



C++ DASTURLASH TILIDA STREOMETRIK MASALALAR YECHISH HAMDA ULARNING AMALIYOTGA TATBIQI

O.T.Muhiddinova

*Muhammad AL Xorazmiy nomidagi Toshkent axborot texnologiyalari
universiteti m.f. dotsenti, Phd*

K.X. Voxidov

*Muhammad AL Xorazmiy nomidagi Toshkent axborot texnologiyalari
universiteti 1-bosqich bakalavri*

Anotatsiya: *Malumki bugungi kunda Phyton, Java, Java S, Delphi kabi ko'plab dasturlash tillari keng ko'lamda foydalanib kelinmoqda. Biroq ushbu dasturlash tillarining asoslari bo'lmish C hamda C++ tillari ham amaliyotda o'z mavqeyini yo'qotmagan. Ushbu ilmiy maqolamiz orqali biz C++ tilida maktab o'quvchilari uchun biroz qiyin bo'lgan streometrik masalalarni C++ tilida yechish va ushbu dasturni tuzishni yoritmoqchimiz. Streometriyada o'quvchilarni asosan qiynadigan mavzu ayqash to'g'ri chiziqlar orasidagi burchakni topish masalasiga asosiy e'tiborni qaratamiz.*

Tayanch so'zlar: *C++ dasturlash tili, streometrik masalalarni yechishdagi muammolar, ayqash to'g'ri chiziqlar, ayqash tog'ri chiziqlar orasidagi burchaklar, koordinatsion metod yordamida streometrik masalalarni yechish, C++ dasturlash tilidagi matematik funksiyalar va ularning vazifalari.*

KIRISH

Matematika amaliyotda juda ko'p masalalarni yechishda yordam beradi. Iqtisodiyot, moliya, arxitektura va qurulish, informatsion texnologiyalarni algoritmlashtirish shular jumlasidandir. Biz bu maqolaning mavzusini tanlash uchun maktab, kollej va akademik litseylarda o'quvchi va talaba yoshlar o'rtasida so'rovnoma olib bordik. So'rovnoma natijalariga ko'ra o'quvchi abituriyentlarni asosan matematikada streometriya masalalari muammo tug'dirishi ayrimlarda fazoviy tasavvur kamligi uchun amaliyotda uchraydigan streometrik muammolarni hal qilishga biroz o'ylanishadi. Biz maqolada yoritmoqchi bo'lgan mavzu esa aynan shunday fazoviy taasuroti yaxshi rivojlanmagan yoki streometriyaga qiynaladigan ammo kasbi uchun geometriya suv va havodek kerak bo'ladigan insonlar uchun. Maqolamizni soddaroq qilish uchun biz streometrik jismlardan namuna sifatida parallelepipedni tanlab oldik va ushbu parallelepipedda 24 ta to'g'ri chiziq yani 12 ta holat uchun masalalarga dastur tuzdik. Dasturni matematik modellashtirish jarayonida maktab o'quvchisi uchun uncha yaxshi bo'lmagan koordinatsion metoddan foydalandik. Bu metod maktab programmasida yo'q ekanligini hisobga olib maqolamizda yana bir ommalashmagan metoddan



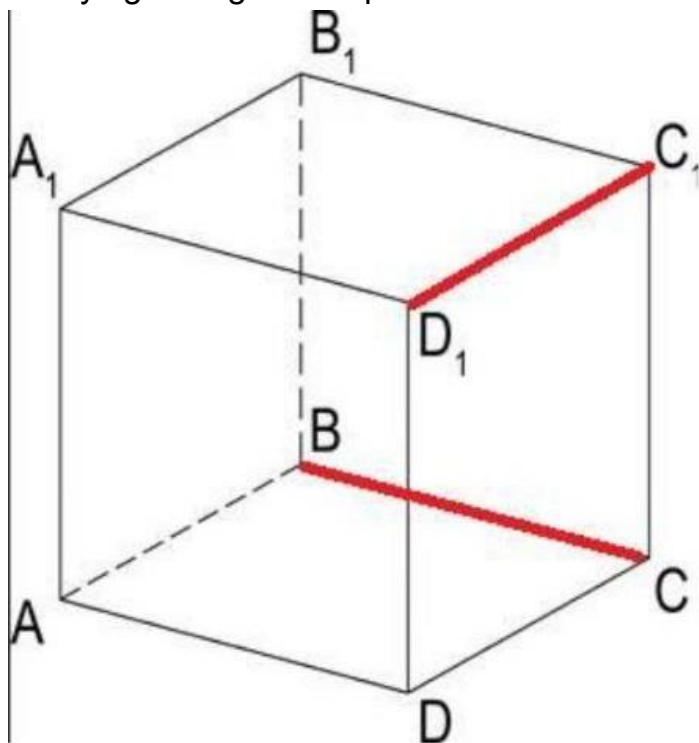
foydalandik. Agarda ayqash to'g'ri chiziqlar uchun geometrik yo'ldan borsak ishimiz ancha qiyinlashadi. Chunki ushbu metod qo'shimcha chizma va hisoblashlarni talab etadi va yakunda dasturning sifatiga ta'sir o'tkazishi mumkin.

Streometrik jismni 3 o'lchamli koordinataga qo'yish ham aslida o'quvchidan chizmachilik ko'nikmalarini talab qiladi. Biroq bu ish qo'shimcha chizmalar o'tkazishdan yengilroq. Ayqash to'g'ri chiziqlar bilan ishlash arxitektura sohasida ayniqsa qo'l keladi. Proektning turli qismlarida joylashgan to'g'ri chiziqlar orasidagi bog'liqlikni topishda bu dastur qo'l keladi.

ASOSIY QISIM

Ayqash to'g'ri chiziqlar degan termini ko'p ishlatdik. Keling dastlab ayqash to'g'ri chiziqlar bilan vazifa bajarishdan oldin ularga ta'rif beramiz.

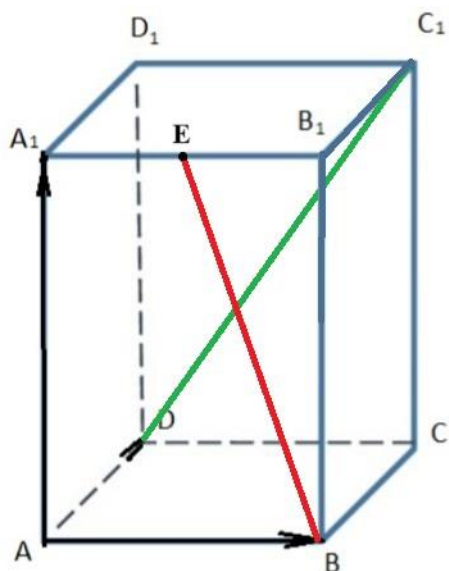
Ayqash to'g'ri chiziqlar-turli tekisliklarda yotuvchi va o'zaro kesishmaydigan to'g'ri chiziqlar.



1-rasm. Ayqash to'g'ri chiziqlar .

Endi esa parallelepipedda ayqash to'g'ri chiziqlar orasidagi burchakni koordinatsion yo'l bilan topishni ko'ramiz.

Masala. Bizga $ABCD A_1 B_1 C_1 D_1$ parallelepiped berilgan bo'lsin. Parallelepipedning balandligi 5 ga asosidagi to'g'ri to'rtburchakning bo'yi 4 ga eni esa 3ga teng. BE va DC_1 to'g'ri chiziqlar orasidagi burchakni toping. (E nuqta $A_1 B_1$ tomon o'rtasi).



AA₁ tomonni oz o'q deb AB tomonni OX o'q deb, DC tomonni esa Oy o'q deb qabul qilamiz. U holda har bir nuqta o'z koordinatasiga ega bo'ladi. Ya'ni A(0,0,0), B(0;4;0), D(3;0;0) C(3;4;0), A₁(0;0;5), B₁(0;4;5), C₁(3;4;5), U holda E nuqtaning koordinatasi (E(x,y,z));

$$X=(X_{a1}+X_{b1})/2; Y=(Y_{a1}+Y_{b1})/2; Z=(Z_{a1}+Z_{b1})/2;$$

$$x=0; y=2; z=5; . U holda BE vector koordinatalri$$

$$BE(0-0;2-4;5-0) \text{ yani } BE(0;-2;5). DC_1 \text{ ning koordinatalari esa } DC_1(0;4;5);$$

Ikki vektor orasidagi burchakni topish formulasini eslatib o'tamiz;

A burchak bo'lsin. Demak ;

$$A=\arccos((BE,DC_1)/ |BE|*|DC_1|)$$

$$(BE,DC_1)=0*0+(-2)*4+5*5=17; |BE| = \sqrt{(0+4+25)} = \sqrt{29} ; |DC|= \sqrt{49}$$

U holda

$$A=\arccos(17/ \sqrt{(41*29)})=60.4919^\circ \text{ demak Javob: } 60.4919^\circ$$

Ko'rib turganingizdek juda ham qiyinchilik tug'ulmadi ana endi esa shu masala uchun C++ dasturlash tilida kod yozib dastur tuzaylik.

```
#include<iostream>// kutubxona
#include<cmath>// matematik amala bajaruvchi kutubxona
using namespace std;// fazo
int main() { // bosh funksiya
int k;
double
AA1, BB1, CC1, DD1, AB, AD, DC, BC, A1B1, A1D1, B1C1, D1C1, BE;
//
parallelepiped tomonlari;
float
A[3],
B[3], C[3], D[3], A1[3], B1[3], C1[3], D1[3], E[3], E1[3], E2[3], E3[3], E4[3], E5[3], E6[3]
, E7[3], E8[3], E9[3], E10[3], E11[3], L[3], K[3]; // har bir nuqtani massiv ko'rinishida
oldik;
string s1, s2; // ushbu qatorlar funksiyasidan biz o'zimizga kerakli ikkita
to'g'ri chiziqni kirgazish uchun oldik;
```



```
cout<<"To'gri burchakli parallelepipeddagi ikki ayqash to'g'ri chiziq
orasidagi burchakni topish"<<endl;
cout<<"\nBalandlikni kiriting: ";
cin>>AA1;// balandlik ni kirgazish;
cout<<"\nEnini kiriting: ";
cin>>AD;// asos enini kirgazmoqdamiz;
cout<<"\nBo'yini kiriting: " ;
cin>>AB;// asos boyini kirgazmoqdamiz;
cout<<"\nDemak tassavur qiling sizning parallapipedingiz
ABCD A1B1C1D1 parallapiped va u holda: "<<endl;
cout<<"\nE ,E1,E2,E3,E4,E5,E6,E7, E8, E9, E10,E11 mos ravishada
A1B1, A1D1, B1C1, D1C1, AB, AD, BC, DC, AA1, BB1, CC1, DD1
tomonlarning o'rtasi"<< endl;
cout<<"\nAA1=BB1=CC1=DD1="<<AA1;
cout<<"\nAD=BC=A1D1=B1C1="<<AD;
cout<<"\nAB=DC=A1B1=D1C1="<<AB;
cout<<"\nKerakli to'g'ri chiziqni tanlang: "<<endl;
cout<<"\nbirinchi togri chiziq : " ;
AA1==BB1==CC1==DD1;
AD==BC==A1D1==B1C1;
AB==DC==A1B1==D1C1;// ushbu holatda tomonlarning moslarini bir
biriga teng ekanligi ko'rsatilmoqda;
cin>>s1;
cout<<"\nikkinchi to'gri chiziq: ";
cin>>s2; // 1- va 2- to'g'ri chiziqlar kirgizilyapti;
A[3]==(0,0,0);
B[3]==(0,AB,0);
D[3]==(AD,0,0);
C[3]==(AD,AB,0);
A1[3]==(0,0,AA1);
B1[3]==(0,AB,AA1);
C1[0]==AD;
C1[1]==AB;
C1[2]==AA1;
D1[3]==(AD,0,AA1);
E[0]==0;
E[1]==AB/2;
E[2]==AA1;
E1[3]==((A1[0]+D1[0])/2,(A1[1]+D1[1])/2, (A1[2]+D1[2])/2);
E2[3]==((B1[0]+C1[0])/2,(B1[1]+C1[1])/2,(B1[2]+C1[2])/2);
E3[3]==((D1[0]+C1[0])/2,(D1[1]+C1[1])/2,(D1[2]+C1[2])/2);
```



```
E4[3]=((A[0]+B[0])/2,(A[1]+B[1])/2,(A[2]+B[2])/2);
E5[3]=((A[0]+D[0])/2,(A[1]+D[1])/2,(A[2]+D[2])/2);
E6[3]=((B[0]+C[0])/2,(B[1]+C[1])/2,(B[2]+C[2])/2);
E7[3]=((D[0]+C[0])/2,(D[1]+C[1])/2,(D[2]+C[2])/2);
E8[3]=((A1[0]+A[0])/2,(A1[1]+A[1])/2,(A1[2]+A[2])/2);
E9[3]=((B1[0]+B[0])/2,(B1[1]+B[1])/2,(B1[2]+B[2])/2);
```

// Biz tepadagi holatda massiv elementlaridan foydalanib koordinatalarni aniqlab oldik. Dasturimiz tushunarlik bo'lishi uchun.

```
if(s1=="BE"&& s2=="DC1") {
    k=1;
}
else if(s1=="BE2"&& s2=="A1D") {
    k=2;
}
else if(s1=="DE3"&& s2=="A1B") {
    k=3;
}
else if(s1=="AE1"&& s2=="BC1") {
    k=4;
}
else if(s1=="AE6"&& s2=="DC1") {
    k=5;
}
else if(s1=="A1B"&& s2=="DC1") {
    k=6;
}
else if(s1=="AC"&& s2=="DC1") {
    k=7;
}
else if(s1=="B1E4"&& s2=="D1C") {
    k=8;
}
else if(s1=="B1E5"&& s2=="D1C") {
    k=9;
}
else if(s1=="A1E3"&& s2=="D1C") {
    k=10;
}
else if(s1=="D1E5"&& s2=="B1C") {
    k=11;
}
```



```
else if(s1=="D1E7"&& s2=="B1C") {
k=12;
}
else {
cout<<"dasturdagi kamchlik uchun uzur so'raymiz";
}
cout<<"\n bu ikki to'g'ri chiziq orasidagi burchak : ";
if shart operatori yordamida biz 12 ta holat uchun qaysidir ma'noda asos
yasab oldik;
switch(k)
{
case 1:
y=(((AB/2-AB)*AB+AA1*AA1)/(sqrt((-AB/2)*(-
AB/2)+AA1*AA1))*sqrt(0+AB*AB+AA1*AA1));
cout<<acos(y)*180/3.14<<"°"; // o'quvchi ilk bor tanishishi mumkin bo'lgan
unksiya accos funksiyasidr. Ushbu funksiya -1 dan 1 gacha bo'lgan oraliq
uchun ishlaydi. Aks holda nan qiymat qaytaradi. Funksiyaning vazifasi biror bir
o'zgaruvchining arccos ni topishdir. Asosan double, long double tiplar bilan
ishlaydi.
break;
case 2:
y=((AD/2)*AD+0-
AA1*AA1)/(sqrt((AD/2)*(AD/2)+0+AA1*AA1))*sqrt(AD*AD+0+AA1));
cout<<acos(y)*180/3.14<<"°";
break;
case 3:
y=((AB*AB+AA1*AA1)/(sqrt((-AD/2)*(-
AD/2)+AB*AB+AA1*AA1))*sqrt(AB*AB+AA1*AA1));
cout<<acos(y)*180/3.14<<"°";
break;
case 4:
y=((0+0+AA1*AA1)/(sqrt(0+(AB/2)*(AB/2)+AA1*AA1))*sqrt(AD*AD+0+AA
1*AA1));
cout<<acos(y)*180/3.14<<"°";
break;
case 5:
y=((AD/2)*0+AB*AB)/(sqrt((AD/2)*(AD/2)+AB*AB))*sqrt((0+AB*AB+AA1*
AA1));
cout<<acos(y)*180/3.14<<"°";
case 6:
```



```
y=(-
AB*AB+AA1*AA1)/(sqrt(0+AB*AB+AA1*AA1)*sqrt(0+AB*AB+AA1*AA1));
cout<<acos(y)*180/3.14<<"°";
break;
case 7:
y=((AD*0+AB*AB+0)/(sqrt(AD*AD+AB*AB)*sqrt(AA1*AA1+AB*AB)));
cout<<acos(y)*180/3.17<<"°";
break;
case 8:
y=((0+AB/2*AB+0)/(sqrt(0+AB*AB+AA1*AA1)*sqrt(0+AB/2*AB/2)));
cout<<acos(y)*180/3.14<<"°";
break;
case 9:
y=((AD/2*0-AB*AB-
AA1*AA1)/(sqrt((AD/2)*(AD/2)+AB*AB+AA1*AA1)*sqrt(0+AB*AB+AA1*AA1)
));
cout<<acos(y)*180/3.14<<"°";
break;
case 10:
y=((0-
AB*AB/2+0)/(sqrt(AA1*AA1+AB*AB)*sqrt(AD*AD+(AB/2)*(AB/2))));
cout<<acos(y)*180/3.14<<"°";
break;
case 11:
y((((-AD/2)*AD+0+AA1*AA1)/(sqrt((-AD/2)*(-
AD/2)+0+AA1*AA1)*sqrt(AD*AD+AA1*AA1)));
cout<<acos(y)*180/3.14<<"°";
break;
case 12:
y((((0+0+AA1)/(sqrt(0+(-AB/2)*(-
AB/2)+AA1*AA1)*sqrt(AD*AD+AA1*AA1+0))));
cout<<acos(y)*180/3.14<<"°";
break;
}
return 0;}
```

Endi esa ushbu dastur natijalaini tadim etamiz.



To'g'ri burchakli parallapipeddagi ikki ayqash to'g'ri chiziq orasidagi burchakni topish

Balandlikni kiriting: 5

Enini kiriting: 3

Bo'yini kiriting: 4

Demak tassavur qiling sizning parallapipedingiz ABCDA1B1C1D1 parallapiped va u holda:

E ,E1,E2,E3,E4,E5,E6,E7, E8, E9, E10,E11 mos ravishda A1B1, A1D1, B1C1, D1C1, AB, AD, BC, DC, AA1, BB1, CC1, DD1 tomonlarning o'rtasi

AA1=BB1=CC1=DD1=5

AD=BC=A1D1=B1C1=3

AB=DC=A1B1=D1C1=4

Kerakli to'g'ri chiziqni tanlang:

birinchi to'g'ri chiziq :BE

ikkinchi to'g'ri chiziq: DC1

bu ikki to'g'ri chiziq orasidagi burchak : 60.4919°

Ko'rib turganingizdek natijalar bir xil.

XULOSA

Bugungi kunda C++ tilida yangi dasturlar ishlatilmoqda. Ammo ular ichida matematik, yoki matematik masalalarni hal qiluvchi dasturlar kam. Boshqa dasturlash tillarida yozilgan dasturlar ham streometrik masalalarni yechishda xatolarga yo'l qo'yishi mumkin. Ammo bugun biz maqolamizda ko'rib chiqqan dasturimiz esa streometriya, va talabalarni qiynagan ayqash to'g'ri chiziqlar orasidagi burchaklarni topishda bir muncha xatolardan holi. Dastur bajarilish asnosida har bir harakatini har bir qadamni va belgilshlarni foydalanuvchiga ko'rsatib tushuntirib o'tadi. Biz yaratgan dasturimiz nafaqat C++ tilida balki boshqa dasturlash tillarida ham qo'llanilishi va asos bo'lishi mumkin. Amaliyotda ushbu dastur o'z nomi bilan streometriyadagi qiyin murakkab masalalarni osonlashtiradi, arxitekturada va qurulish jarayonida quruvchi injinrlarning vaqtini tejaydi va yuqori aniqlikdagi ko'rsatkich bilan ish samaradorligini oshiradi. Koordinatsion metod orqali nafaqat qurulish balki trassalardagi mashinalar harakatini ham nazorat qilish va analitik tahlil qilish mumkin. Dasturimiz koordinatsion metodga asoslangan ekan demak ushbu dastur orqali 3 o'lchamli fazodagi hodisalarni tahlil qilishimiz mumkin. Biz bu dasturni "SAVOSTreometric" deya nomladik.

FOYDALANILGAN ADABIYOTLAR:

1. Sharigin "Учебное пособие по геометрии"// Moskva-1994;
2. T.N Qori Niyoziy "Analitik geometriya"// O'qituvchi-1970;
3. Gilbert Shildt "Справочник по C++" // VILYAMS; Москва- 2006

FOYDALANILGAN SAYTLAR:

1. <https://www.guru999.com;>