

CHIZIQLI TENGLAMALAR SISTEMASI YECHIMINI C++ DASTURI YORDAMIDA KRAMER USULIDA TOPISH

Butaboyeva Durdona Olimjon qizi

<https://doi.org/10.5281/zenodo.15104381>

Ma'lumki, kramer usuli (yoki Cramer's rule) - determinantlar yordamida chiziqli tenglamalar sistemasini yechish usuli bo'lib, kvadrat matritsaga ega bo'lgan sistemalar uchun qo'llaniladi. Bu usul har bir noma'lumning qiymatini topish uchun asosiy matritsaning determinantlari bilan ishlaydi. Asosan 2×2 , 3×3 kabi kichik o'lchamli tenglamalar sistemalarida samarali bo'ladi.

Kramer usuli odatda determenantlar usuli ham deb ataladi. Bu usulning algoritmi quyidagicha. Dastlab quyidagi $(n+1)$ ta n - tartibli

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} \quad \Delta_{x_1} = \begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ b_n & a_{n2} & \dots & a_{nn} \end{vmatrix} \quad \Delta_{x_n} = \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_n \end{vmatrix}$$

$$x_1 = \frac{\Delta_{x_1}}{\Delta}, \quad x_2 = \frac{\Delta_{x_2}}{\Delta}, \dots,$$

determinantlarning qiymatlari hisoblanadi va noʻmalumlar

$$x_n = \frac{\Delta_{x_n}}{\Delta}$$

Δ formulalar yordamida topiladi [5].

Chiziqli tenglamalar sistemasi yechimini Oddiy iteratsiya usulida topish

[illegible]

(1) tenglamalar tizimi almashtirishlar va belgilashlar natijasida quyidagi tizimga olib kelinadi:

[illegible]

$$\beta_i = \frac{b_i}{a_{ii}}; \quad \alpha_{ij} = -\frac{a_{ij}}{a_{ii}}; \quad a_{ii} = 0$$

Bu yerda

Bu yangi olingan tizimni qulaylik uchun matritsa ko'rinishida yozamiz

$$X = B + AX$$

Bu yerda - $X=\{x_i\}$, $B=\{\beta_j\}$ vektor; A diagonal elementlari nolga teng matritsa $A=\{\alpha_{ij}\}$

Bu tizimni ketma-ket yaqinlashish usuli bilan yechamiz. $B=\{\beta_{ij}\}$ vektorini boshlang'ich nolinchini yaqinlashish deb olib, $(k+1)$ yaqinlashish hisoblash formulasini quyidagicha yozamiz [3]

$$X^{(k+1)} = B + AX^{(k)} \quad (k = 0, 1, 2, \dots, n).$$

Bu ketma-ket yaqinlashish $x^{(0)}, x^{(1)}, x^{(2)}, \dots, x^{(k)}$ limitga ega bo'lsa, u holda bu limit tizim yechimi bo'ladi. Iteratsiya usuli bo'yicha hisoblash oldindan berilgan aniqlik $\varepsilon > 0$ son uchun

$$|X^k - X^{k-1}| \leq \varepsilon$$

tengsizlik bajarilgunga qadar davom etadi. Iteratsiya jarayoni yaqinlashuvchi bo'lishi uchun A matritsaning

$$\|A\|_1 = \max_i \sum_{j=1}^n |\alpha_{ij}| < 1; \quad \|A\|_2 = \max_j \sum_{i=1}^n |\alpha_{ij}| < 1; \quad \|A\|_3 = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |\alpha_{ij}|^2} < 1$$

formularidan birortasi bajarilishi yetarlidir [4].

Kramer usuli (Cramer's rule) chiziqli tenglamalar sistemasining yechimlarini determinantlar yordamida topishga asoslangan. Quyida C++ tilida n o'lchovli chiziqli tenglamalar sistemasini Kramer usulida yechish uchun dastur algoritmini tuzamiz [1].

Dastur algoritmi:

1. Matritsa determinantini hisoblash.
2. Har bir o'zgaruvchi uchun determinantlarni o'zgartirilgan matritsa yordamida hisoblash.
3. Kramer usuliga ko'ra o'zgaruvchilarning qiymatlarini topish.

Dastur uchta tenglama va uchta noma'lum uchun yozildi [6].

```
#include <iostream>
#include <vector>
using namespace std;
// Matritsa determinantini hisoblaydigan funksiya (3x3 matritsa uchun)
double determinant3x3(vector<vector<double>> mat) {
    double det = mat[0][0] * (mat[1][1] * mat[2][2] - mat[1][2] * mat[2][1])
        - mat[0][1] * (mat[1][0] * mat[2][2] - mat[1][2] * mat[2][0])
        + mat[0][2] * (mat[1][0] * mat[2][1] - mat[1][1] * mat[2][0]);
    return det;
}
// Kramer usuli bilan tenglamalar sistemasini yechish funksiyasi [2]
void kramerMethod(vector<vector<double>> A, vector<double> B) {
    // Asosiy matritsaning determinantini hisoblash
    double detA = determinant3x3(A);
    if (detA == 0) {
        cout << "Tizimning yechimi yo'q yoki cheksiz ko'p yechimga ega." << endl;
        return;
    }
    // X1 o'zgaruvchisi uchun determinantni hisoblash
```

```

vector<vector<double>> A1 = A;
for (int i = 0; i < 3; i++) {
    A1[i][0] = B[i]; // 1-ustunni B vektori bilan almashtiramiz
}
double detA1 = determinant3x3(A1);

// X2 o'zgaruvchisi uchun determinantni hisoblash
vector<vector<double>> A2 = A;
for (int i = 0; i < 3; i++) {
    A2[i][1] = B[i]; // 2-ustunni B vektori bilan almashtiramiz
}
double detA2 = determinant3x3(A2);
// X3 o'zgaruvchisi uchun determinantni hisoblash
vector<vector<double>> A3 = A;
for (int i = 0; i < 3; i++) {
    A3[i][2] = B[i]; // 3-ustunni B vektori bilan almashtiramiz
}
double detA3 = determinant3x3(A3);
// Kramer usuli bo'yicha yechimlarni hisoblash
double x1 = detA1 / detA;
double x2 = detA2 / detA;
double x3 = detA3 / detA;
// Yechimlarni ekranga chiqaramiz
cout << "Tenglamalar sistemasining yechimlari:" << endl;
cout << "x1 = " << x1 << endl;
cout << "x2 = " << x2 << endl;
cout << "x3 = " << x3 << endl;
}
int main() {
    // 3x3 matritsa (tenglamalar sistemasining koeffitsientlari)
    vector<vector<double>> A = {
        {2, -1, 3},
        {4, 1, 2},
        {3, 2, -4}
    };
    // B vektori (tenglamalarning o'ng tomoni)
    vector<double> B = {5, 9, -6};
    // Kramer usuli orqali yechishni chaqiramiz
    kramerMethod(A, B);
    return 0;}

```

- determinant3x3 funksiyasi 3x3 matritsaning determinantini hisoblaydi.

- kramerMethod funksiyasi Kramer usuliga ko'ra x_1 , x_2 va x_3 o'zgaruvchilarining qiymatlarini topadi.

Dasturdagi matritsa va vektorlarni xohlaganingizcha o'zgartirishingiz mumkin.

Foydalanilgan adabiyotlar/Используемая литература/References:

1. Shohamidov Sh. Amaliy matematika unsurlari. O'quv qo'llanma. Toshkent, "Fan va texnologiya", 2016, 212 b.
2. Raisov M. Matematik programmalash. Darslik. "Cho'lpon nomidagi" NMIU Tashkent: 2013, 208 b.
3. Xabibullayeva I. Iqtisodiy matematik usullar va modellar. O'quv qo'llanma. Toshkent, "Tafakkur bo'stoni", 2012, 112 b.
4. Isroilov M. Hisoblash usullari. Darslik. Toshkent, "Iqtisod-Moliya", 2008, 320 b
5. Mo'minov Sh. Matematik modellar va usullar. Darslik. Toshkent, "Turon-Istiqbol", 2006, 272 b.
6. Safoeva M. Matematik programmalash. O'quv qo'llanma. Toshkent, "UJBHT", 2004, 240 b.



INNOVATIVE
ACADEMY