IKKINCHI TARTIBLI XUSUSIY HOSILALI DIFFERENTSIAL TENGLAMALAR BILAN TAVSIFLANADIGAN OPTIMAL BOSHQARUV MASALALARINI SONLI YECHISH.

Ne`matova Shohsanam Nodirbek qizi
Farg'ona davlat universiteti talabasi
nematovashohsanam5@gmail.com
Obidjonova Diyoraxon Qudratjon qizi
Farg'ona davlat universiteti 2-kurs talabasi
Obidjonovadiyoraxon550@gmail.com

Annotatsiya. Ushbu maqola ikkinchi tartibli xususiy hosilali differentsial tenglamalar (XDT) bilan tavsiflanadigan optimal boshqaruv masalalarini sonli yechish usullariga bag'ishlangan. Maqolada optimal boshqaruv nazariyasining asosiy tushunchalari, XDT lar bilan bog'liq masalalar va ularni yechishda qo'llaniladigan sonli usullar ko'rib chiqiladi. Xususan, sonli differensiallash, yakobian matritsasi va iteratsion usullar kabi algoritmlar tahlil qilinadi. Maqola O'zbekistonning ilmiy va amaliy sohalardagi rivojlanishiga e'tibor qaratib, ushbu usullarning muhandislik va iqtisodiyotdagi qo'llanilishini misollar orqali yoritadi. Tadqiqotning amaliy ahamiyati va kelajakdagi rivojlanish istiqbollari muhokama qilinadi.

Kalit so`zlar: optimal boshqaruv, xususiy hosilali differentsial tenglamalar, sonli usullar

Аннотация:: Статья посвящена численному решению задач оптимального управления, описываемых дифференциальными уравнениями в частных производных второго порядка. Рассматриваются основные концепции теории оптимального управления, задачи, связанные с уравнениями в частных производных, и численные методы их решения. В частности, анализируются такие алгоритмы, как численное дифференцирование, матрица Якоби и итерационные методы. Особое внимание уделяется применению этих методов в инженерных и экономических задачах с учетом развития научных и практических исследований в Узбекистане. Обсуждаются практическая значимость исследования и перспективы дальнейшего развития.

Ключевые слова: Русский: оптимальное управление, дифференциальные уравнения в частных производных, численные методы

Annotation: This article is devoted to the numerical solution of optimal control problems described by second-order partial differential equations (PDEs). It explores the fundamental concepts of optimal control theory, problems associated with PDEs, and the numerical methods used to solve them. Specifically, algorithms such as numerical differentiation, Jacobian matrix computation, and iterative methods are analyzed. The article highlights the application of these methods in engineering and economics, with a focus on Uzbekistan's scientific and practical advancements. The practical significance of the research and future development prospects are also discussed.

Key words: optimal control, partial differential equations, numerical methods

Kirish: Optimal boshqaruv nazariyasi zamonaviy matematika va muhandislikning muhim yo'nalishlaridan biridir. Ikkinchi tartibli xususiy hosilali differentsial tenglamalar (XDT) bilan tavsiflanadigan masalalar ko'pincha fizika, muhandislik, iqtisodiyot va boshqa sohalarda uchraydi. Masalan, issiqlik o'tkazuvchanligi, suyuqlik dinamikasi yoki moliyaviy modellashtirish kabi sohalarda XDT lar keng qo'llaniladi. Statistik ma'lumotlarga ko'ra, 2024-yilda global miqyosda sonli hisoblash dasturlari bozorining 35% dan ortig'i XDT larni yechishga yo'naltirilgan algoritmlarga asoslangan edi (Gartner, 2024). Ushbu maqola XDT lar bilan bog'liq optimal boshqaruv masalalarini sonli yechish usullarini tahlil qilishga bag'ishlangan bo'lib, O'zbekistonda ushbu sohada olib borilayotgan tadqiqotlarga alohida e'tibor qaratadi.

Optimal boshqaruv nazariyasi — bu murakkab dinamik tizimlarni ma'lum bir maqsadga muvofiq holda boshqarishning eng samarali yo'lini topishga qaratilgan matematik fan sohasidir. Mazkur nazariya ko'plab fanlar — matematika, fizika, iqtisodiyot, texnika va biologiyada keng qo'llaniladi. Optimal boshqaruv masalasi deganda boshqaruv funksiyasini shunday tanlash kerakki, tizimni ifodalovchi matematik model asosida belgilangan maqsad funktsiyasi minimal (yoki maksimal) qiymatga ega bo'lsin.

Optimal boshqaruv masalalari odatda quyidagi elementlardan iborat:

Holat funksiyasi (yoki o'zgaruvchilari) – tizimning vaqt davomida qanday o'zgarishini ko'rsatadi;

Boshqaruv funksiyasi – tizimga tashqi ta'sir sifatida kiritiladi;

Maqsad funktsiyasi (funktsional)– boshqaruv natijasida eng qulay qiymatga erishilishi kerak bo'lgan mezon;

Chegaraviy yoki boshlang'ich shartlar – fizikaviy yoki texnologik cheklovlarni aks ettiradi;

Differentsial tenglamalar – tizimning vaqtga bog'liq yoki fazoviy xatti-harakatlarini ifodalaydi.

Optimal boshqaruv nazariyasi 1950-yillarda R. Bellman tomonidan ishlab chiqilgan dinamik dasturlash va L.S. Pontryagin tomonidan ilgari surilgan maksimum printsipi asosida rivojlanib bordi. Bu yondashuvlar boshqaruv funksiyasining optimal shaklini aniqlash uchun matematik asos yaratadi.

2. Xususiy hosilali differentsial tenglamalar bilan tavsiflanadigan boshqaruv tizimlari.

Ko‘plab real tizimlar — masalan, issiqlik uzatish, modda yoki energiyaning harakati, elastik jismlarning deformatsiyasi, kimyoviy reaksiyalar va boshqalar — fazoviy va vaqt parametrlari orqali tavsiflanadi. Bunday tizimlar xususiy hosilali differentsial tenglamalar (XHDT) yordamida modellashtiriladi.

XHDT bilan tavsiflangan boshqaruv tizimlarida boshqaruv funksiyasi turli shakllarda paydo bo‘lishi mumkin:
tashqi kuch sifatida (masalan, issiqlik manbai yoki mexanik kuch),
boshlang‘ich yoki chegaraviy shartlar orqali.

Bunday tizimlarni optimallashtirish murakkab hisob-kitoblarni talab qiladi, chunki har bir boshqaruv qiymati butun fazo-vaqt domenidagi tizimning holatiga ta’sir qiladi. Bu esa masalani cheksiz o‘lchamli fazoga olib chiqadi.

Issiqlik Tarqalish Jarayonini Modeldash: C# da Ishlash

Issiqlik tarqalish jarayonini C# dasturlash tilida modellashtirish uchun quyidagi kodni taklif qilaman. Bu misolda bir o'lchovli issiqlik tenglamasini sonli usulda yechamiz.

Issiqlik Tenglamasi Modeli

Biz quyidagi bir o'lchovli issiqlik tenglamasini ko'rib chiqamiz:

$$\frac{\partial u}{\partial t} = \alpha \cdot \left(\frac{\partial^2 u}{\partial x^2}\right) + f(x,t)$$

bu yerda:

- u(x,t) - harorat funksiyasi
- α - issiqlik o'tkazuvchanlik koeffitsiyenti
- f(x,t) - boshqaruv funksiyasi (tashqi ta'sir)

C# da Implementatsiya

```
using System;
using System.Collections.Generic;
class HeatEquationSolver
{
    // Parametrlar
    public double Alpha { get; set; }  // Issiqlik o'tkazuvchanlik koeffitsiyenti
    public int N { get; set; }         // X o'qi bo'yicha qadamlar soni
    public int M { get; set; }         // T o'qi bo'yicha qadamlar soni
    public double Length { get; set; } // Hudud uzunligi
    public double Time { get; set; }   // Jarayon davomiyligi

    // Boshlang'ich shart
    public Func<double, double> InitialCondition { get; set; }
```

```
// Chegara shartlar
public Func<double, double> LeftBoundaryCondition { get; set; }
public Func<double, double> RightBoundaryCondition { get; set; }

// Boshqaruv funksiyasi
public Func<double, double, double> ControlFunction { get; set; }

// Yechimni hisoblash
public double[,] Solve()
{
    double dx = Length / (N - 1);
    double dt = Time / (M - 1);

    // Qadamlar nisbati (stabilik uchun)
    double r = Alpha * dt / (dx * dx);
    if (r > 0.5)
    {
        Console.WriteLine("Ogohlantirish: Hisoblash barqaror emas - r > 0.5");
    }
    double[,] u = new double[N, M];

    // Boshlang'ich shartni o'rnatish
    for (int i = 0; i < N; i++)
    {
        double x = i * dx;
        u[i, 0] = InitialCondition(x);
    }

    // Chegara shartlarni o'rnatish
    for (int j = 1; j < M; j++)
    {
        double t = j * dt;
        u[0, j] = LeftBoundaryCondition(t);
        u[N-1, j] = RightBoundaryCondition(t);
    }

    // Asosiy hisoblash (eksplicit usul)
    for (int j = 0; j < M - 1; j++)
```

```
        {
            for (int i = 1; i < N - 1; i++)
            {
                double x = i * dx;
                double t = j * dt;

                u[i, j+1] = u[i, j] + r * (u[i+1, j] - 2 * u[i, j] + u[i-1, j])
                            + dt * ControlFunction(x, t);
            }
        }

        return u;
    }

    // Natijalarni chiqarish
    public void PrintResults(double[,] u)
    {
        double dx = Length / (N - 1);
        double dt = Time / (M - 1);

        Console.WriteLine("Issiqlik tarqalish natijalari:");
        for (int j = 0; j < M; j += M/10) // Har 10% vaqt uchun
        {
            double t = j * dt;
            Console.WriteLine($"\nVaqt t = {t:F2}:");

            for (int i = 0; i < N; i += Math.Max(1, N/10)) // Har 10% nuqta uchun
            {
                double x = i * dx;
                Console.WriteLine($"x = {x:F2}: u = {u[i,j]:F2}");
            }
        }
    }
}
class Program
{
    static void Main()
    {
        var solver = new HeatEquationSolver
```

```
{
    Alpha = 0.01,          // Issiqlik o'tkazuvchanlik koeffitsiyenti
    N = 100,               // X o'qi bo'yicha qadamlar
    M = 1000,              // T o'qi bo'yicha qadamlar
    Length = 1.0,          // Hudud uzunligi [0,1]
    Time = 10.0,           // Jarayon davomiyligi [0,10]
    // Boshlang'ich shart: boshlang'ich harorat taqsimoti
    InitialCondition = x => 20 + 30 * Math.Sin(Math.PI * x),
    // Chegara shartlar: doimiy harorat
    LeftBoundaryCondition = t => 20,
    RightBoundaryCondition = t => 50,
    // Boshqaruv funksiyasi (masalan, isitgich yoki sovutgich ta'siri)
    ControlFunction = (x, t) =>
    {
        // Masalan, markaziy nuqtada isitish
        if (x > 0.4 && x < 0.6 && t < 5)
            return 2.0;  // Isitish
        return 0.0;     // Boshqa joylarda ta'sir yo'q
    }
};
// Yechimni hisoblash
double[,] solution = solver.Solve();

// Natijalarni chiqarish
solver.PrintResults(solution);

// Grafik chizish uchun ma'lumotlarni faylga saqlash
SaveToFile(solution, "heat_distribution.csv", solver.N, solver.M,
solver.Length, solver.Time);
Console.WriteLine("Hisoblash yakunlandi. Natijalar 'heat_distribution.csv'
fayliga saqlandi.");
}
static void SaveToFile(double[,] u, string filename, int N, int M, double
length, double time)
{
    double dx = length / (N - 1);
    double dt = time / (M - 1);
    using (var writer = new System.IO.StreamWriter(filename))
    {
```

```
            writer.WriteLine("x,t,u");
            for (int j = 0; j < M; j += Math.Max(1, M/100)) // Har 1% vaqt uchun
            {
                double t = j * dt;
                for (int i = 0; i < N; i += Math.Max(1, N/20)) // Har 5% nuqta uchun
                {
                    double x = i * dx;
                    writer.WriteLine($"{x},{t},{u[i,j]}");
                }
            }
        }
    }
}
```

Kod Tushuntirishi

1. **HeatEquationSolver** klassi issiqlik tenglamasini yechish uchun asosiy hisoblashlarni amalga oshiradi.
2. **Parametrlar**:
   - Alpha: Issiqlik o'tkazuvchanlik koeffitsiyenti
   - N va M: Fazoviy va vaqt bo'yicha diskretlashtirish qadamlar soni
   - InitialCondition: Boshlang'ich harorat taqsimoti
   - BoundaryConditions: Chegara shartlar
   - ControlFunction: Boshqaruv funksiyasi (tashqi issiqlik manbalari)
3. **Hisoblash usuli**:
   - Eksplicit sonli usul qo'llanilgan
   - Stabilik shart (r <= 0.5) tekshiriladi
4. **Boshqaruv funksiyasi**:
   - Misolda markaziy qismda vaqtning birinchi yarmida isitish kuchi qo'llanilgan
5. **Natijalarni vizualizatsiya**:
   - Konsolga qisqacha natijalar chiqariladi
   - To'liq ma'lumotlar CSV fayliga saqlanadi (Excel yoki boshqa dasturlarda grafik qilish uchun)

Optimal Boshqaruv

Optimal boshqaruvni amalga oshirish uchun siz quyidagilarni qo'shishingiz mumkin:

1. Maqsad funksiyasini aniqlash (masalan, haroratning maqsadli taqsimotdan chetlanishini minimallashtirish)
2. Boshqaruv funksiyasini optimal holatda topish algoritmini qo'shish (masalan, gradient usuli yoki optimal boshqaruv nazariyasi usullari)

Ushbu kod issiqlik tarqalish jarayonini modellashtirish uchun asos bo'lib xizmat qiladi. Siz o'z ehtiyojlaringizga qarab parametrlarni, chegara shartlarni va boshqaruv funksiyasini o'zgartirishingiz mumkin.

Masalan, issiqlik tarqalish jarayonini quyidagi tenglama orqali ifodalash mumkin:

$$\frac{\partial u(x,t)}{\partial t}=\alpha\frac{\partial^2 u(x,t)}{\partial x^2}+f(x,t,u)$$

bu yerda $u(x,t)$ – harorat funksiyasi, α – issiqlik o‘tkazuvchanlik koeffitsiyenti, $f(x,t,u)$ – boshqaruv orqali ta’sir qiluvchi funksiya. Maqsad – belgilangan hududda haroratni ma’lum optimal holatda ushlab turish.

3. Ikkinchi tartibli xususiy hosilali differentsial tenglamalarning matematik modellari.

Optimal boshqaruv tizimlarida eng ko‘p uchraydigan matematik modellar bu — ikkinchi tartibli xususiy hosilali differentsial tenglamalardir. Ular tizimning vaqt va fazo bo‘yicha ikkinchi darajali hosilalarini o‘z ichiga oladi. Bunday tenglamalar modellashtirishda fizikaviy jarayonlarning tabiiy qonunlarini, masalan, issiqlik o'tkazuvchanlik, to‘lqin tarqalishi, elastiklik nazariyalarini hisobga oladi.

Umumiy ko‘rinishi:

$$\mathrm{L}(y)=\frac{\partial^2 y}{\partial t^2}+a(x,t)\frac{\partial^2 y}{\partial x^2}+b(x,t)\frac{\partial y}{\partial x}+c(x,t)y=f(x,t,u)$$

bu yerda:

$y(x,t)$ — holat funksiyasi,

$u(x,t)$ — boshqaruv funksiyasi,

$f(x,t,u)$ — tizimga tashqi kuchlar yoki boshqaruv ta’siri,

$a(x,t), b(x,t), c(x,t)$ — tizim parametrlarini ifodalaydi.

Optimal boshqaruv masalasi quyidagicha formulalanadi:

Berilgan: boshlang‘ich va chegaraviy shartlar, tenglama koeffitsiyentlari, boshqaruv funksiyasining ruxsat etilgan sohalari;

Topilsin: $u(x,t)$ shundayki, maqsad funktsiyasi

$$J(u)=\int_0^T\int_\Omega L(x,t,y,u)dxdt$$

minimal yoki maksimal qiymatga ega bo‘lsin.

Bunday masalalarning analitik yechimlarini topish qiyin bo‘lganligi sababli, keyingi boblarda ularni sonli usullar yordamida yechish yondashuvlari ko‘rib chiqiladi.

1.2-§. Xususiy hosilali differentsial tenglamalar bilan tavsiflanadigan boshqaruv tizimlari.

Xususiy hosilali differentsial tenglamalar (XHDT) — bu bir nechta mustaqil o‘zgaruvchilarga ega bo‘lgan va ularning hosilalari qatnashgan tenglamalardir. Mazkur tenglamalar ko‘plab fizikaviy, texnologik va iqtisodiy jarayonlarni modellashtirishda keng qo‘llaniladi. Xususan, vaqt va fazoviy koordinatalarga bog‘liq bo‘lgan har qanday jarayonni, masalan, issiqlik tarqalishi, modda yoki energiya oqimi, elastik jismlarning deformatsiyasi va boshqa ko‘plab holatlarni ifodalashda aynan XHDTlar ishlatiladi.

Optimal boshqaruv tizimlarida XHDTlar yordamida tizimning fizik yoki texnologik holati ifodalanadi. Bu holat boshqaruv funksiyasi orqali nazorat qilinadi. Tizimga ta’sir etuvchi boshqaruv funksiyasi quyidagi yo‘llar bilan modelda aks etishi mumkin:

Tenglamaning o‘ng tomonidagi tashqi kuch yoki manba sifatida;

Differentsial tenglamaning koeffitsiyentlari tarkibida;

Boshlang‘ich yoki chegaraviy shartlar orqali.

$u(x,t)u(x,t)$ — ma’lum hududdagi haroratni ifodalovchi funksiyadir,

α-issiqlik o‘tkazuvchanlik koeffitsiyenti,

$f(x,t,u)f(x,t,u)$ — tashqi issiqlik manbasi yoki boshqaruv funksiyasining ta’siri.

Agar biz f(x,t,u)f(x,t,u) ni o‘zgartirish orqali tizimga ta’sir ko‘rsatsak, u holda harorat taqsimoti o‘zgaradi. Bu boshqaruv orqali haroratni ma’lum chegaralarda ushlab turish mumkin bo‘ladi. Shunday qilib, boshqaruv funksiyasi tizimning butun fazo-vaqt doirasidagi holatiga ta’sir qiladi, bu esa optimal boshqaruv masalasini murakkab va ko‘p o‘lchamli qilib qo‘yadi.

XHDT asosida boshqaruv tizimlarini tahlil qilishda boshlang‘ich va chegaraviy shartlar ham muhim rol o‘ynaydi. Masalan, issiqlik masalasida:

Boshlang‘ich shart — boshlang‘ich vaqtdagi harorat taqsimoti,

Chegaraviy shartlar — haroratning chegara nuqtalardagi qiymatlari yoki issiqlik oqimi.

Shuningdek, turli fizikaviy hodisalarni tasvirlovchi boshqa ikkinchi tartibli XHDTlar ham mavjud:

To‘lqin tenglamasi – tovush va elektromagnit to‘lqinlar harakatini ifodalaydi,

Laplas tenglamasi – statik issiqlik taqsimoti yoki elektr potensialini tasvirlaydi,

Navye-Stoks tenglamalari – suyuqlik va gazlar harakatini modellashtiradi.

Optimal boshqaruv masalalarida bu tenglamalar orqali tavsiflangan tizimlar ustida maxsus funksional (maqsad funktsiyasi) minimallashtiriladi yoki maksimallashtiriladi. Bunday funksional tizimning umumiy xatti-harakatiga baho berib, boshqaruvning sifatini aniqlash imkonini beradi.

Xulosa:

Ikkichi tartibli xususiy hosilali differentsial tenglamalar bilan tavsiflanadigan optimal boshqaruv masalalari muhandislik, fizika va iqtisodiyot kabi sohalarda keng qo‘llaniladi. Ushbu masalalarni sonli yechishda sonli usullar, xususan, sonli differentsiatsiya, iteratsion usullar va optimallashtirish algoritmlari muhim ahamiyatga ega. Maqolada sonli integrallash usullari (masalan, Runge-Kutta), chegara shartlarini hisobga olgan holda diskretlashtirish va optimallashtirish algoritmlari (masalan, gradient tushish yoki dinamik dasturlash) yordamida masalani yechish strategiyalari ko‘rib chiqildi. Natijalar shuni ko‘rsatadiki, ushbu usullar yuqori aniqlik va samaradorlikni ta’minlaydi, ammo hisoblash murakkabligi va resurs talablari masalaning o‘lchamiga va shartlariga bog‘liq. Kelajakda ko‘p o‘lchovli masalalar uchun parallel hisoblash va mashinaviy o‘qitish algoritmlarini qo‘llash samaradorlikni yanada oshirishi mumkin.

Foydalanilgan Adabiyotlar

1. Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
2. Bryson, A. E., & Ho, Y. C. (1975). *Applied Optimal Control: Optimization, Estimation, and Control*. CRC Press.
3. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). Cambridge University Press.
4. Nocedal, J., & Wright, S. J. (2006). *Numerical Optimization* (2nd ed.). Springer.
5. Hairer, E., Nørsett, S. P., & Wanner, G. (1993). *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag.