

Тел: +7 (707) 900 92 67

Почта: saken.yan@yandex.com

11 ЛЕКЦИЯ

ФИЗИКАЛЫҚ МІНДЕТТЕРДІ КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ. §11. МОНТЕ-КАРЛО ӘДІСІ.

Жалпы түсінік.

Монте-Карло әдістері - стохастикалық процестерді зерттеуге арналған сандық әдістердің тобы болып табылады. Әдістің жалпы мағынасы мынада:

Физикалық процесс кездейсоқ айнымалы генераторының көмегімен математикалық модельмен сипатталады, алынған мәліметтер негізінде модель бірнеше рет есептеледі, қарастырылып отырған процестің ықтималдылық сипаттамалары мен қасиеттері есептеледі.

Монте-Карло әдісі физика, химия, биология және т.б. салаларда қолданылады.

Әдістің заманауи нұсқасы Манхэттен жобасы (атом бомбасын жасау жобасы) аясында құрылған, Бұл жоба барысында нейтрондардың әр түрлі материалдарда жүру қашықтығын монте-карло әдісі арқылы модельдеу жүргізілді. Бұл жобадан кейін монте-карло әдісі көптеген басқа салаларда таралды.

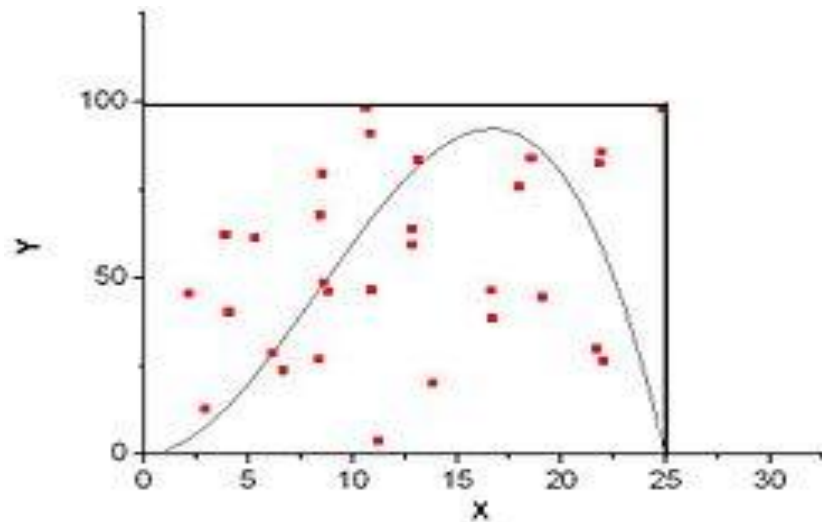
Монте-Карло әдісінің үлкен артықшылығы - бұл модельдеу барысында нақты әлемнің кездейсоқтығы мен күрделілігін ескеруге мүмкіндік береді.

Функцияны интегралдаудың монте-Карло алгоритмі.

Берілген функциямен шектелген облыстың ауданын есептеу үшін келесі стохастикалық алгоритмді қолдануға болады:

1. Берілген функцияны тіктөртбұрышпен қоршаймыз (көптеген өлшемді жағдайында n өлшемді параллелепипедпен), оның «ауданы» S_{rect} оңай есептеледі;
2. Осы тіктөртбұрыш (параллелепипед) ішінде N – нүктелерді кездейсоқ етіп орналастырамыз.
3. Функция графигінің астында орналасқан нүктелердің санын (K дана) есептейміз.
4. Сонда берілген функциямен шектелген облыстың ауданы мына қатынаспен анықталады

$$S = S_{rect} \frac{K}{N}$$



Сурет.12.1. - Монте-Карло әдісі арқылы функцияның интегралын есептеу.

Кіріспе.

Қарастыратын бөлшегіміз нейтрон болсын. Нейтронның затпен әрекеттесуін компьютерлік модельдеу - атом энергетикасының негізгі міндеттерінің бірі болып табылады. Ядролық реакторларда жүретін процестерді компьютерлік модельдеу барысында нейтрондардың бір бірімен әсерлесуі қарастырылмайды.

Реакторлар физикасы мен сәулеленуден қорғану мәселелерінің сызықтығы нейтрондардың аз популяциясының (әдетте $10^3 - 10^5$ бөлшектерінің) затпен әсерлесуін компьютерлік модельдеу нәтижелерін эксперимент барысында шешуге тура келетін үлкен популяциялары үшін қолдануға мүмкіндік береді (атом электр станцияларының реакторларында, мысалы, $10^{15} - 10^{16}$ нейтрондар ағыны бар).

Егер ортаның изотоптық құрамы және нейтрондардың осы изотоптармен әрекеттесуінің микроскопиялық заңдары белгілі болса, онда біздің қолымызда нейтрондардың компьютерде таралу процесін модельдеу үшін қажетті барлық ақпарат бар деп есептейміз. Сол себепті нейтрондардың таралуын модельдеу кездейсоқ процесс болып табылады, өйткені аталған микроскопиялық заңдар, әдетте, детерминистік емес, кездейсоқ сипатқа ие болады. Нейтронның ядромен әрекеттесуі бірнеше реакция каналдардың біреуі арқылы өтуі мүмкін. Тәжірибе (немесе ядро теориясы) реакция каналдарының ықтималдығын немесе шексіз көп реакция каналдары болған жағдайда ықтималдық тығыздығын анықтауға мүмкіндік береді.

Есептің қойылуы.

Айталық, қалыңдығы D болатын біртекті пластинаға нормал бағытта түскен нейтрондардың осы пластинадан өту ықтималдылығын есептеп табу қажет болсын. Айталық, нейтронның затпен әрекеттесуі барысында тек екі процесс қана — изотроп шашырау және нейтронның жұтылуы. Осы екі процесстің (ядролық реакция) макроэкономикалық көлденен қималары сәйкесінше Σ_s және Σ_a болсын. Олай болса, нейтронның затпен әсерлесуінің толық макроэкономикалық қимасы $\Sigma_t = \Sigma_s + \Sigma_a$ тең.

Толық қиманың анықтамасы бойынша нейтрон dl жолында $\Sigma_t dl$ ықтималдылығымен затпен әсерлеседі. Макроэкономикалық қимасы Σ_t болатын шексіз біртекті ортада екі соқтығысу арасында нейтронның жүру жолын кездейсоқ шама деп алып, оның $\rho(l)$ үлестірілу тығыздығын табайық. Егер нейтрон l жолында зат ядроларымен соқтығыспай, содан кейін dl жолында соқтығысса, онда нейтронның еркін жүру жолы $(l, l + dl)$ аралығында болады. Демек, $p(l)$ үшін келесі теңдік орындалады.

$$p(l)dl = [1 - \int_0^l p(x)dx]\Sigma_t dl \quad (12.1)$$

Бұл жерде $\Sigma = n \cdot cs$ — реакция каналының макроэкономикалық көлденен қимасы. cs — реакция каналының микроэкономикалық көлденен қимасы. Осы теңдеуді нормалау шарты

$$\int_0^\infty p(x)dx = 1 \quad (12.2)$$

арқылы шешетін болсақ

$$p(l) = \Sigma_t \exp(-\Sigma_t l) \quad (12.3)$$

аламыз.

Сонымен, шексіз біртекті ортада нейтронның еркін жүру жолы экспоненциалды түрде таралған кездейсоқ шама болып табылады. Олай болса $(0, \infty)$ аралығында жатқан осындай кездейсоқ шаманың мәндерін $(0, 1)$ аралығында біркелкі үлестірілген кездейсоқ сандардың көмегімен алуға болады. Алдағыда біз оларды жай ғана кездейсоқ сандар деп атап γ арқылы белгілейтін боламыз. Енді нейтронның еркін жүру жолын модельдеуге арналған өрнектерін егжей-тегжейлі қарастырмай тек γ арқылы анықталатын соңғы формуланы берейік. Бұл формуланың түрі мынадай:

$$l = -\frac{1}{\Sigma_t} \ln \gamma \quad (12.4)$$

Алгоритм.

Енді мәселемізді шешуге кірісейік. Біз нейтронның траекториясын модельдеуді нейтрон пластинаның бетіне түскен сәттен бастаймыз, яғни $x = 0$ нүктесінен.

(12.4) формуласының көмегімен кездейсоқ еркін жүру жолының ұзындығын l есептеп, оның D-дан аспауын тексереміз.

Егер нейтрон пластинадан өтсе, біз бұл оқиғаны тіркеп, келесі нейтронның траекториясын модельдеуге өтеміз.

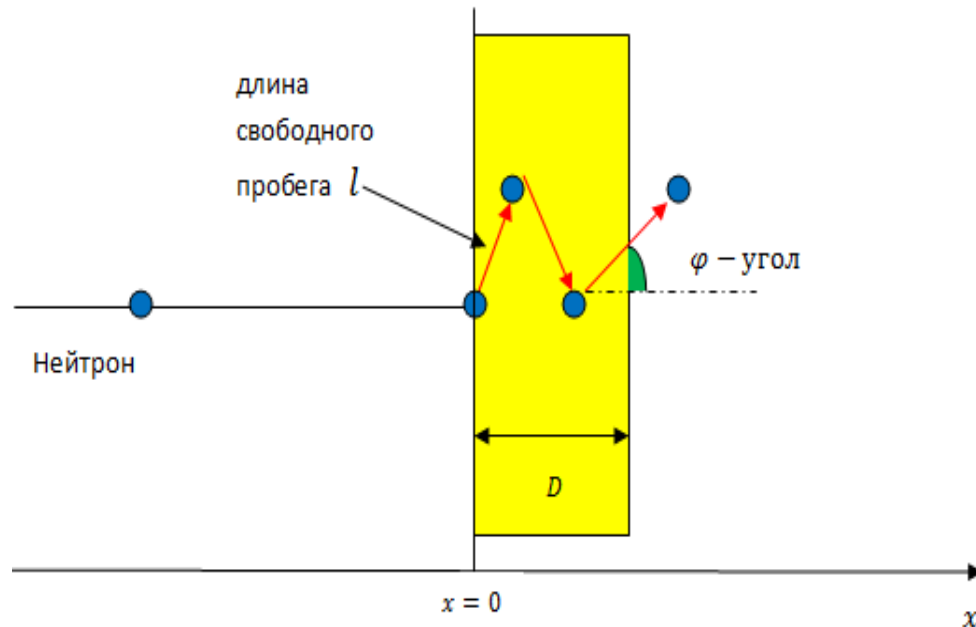
Егер нейтрон пластина шекарасынан шықпаса, онда соқтығысу болды деп есептейміз. Содан кейін random арқылы жаңа кездейсоқ санды қолданып, қандай оқиға болғанын анықтаймыз - шашырау немесе нейтронды жұту (егер $\gamma < \Sigma_s/\Sigma_t$ болса шашырау).

Егер нейтрон жұтылса онда бұл оқиғаны тіркеп жаңа нейтронның траекториясын модельдеуге көшеміз.

Егер шашырау болса, онда шашыраудың жаңа бағытын анықтап, жаңа еркін жүру жолының ұзындығын есептеп табамыз да нейтронның пластина ішінде жатуын тексереміз.

Нейтронның шашырау бұрышын модельдеуге (12.5) формуласымен есептейміз.

$$\cos \varphi = 2\gamma - 1 \quad (12.5)$$



Модельдеу процессін визуализациялау үшін Visual Studio-да «Tao framework» библиотекасын орнатуымыз керек. Орнату туралы ақпаратты мына жерден таба аласыз: <http://esate.ru/uroki/OpenGL/uroki-OpenGL-c-sharp/ystanovka-tao-framework-2-1-0/>

1. «Приложение Windows Forms» типтес жаңа проект құрамыз.
2. форманың атын Form1 ден MainForm деп ауыстырамыз, сонымен қатар қасиеттер терезесінде Text қасиетін Form1 ден Lecture-8 ауыстырамыз.
3. Сонымен қатар форма қасиеттерін мынадай қылдырып береміз Size = 600; 530., MaximizeBox = False., MaximumSize = 600; 530., MinimumSize = 600; 530.
4. Панель элементов → «общее» деген жерде тінтуірдің оң жақ түймесін басып, «выбрать элементы» дегенді таңдаңыз → Компоненты.NET Framework деген жерге барып «Обзор» басыңыз. Taoframework орнатылған қалтаға барып, Taoframework → bin и Tao.Platform.Windows.dll файлын таңдап «Открыть» дегенді басыңыз. Сонда тізімде SimpleOpenGLControl деген пайда болады және сонын қасындағы құсбелгіні таңдап ОК басыңыз.
5. Енді біздің «Панель элементов» деген жерде SimpleOpenGLControl деген «контрол» болады. Оны формаға апарып салыңыз. «окно свойств» деген жерде атын «simpleOpenGLControl1» дегеннен «GLWindow» деп ауыстырыңыздар. Сол жерде оның өлшемдерін 559; 400 деп орнатыңыздар.
6. «обозреватель решений» деген жерге барып «Ссылки» дегеннің үстінен тінтуірдің оң жақ түймесін басып, «добавить ссылку» дегенді таңданыз. Обзор → «обзор». Taoframework орнатылған қалтаға барып, Taoframework → bin и мына (Ctrl) 3 файлдарды таңданыздар Tao.Platform.Windows.dll, Tao.OpenGL.dll және Tao.FreeGlut.dll → «добавить». ОК басыңыздар.
7. «GLWindow» контролының сол жағынан Label орнатыңыздар оның қасиеттерін AutoSize = False, ал Text="Число **налетающих нейтронов**" деп орнатыңыздар. Label-дың оң жағына «TextBox» қойыңыздар да, қасиетінде Text=TextBoxInNumber деп жазыңыздар.

8. 7-ші шарттағыдай создайте два Label типтес тағы екі Label орнатыңыздар Text қасиеті «Число прошедших через пластину» және «Число поглощенных нейтронов» деген сәйкесінше. және тағы екі TextBox орнатыңыздар Text=TxtBoxOutNumber және Text=TxtBoxCapNumber сәйкесінше.

9. BtnClear және BtnStart атты екі түйме орнатыңыздар Text = «Clear» Text = «Start» сәйкесінше. Түймеге екі рет басу арқылы код орналасқан файлға өтіңіздер.

Сол жерде төмендегі кодты орналастырыңыздар:

```
//Генератор случайных чисел.
public static class Rand
{
    private static Random rand = new Random();

    public static double Next()
    {
        return (double)rand.Next(100000000) / 100000000.0;
    }
}

//Класс Нейтрон.
public class Neutron
{
    public double ax;
    public double ay;
    public double bx;
    public double by;

    public Neutron(double ax, double ay)
    {
        this.ax = ax;
        this.ay = ay;
        this.bx = this.ax;
        this.by = this.ay;
    }

    public void
    Next(HomoTarget target, double cosine)
    {
        this.ax = this.bx;
        this.ay = this.by;

        double L = (-1.0 / (0.1 * target.n *
        target.cs)) * (Math.Log(Rand.Next()));
    }
}
```

```

//Класс Однородная среда.
public abstract class HomoTarget
{
    //Концентрация задается в 10^23 ед.
    public double n;
    //Сечение задается в барнах 10^-24 см^2.
    public double cs;

    public abstract void
    GLDraw();
    public abstract bool
    IsInside(double x, double y);
    public abstract bool
    Isdead();
}

//Класс Безграничная по y прямоугольная пластина.
public class YInRectTarget : HomoTarget
{
    private double cs_capture;
    private double cs_elastic;

    public readonly double x;
    public readonly double d;

    public YInRectTarget(double x, double d, double n, double cs_cap,
                        double cs_elc)
    {
        this.n = n;
        this.x = x;
        this.d = d;

        this.cs_capture = cs_cap;
        this.cs_elastic = cs_elc;

        this.cs = cs_cap + cs_elc;
    }

    public override void
    GLDraw()
    {
        // Рисуем пластину
        Gl.glBegin(Gl.GL_QUADS);
        Gl.glColor3d(1.0, 1.0, 0.0);
        Gl.glVertex2d(this.x, -1000.0);
        Gl.glVertex2d(this.x, 1000.0);
        Gl.glVertex2d(this.x + this.d, 1000.0);
    }
}

```

```

public partial class MainForm : Form
{
    private HomoTarget Target;
    public MainForm()
    {
        InitializeComponent(); //Инициализация компонент WinForm.
        this.GLWindow.InitializeContexts(); //Инициализация компонент OpenGL.
        Glut.glutInit();
    }

    private void MainFormLoad(object sender, EventArgs e)
    {
        this.Target = new YInfractTarget(0,5,0.3295,0.712178,10.7736);
        Glut.glutInitDisplayMode(Glut.GLUT_RGB | Glut.GLUT_DOUBLE);
        Gl.glClearColor(0.9f, 0.9f, 0.9f, 1f);
        Gl.glViewport(0, 0, this.GLWindow.Width, this.GLWindow.Height);
        Gl.glMatrixMode(Gl.GL_PROJECTION);
        Gl.glLoadIdentity();
        Glu.gluOrtho2D(-10, 10.0, -10.0, 10.0);
        Gl.glMatrixMode(Gl.GL_MODELVIEW);
        Gl.glClear(Gl.GL_COLOR_BUFFER_BIT);
        this.Target.GLDraw();
    }

    private void BtnStart_Click(object sender, EventArgs e)
    {
        this.GLWindow.Invalidate();    this.MainFormLoad(null, null);
        this.BtnClear.Enabled = false; this.BtnStart.Enabled = false;
        int num_in = 0; int num_out = 0; int num_cap = 0;

        try { num_in = int.Parse(this.TxtBoxInNumber.Text); }
        catch { this.BtnClear.Enabled = true; return; }

        for (int i = 0; i < num_in; i++){
            Neutron neutron = new Neutron(0, 0);
            double cosine = 1.0;
            while (true)
            {
                neutron.Next(this.Target, cosine);
                neutron.GLDraw();
                if (!this.Target.IsInside(neutron.bx, neutron.by))
                { num_out++; break; }
                if (this.Target.Isdead())
                { num_cap++; break; }
                cosine = 2 * Rand.Next() - 1;
            }
        }
        this.TxtBoxCapNumber.Text = num_cap.ToString();
        this.TxtBoxOutNumber.Text = num_out.ToString();
        Gl.glFlush();
        this.GLWindow.Invalidate();
        this.BtnClear.Enabled = true;
    }

    private void BtnClear_Click(object sender, EventArgs e)
    {
        this.BtnClear.Enabled = false;
        this.BtnStart.Enabled = false;
        this.TxtBoxInNumber.Text = "";
    }
}

```

