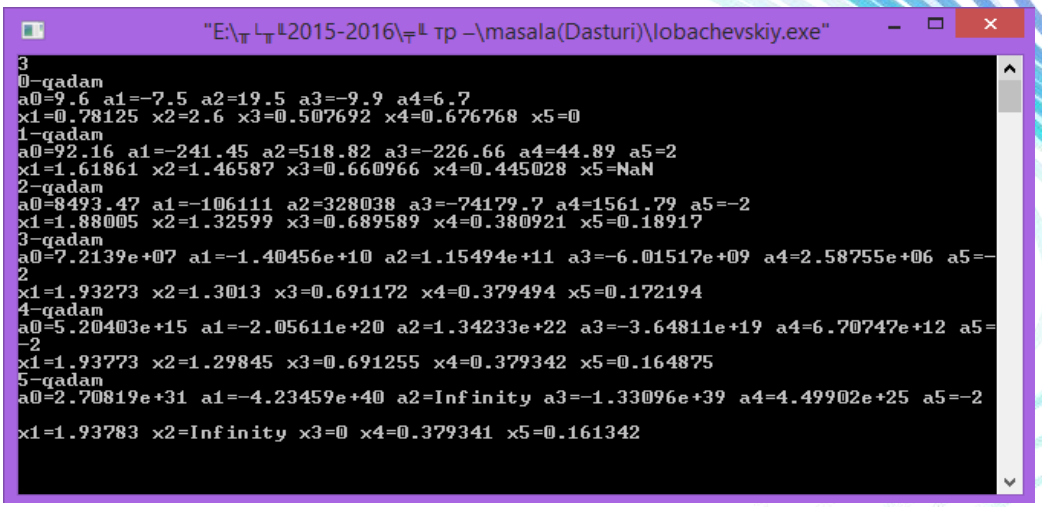
topamiz;

1) Shu zaylda 3 va 4-bandlarni bir necha bor takroran bajarib, taqribiy ildizlarning yaqinlashish dinamikasini ko'ramiz.

2) nihoyat, topilgan natijalarni tahlil qilib, tenglamaning taqribiy ildizlarini aniqlaymiz. Ko'phadlarni aniqlashda dastur kodidan foydalanish.

```
#include<iostrea
m>
#include<math.h
> using
namespace std;

double a[101][101],x[101][101];

double f(int s,int k){

        float s1=0;

    for(int i=1;i<=k;i++)s1+=pow(-1,i)*2.0+a[s-1][k-i]*a[s-1][k+i]; //
koeffitsiyentlar yig'indisini hisoblab olish

        return s1;

  }

int
main(){
float n;

    cin>>" TALABANING TARTIB RAQAMI ">>n;

    a[0][0]=n*3.2; //
    koeffitsiyentlarni hisoblash
    a[0][1]=-n*2.5;

    a[0][2]=n*6.5;

    a[0][3]=-n*3.3;

    a[0][4]=6.7;

    cout<<0<<"-
      qadam"<<endl;
      for(int
      k=0;k<5;k++){

            cout<<"a"<<k<<'='<<a[0][k]<<' ';

      }

      cout<<endl;

      for(int k=1;k<6;k++){
```

```
            x[0][k]=pow(-a[0][k]*1./a[0][k-1],1.0/pow(2,0));// hisoblash formulasi
            cout<<"x"<<k<<'='<<x[0][k]<<' ';
        }
        cout<<en
    dl;  for(int
    s=1;s<6;s+
    +){
          cout<<s<<"-
        qadam"<<endl; for(int
        k=0;k<6;k++){
            a[s][k]=pow(-1,k)*(pow(a[s-
            1][k],2)+f(s,k));
            cout<<"a"<<k<<'=';
            cout<<a[s][k]<<' ';
        }
        cout<<endl;
        for(int k=1;k<6;k++){
            x[s][k]=pow(-a[s][k]*1./a[s][k-1],1.0/pow(2,s));
            cout<<"x"<<k<<'='<<x[s][k]<<' ';
        }
        cout<<endl;
    }
cout<<"\n\n\n Begimov O'ktam Ibrogimovich ";
}
```



Xulosa qilib shuni aytish mumkinki, sonli usullar va dasturlash fanlarining barcha mavzulari bo'yicha taqribiy ildizlarini aniqlashda dasturlash tillarining o'rni bo'lak xisoblanadi. Boshqacha qilib aytganda dasturlash tillari yordamida juda kichik bo'lgan xatoliklarni aniqlashning imkoniyati mavjud. Bu mavzumizda Lobachevskiy usuli algebraik ko'phadlarni taqribiy qiymatlarini aniqlashda Code:Blocks dasturlash tilidan foydalandek. Demak, talabalar taqribiy qiymatlarni xisoblashda dasturlash tillarini mukammal bilishlari ta'lab qilinadi, shu asosda darslarni olib borish ham vaqtdan ham unumdorligi bilan ajralib turadi.

Foydalanilgan adabiyotlar:

1.Atkinson, K. E. (1989). *An Introduction to Numerical Analysis*. 2nd ed., Wiley.
2.Burden, R. L., & Faires, J. D. (2010). *Numerical Analysis*. 9th ed., Brooks Cole.
3.Gerald, C. F., & Wheatley, P. O. (2004). *Applied Numerical Analysis*. Pearson.
4.Tikhonov, A. N., & Samarskii, A. A. (1977). *Equations of Mathematical Physics*. Dover Publications.
5.Toshpulatov A. Sh. (2020). *Raqamli usullar va ularni dasturlash asoslari*. Toshkent: TDPU nashriyoti.
6.Zafarov O. A. (2018). *Numerik metodlar va dasturlash*. Samarqand: SamDU nashriyoti.
7.Lobachevskiy, N. I. (1834). *Imaginary Roots of Algebraic Equations*. (Asar tarjimasi va tahlil qilingan versiyalari)
8.Python.org Documentation.URL: https://docs.python.org/
9.NumPy & SciPy Documentation
URL: https://numpy.org/doc/, https://scipy.org/docs.html