GAUSS-SEIDEL VA YAKOBI ITERATSION USULLARI ASOSIDA HISOBLASHLAR

Ismoilov Axrorjon Ikromjonovich

Farg'ona Davlat Universiteti amaliy matematika va informatika kafedrasi katta o'qituvchisi

Email: ismoilovaxrorjon@yandex.com

Ismoilov Javohir Ulug'bek o`g`li

Farg'ona Davlat Universiteti "Kompyuter ilmlari va dasturlash texnologiyalari" yo'nalishi 23.12-guruh 2-bosqich talabasi

Email: javohir20060612@gmail.com

Habibullayev Javohir Odilbek o'g'li

Farg'ona Davlat Universiteti "Kompyuter ilmlari va dasturlash texnologiyalari" yo'nalishi 23.12-guruh 2-bosqich talabasi

Email: habibullayevjavohirbek5@gmail.com

Annotatsiya

Mazkur maqolada chiziqli algebraik tenglamalar sistemasini iteratsion usullar orqali yechish masalasi yoritilgan. Xususan, Yakobi va Gauss-Seidel usullarining algoritmlari, konvergentlik shartlari, ularning o'zaro farqlari hamda amaliy misollar asosida taqqoslanishi ko'rib chiqilgan. Shuningdek, bu usullar kompyuter hisoblashlarida qanday qo'llanilishi va ularning afzalliklari haqida fikr yuritilgan. Matritsali ifodalarni bosqichma-bosqich iteratsiya orqali yechish imkoniyatlari real misollar bilan tahlil qilinadi.

Kalit so'zlar: Iteratsion usullar, Yakobi usuli, Gauss-Seidel usuli, konvergentlik, chiziqli tenglamalar

Аннотация

В данной статье рассматривается задача решения системы линейных алгебраических уравнений с использованием итерационных методов. В частности, исследуются алгоритмы методов Якоби и Гаусса-Зейделя, условия сходимости, их различия и сравнение на основе практических примеров. Также обсуждается применение этих методов в компьютерных вычислениях и их преимущества. Возможности пошагового итерационного решения матричных выражений анализируются на конкретных примерах.

Ключевые слова: итерационные методы, метод Якоби, метод Гаусса-Зейделя, сходимость, линейные уравнения

Annotation

This article discusses the problem of solving systems of linear algebraic equations using iterative methods. In particular, it examines the algorithms of the Jacobi and Gauss-Seidel methods, their convergence conditions, mutual differences, and comparison based on practical examples. The article also explores how these methods are applied in computer computations and highlights their advantages. The step-by-step iterative solution of matrix expressions is analyzed through real-life examples.

Keywords: iterative methods, Jacobi method, Gauss-Seidel method, convergence, linear equations

Kirish

Zamonaviy ilmiy hisoblashlarning asosiy yo‘nalishlaridan biri bu – murakkab chiziqli algebraik tenglamalar sistemasini samarali yechish masalasidir. Bunday sistemalar muhandislik, fizika, iqtisodiyot va boshqa sohalarda keng uchraydi. Katta o‘lchamli sistemalarni bevosita (an’anaviy) usullar orqali yechish ko‘p hisoblash resurslarini talab qiladi. Shu sababli, iteratsion yondashuvlar – ayniqsa Yakobi va Gauss-Seidel usullari – bu sohada katta ahamiyat kasb etadi. Bu usullar sonli

ISSN 2181-

analizning klassik adabiyotlarida keng yoritilgan bo‘lib, ular chiziqli sistemalarning taxminiy yechimini bosqichma-bosqich yaqinlashtirishga asoslangan. Shuningdek, bu usullar kompyuter hisoblashlarida qanday qo‘llanilishi va ularning afzalliklari haqida fikr yuritilgan.

Asosiy qism

Yakobi usuli

Yakobi usuli har bir tenglamadan tegishli x_i ni ifodalaydi va uni faqat oldingi iteratsiyadagi qiymatlar asosida hisoblaydi. Bu usul quyidagi ko‘rinishda ifodalanadi:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ijx_j^{(k)}}\right)$$

Bu yerda $a_{ii} \neq 0$ bo‘lishi kerak. Ushbu usuldagi barcha qiymatlar bir vaqtda yangilanadi (Chapra & Canale, 2015).

Gauss-Seidel usuli

Gauss-Seidel usuli Yakobi usulining yaxshilangan varianti bo‘lib, yangi hisoblangan qiymatlarni darhol keyingi hisoblarda ishlatadi:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ijx_j^{(k+1)}} - \sum_{j=i+1}^{n} a_{ijx_j^{(k)}})\right)$$

Bu yondashuvda har bir yangi x_i darhol foydalaniladi, natijada yaqinlashuv tezlashadi (Süli & Mayers, 2003).

Konvergentlik shartlari

Har ikki iteratsion usul uchun konvergentlik **diagonal ustunlik** (diagonal dominance) sharti bajarilganda kafolatlanadi. Bu shart quyidagicha ifodalanadi:

ISSN 2181-

$$|a_{ii}| > \sum_{j\neq 0} |a_{ij}|$$

Aks holda iteratsiyalar yaqinlashmasligi yoki sust yaqinlashuvi mumkin (Atkinson, 1989).

Yakobi va Gauss-Seidel usullarining taqqoslanishi

Yakobi va Gauss-Seidel usullari iteratsion yondashuvlarga asoslangan bo‘lib, ular chiziqli algebraik tenglamalar sistemasini bosqichma-bosqich yaqinlashtirish orqali yechishga xizmat qiladi. Bu ikki usul bir-biridan bir nechta muhim jihatlar bilan farqlanadi:

1. **Yaqinlashuv tezligi**: Yakobi usuli odatda sekinroq yaqinlashadi. Bu usulda har bir yangi qiymat avvalgi iteratsiyadagi natijalarga asoslanib hisoblanadi. Aksincha, Gauss-Seidel usuli esa tezroq yaqinlashadi, chunki u har bir yangi qiymatni hisoblashda allaqachon yangilangan qiymatlardan foydalanadi. Bu esa iteratsiya jarayonini tezlashtiradi.

2. **Hisoblash strukturasining tabiati**: Yakobi usulining muhim afzalligi uning mustaqil va parallel hisoblashlarga mos kelishidir. Har bir tenglama alohida ko‘rilganligi sababli, uni bir vaqtning o‘zida bir nechta protsessorlarda bajarish mumkin. Gauss-Seidel usulida esa hisoblashlar ketma-ket amalga oshiriladi, chunki keyingi o‘zgaruvchi avvalgisining yangilangan qiymatiga bog‘liq bo‘ladi. Bu esa parallel hisoblashni qiyinlashtiradi.

3. **Amalga oshirish murakkabligi**: Yakobi usuli oddiyroq va dasturlashda ham nisbatan osonroq amalga oshiriladi. Gauss-Seidel usuli esa murakkabroq bo‘lib, u natijalarni doimiy yangilab borishni talab qiladi, bu esa dasturiy jihatdan ehtiyotkorlik bilan yondashishni taqozo etadi.

4. **Barqarorlik darajasi**: Yakobi usuli ba’zi hollarda barqarorlik jihatidan pastroq bo‘lishi mumkin, ayniqsa, sistema matritsasi diagonal

ustunlikka ega bo‘lmasa. Gauss-Seidel usuli esa ko‘proq hollarda barqarorroq ishlaydi va yechimga ishonchliroq yaqinlashadi.

Yakobi va Gauss-Seidel usullari **differensial tenglamalarni sonli yechishda** (masalan, to‘r metodlarida) keng qo‘llaniladi. Bunday holatda, chiziqli sistema differensial tenglama diskretlashtirilgach hosil bo‘ladi (Burden & Faires, 2011). Katta hajmdagi siyrak matritsali sistemalarda ayniqsa Gauss-Seidel usuli foydaliroq hisoblanadi.

Amaliy misol

Misol uchun chiziqli tenglamalar sistemasini olaylik:

$$\begin{cases} 10x + 2y + z = 9 \\ 2x + 20y - 2z = -44 \\ -2x + 3y + 10z = 22 \end{cases}$$

C#dagi dastur kodi:

```
using System;

class Program
{
   static void Main()
   {
      double[,] A = {
         { 10, 2, 1 },
         { 2, 20, -2 },
         { -2, 3, 10 }
      };

      double[] b = { 9, -44, 22 };
      int n = b.Length;
```

```
    Console.WriteLine("Yaqinlashtirishlar soni: 25\n");
    Console.WriteLine(">>> Yakobi usuli:");
    JacobiMethod(A, b, n, 25);

    Console.WriteLine("\n>>> Gauss-Seidel usuli:");
    GaussSeidelMethod(A, b, n, 25);
}

static void JacobiMethod(double[,] A, double[] b, int n, int iterations)
{
    double[] x = new double[n]; // natijaviy vektor
    double[] x_new = new double[n];

    for (int it = 0; it < iterations; it++)
    {
        for (int i = 0; i < n; i++)
        {
            double sum = 0;
            for (int j = 0; j < n; j++)
            {
                if (i != j)
                    sum += A[i, j] * x[j];
            }
            x_new[i] = (b[i] - sum) / A[i, i];
        }

        // Natijani chiqarish
        Console.Write("Iteratsiya " + (it + 1) + ": ");
        for (int i = 0; i < n; i++)
```

```
            {
                Console.Write($"{x_new[i]:F4} ");
                x[i] = x_new[i]; // keyingi iteratsiya uchun yangilash
            }
            Console.WriteLine();
        }
    }

    static void GaussSeidelMethod(double[,] A, double[] b, int n, int iterations)
    {
        double[] x = new double[n];

        for (int it = 0; it < iterations; it++)
        {
            for (int i = 0; i < n; i++)
            {
                double sum = 0;
                for (int j = 0; j < n; j++)
                {
                    if (i != j)
                        sum += A[i, j] * x[j];
                }
                x[i] = (b[i] - sum) / A[i, i];
            }

            // Natijani chiqarish
            Console.Write("Iteratsiya " + (it + 1) + ": ");
            for (int i = 0; i < n; i++)
            {
```

ISSN 2181-

```
                Console.Write($"{x[i]:F4} ");
            }
            Console.WriteLine();
        }
    }
}
```

Natija:

```
Консоль отладки Microsoft Visual Studio
Yaqinlashtirishlar soni: 25

>>> Yakobi usuli:
Iteratsiya 1: 0,9000 -2,2000 2,2000
Iteratsiya 2: 1,1200 -2,0700 3,0400
Iteratsiya 3: 1,0100 -2,0080 3,0450
Iteratsiya 4: 0,9971 -1,9965 3,0044
Iteratsiya 5: 0,9989 -1,9993 2,9984
Iteratsiya 6: 1,0000 -2,0000 2,9996
Iteratsiya 7: 1,0001 -2,0000 3,0000
Iteratsiya 8: 1,0000 -2,0000 3,0000
Iteratsiya 9: 1,0000 -2,0000 3,0000
Iteratsiya 10: 1,0000 -2,0000 3,0000
Iteratsiya 11: 1,0000 -2,0000 3,0000
Iteratsiya 12: 1,0000 -2,0000 3,0000
Iteratsiya 13: 1,0000 -2,0000 3,0000
Iteratsiya 14: 1,0000 -2,0000 3,0000
Iteratsiya 15: 1,0000 -2,0000 3,0000
Iteratsiya 16: 1,0000 -2,0000 3,0000
Iteratsiya 17: 1,0000 -2,0000 3,0000
Iteratsiya 18: 1,0000 -2,0000 3,0000
Iteratsiya 19: 1,0000 -2,0000 3,0000
Iteratsiya 20: 1,0000 -2,0000 3,0000
Iteratsiya 21: 1,0000 -2,0000 3,0000
Iteratsiya 22: 1,0000 -2,0000 3,0000
Iteratsiya 23: 1,0000 -2,0000 3,0000
Iteratsiya 24: 1,0000 -2,0000 3,0000
Iteratsiya 25: 1,0000 -2,0000 3,0000
```

```
>>> Gauss-Seidel usuli:
Iteratsiya 1: 0,9000 -2,2900 3,0670
Iteratsiya 2: 1,0513 -1,9984 3,0098
Iteratsiya 3: 0,9987 -1,9989 2,9994
Iteratsiya 4: 0,9998 -2,0000 3,0000
Iteratsiya 5: 1,0000 -2,0000 3,0000
Iteratsiya 6: 1,0000 -2,0000 3,0000
Iteratsiya 7: 1,0000 -2,0000 3,0000
Iteratsiya 8: 1,0000 -2,0000 3,0000
Iteratsiya 9: 1,0000 -2,0000 3,0000
Iteratsiya 10: 1,0000 -2,0000 3,0000
Iteratsiya 11: 1,0000 -2,0000 3,0000
Iteratsiya 12: 1,0000 -2,0000 3,0000
Iteratsiya 13: 1,0000 -2,0000 3,0000
Iteratsiya 14: 1,0000 -2,0000 3,0000
Iteratsiya 15: 1,0000 -2,0000 3,0000
Iteratsiya 16: 1,0000 -2,0000 3,0000
Iteratsiya 17: 1,0000 -2,0000 3,0000
Iteratsiya 18: 1,0000 -2,0000 3,0000
Iteratsiya 19: 1,0000 -2,0000 3,0000
Iteratsiya 20: 1,0000 -2,0000 3,0000
Iteratsiya 21: 1,0000 -2,0000 3,0000
Iteratsiya 22: 1,0000 -2,0000 3,0000
Iteratsiya 23: 1,0000 -2,0000 3,0000
Iteratsiya 24: 1,0000 -2,0000 3,0000
Iteratsiya 25: 1,0000 -2,0000 3,0000
```

Ushbu chiziqli sistema uchun ikkala usul ham quyidagi yaqin yechimni beradi:

$$x = 1.000, \qquad y = -2.000, \qquad z = 2.000$$

Gauss-Seidel usuli natijaga tezroq yaqinlashadi. Yakobi usuli esa ko‘proq iteratsiya talab qiladi.

Xulosa

Yakobi va Gauss-Seidel iteratsion usullari chiziqli tenglamalar sistemasining sonli yechimini topishning klassik va samarali metodlari hisoblanadi. Yakobi usuli soddaligi bilan ajralib tursa, Gauss-Seidel usuli yaqinlashuv tezligi bilan ustunlik qiladi. Har ikki metodning konvergentligi uchun matritsaning strukturasiga bog‘liq shartlar bajarilishi lozim. Bu usullar klassik sonli analiz adabiyotlarida isbotlangan nazariy asosga ega va amaliy hisoblashlarda keng qo‘llaniladi.

Foydalanilgan adabiyotlar

1. Burden, R. L., & Faires, J. D. (2011). ***Numerical Analysis***. Cengage Learning.
2. Chapra, S. C., & Canale, R. P. (2015). ***Numerical Methods for Engineers***. McGraw-Hill Education.
3. Süli, E., & Mayers, D. F. (2003). ***An Introduction to Numerical Analysis***. Cambridge University Press.
4. Atkinson, K. (1989). ***An Introduction to Numerical Analysis***. John Wiley & Sons.
5. Давыдов, Ю.Н. (2005). ***Численные методы***. Москва: Высшая школа.