

Teorema 8.1 *Rastojanje tačke M od prave p koja je zadata svojom tačkom P i vektorom pravca $\vec{p}$ dato je jednačinom*

$$d = \frac{|\vec{p} \times \vec{PM}|}{|\vec{p}|}.$$

Ova formula bez izmena važi i u prostoru $\mathbb{E}^3$.

Površina $P_\diamond$ paralelograma razapetog vektorima $\vec{p}$ i $\vec{PM}$ jednaka je normi vektorskog proizvoda tih vektora. Sa druge strane ta je površina jednaka proizvodu dužine osnove $|\vec{p}|$ i visine d paralelograma. Dakle važi,

$$|\vec{p} \times \vec{PM}| = P_\diamond = |\vec{p}| \cdot d,$$

odakle sledi tražena formula. $\square$

Teorema 8.2 *Rastojanje tačke $M(x_0, y_0)$ od prave $p : ax + by + c = 0$ dato je formulom*

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}.$$

Odredićemo jednačinu normale n iz tačke M na pravu p . Vektor normale je $\vec{n} = (a, b)$ tako da je svaka tačka normale data sa

$$x = x_0 + ta, \quad y = y_0 + tb,$$

za neko $t \in \mathbb{R}$ (pri tome se za $t = 0$ dobija tačka M). Presečna tačka N pravih n i p dešava se u trenutku t koji je rešenje jednačine:

$$a(x_0 + ta) + b(y_0 + tb) + c = 0,$$

odakle je

$$t = -\frac{ax_0 + by_0 + c}{a^2 + b^2}.$$

Kako je predjeni put jednak proizvodu intenziteta brzine i vremena dobijamo:

$$d = V|t| = |\vec{n}| |t| = \sqrt{a^2 + b^2} \left| \frac{ax_0 + by_0 + c}{a^2 + b^2} \right| = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}},$$

što je i trebalo dokazati. $\square$

8.2 Preseci pravih, polupravih i duži

8.2.1 Presek pravih zadatih parametarski

Pretpostavimo da su date dve prave: prava p tačkom P i vektorom $\vec{p}$ i prava q tačkom Q i vektorom $\vec{q}$. Cilj nam je da odredimo presek tih dveju pravih, ako postoji.

Tačke $M = P + t \vec{p}$ i $N = Q + s \vec{q}$ za $t, s \in \mathbb{R}$ predstavljaju proizvoljne tačke pravih p i q , redom. U presečnoj tački važi: $P + t \vec{p} = Q + s \vec{q}$, odnosno

$$t \vec{p} - s \vec{q} = Q - P = \vec{PQ}.$$

Množenjem prethodne jednačine vektorski sa desne strane vektorima $\vec{p}$ i $\vec{q}$ dobijamo:

$$s(\vec{p} \times \vec{q}) = \vec{PQ} \times \vec{p}, \quad t(\vec{p} \times \vec{q}) = \vec{PQ} \times \vec{q}. \quad (30)$$

Rezultati svih ovih vektorskih proizvoda imaju prve dve koordinate jednake nuli, pa dobijamo uslov samo iz trećih koordinata. Da bismo to zapisali označimo (kao u poglavlju 4.2.2) za vektore $\vec{p} = (p_x, p_y)$ i $\vec{q} = (q_x, q_y)$ treću koordinatu njihovog vektorskog proizvoda sa

$$D(\vec{p}, \vec{q}) := p_x q_y - p_y q_x.$$

Sada su jednačine (30) ekvivalentne sa

$$sD(\vec{p}, \vec{q}) = D(\vec{PQ}, \vec{p}), \quad tD(\vec{p}, \vec{q}) = D(\vec{PQ}, \vec{q}).$$

Razlikujemo sledeće slučajeve:

a) Kada je $D(\vec{p}, \vec{q}) \neq 0$, tj. vektori $\vec{p}$ i $\vec{q}$ nisu kolinearni, **postoji jedinstvena presečna tačka** koja se dobija ma kojim od parametara

$$s = \frac{D(\vec{PQ}, \vec{p})}{D(\vec{p}, \vec{q})} \quad \text{ili} \quad t = \frac{D(\vec{PQ}, \vec{q})}{D(\vec{p}, \vec{q})}.$$

b) Kada je $D(\vec{p}, \vec{q}) = 0$ i $D(\vec{PQ}, \vec{p}) = 0$, prave imaju isti pravac i tačka Q pripada pravoj p , pa se **prave poklapaju**.

c) Kada je $D(\vec{p}, \vec{q}) = 0$ i $D(\vec{PQ}, \vec{p}) \neq 0$, prave imaju isti pravac, ali tačka Q ne pripada pravoj p , pa su **prave paralelne**.

8.2.2 Presek duži, polupravih i pravih

Kada su duži AB i CD date svojim temenima najzgodnije ih je zapisati parametarski (vidi poglavlje (4.1.4)) tj.

$$M = A + t \vec{AB}, \quad \text{odnosno} \quad N = C + s \vec{CD},$$

za $0 \leq t, s \leq 1$. Slično kao kod preseka dve prave imamo tri slučaja.

a) Kada je $D(\vec{AB}, \vec{CD}) \neq 0$ postoji jedinstvena presečna tačka pravih AB i CD . Potrebno je još proveriti da li ta ta tačka pripada obema dužima, tj. da li je $0 \leq t \leq 1$ i $0 \leq s \leq 1$.

b) Kada je $D(\vec{AB}, \vec{CD}) = 0$ i $D(\vec{AC}, \vec{CD}) = 0$, prave AB i CD se poklapaju. Pitanje, koliko se, i da li se duži AB i CD poklapaju zahteva dodatnu analizu.

c) Kada je $D(\vec{AB}, \vec{CD}) = 0$ i $D(\vec{AC}, \vec{CD}) \neq 0$ duži AB i CD su disjunktne. Poluprava $[AB)$ zadata temenom A i tačkom B je parametarski data sa

$$M = A + t \vec{AB},$$

za $t \geq 0$. Zato pri određivanju preseka poluprave sa drugom polupravom, duži ili pravom treba proveriti ispunjenost uslova $t \geq 0$, nakon što se parametar t izračuna. Slično se razmišlja u ostalim kombinacijama.

8.3 Presek pravih zadatih implicitno

Neka su prave p i q zadate implicitno

$$p : ax + by + c = 0, \quad q : a_1x + b_1y + c_1 = 0.$$

Jedna mogućnost je da se formulama (14) prave predstavljamo parametarski, a presek zatim odredi kao u poglavlju (8.2.1).

Druga mogućnost je da se primeni Kramerovo pravilo. Označimo determinante:

$$\Delta = \begin{vmatrix} a & b \\ a_1 & b_1 \end{vmatrix}, \quad \Delta_x = \begin{vmatrix} c & b \\ c_1 & b_1 \end{vmatrix}, \quad \Delta_y = \begin{vmatrix} a & c \\ a_1 & c_1 \end{vmatrix}.$$

Ako je $\Delta \neq 0$ postoji jedinstveno (x_0, y_0) dato sa

$$x_0 = \frac{\Delta_x}{\Delta}, \quad y_0 = \frac{\Delta_y}{\Delta}.$$

Ako je $\Delta = \Delta_x = \Delta_y = 0$ prave se poklapaju. U preostalim slučajevima prave su paralelne.

8.3.1 O aritmetici u pokretnom zarezu

Račun u pokretnom zarezu u najvećem broju slučajeva (naročito prilikom množenja, deljenja, korenovanja...) dovodi do grešaka. U ovom poglavlju navodimo primere kako te male greške mogu dovesti do pogrešnih zaključaka ili grešaka u toku izvršavanja programa.

U slučaju paralelnih pravih p i q , umesto da bude nula, vrednost $D(\vec{p}, \vec{q})$ biće veoma mala, ali uglavnom različita od nule. To dovodi do toga da naš algoritam te prave proglasi pravama koje se seku. Da stvar bude gora, kad pokušamo da odredimo presek, deleći sa veoma malim brojem $D(\vec{p}, \vec{q})$ lako dolazi do greške prekoračenja.

Standardno rešenje ovog problema je da se predefiniše dozvoljena greška $\epsilon > 0$ i ako važi

$$|D(\vec{p}, \vec{q})| < \epsilon, \quad (31)$$

smatramo da je $|D(\vec{p}, \vec{q})|$ jednako nuli. Ovo rešenje primenjujemo za sva poredjenja sa nulom promenljivih dobijenih računom u pokretnom zarezu.

Navedeni problem nije jedini. Recimo, neka je $\epsilon = 0.00001$ i prave p i q ortogonalne, tj. takve da je $\sin \angle(\vec{p}, \vec{q}) = 1$. Može se desiti da su vektori $\vec{p}$ i $\vec{q}$ malih dužina (naprimer, da su određeni veoma bliskim tačkama), recimo $|\vec{p}|, |\vec{q}| \leq 0.001$. Odatle je

$$|D(\vec{p}, \vec{q})| = |\vec{p} \times \vec{q}| = |\vec{p}| |\vec{q}| \sin \angle(\vec{p}, \vec{q}) \leq 0.001 \cdot 0.001 = 0.000001 < \epsilon,$$

odakle bi zaključili da su prave paralelne!!!

Jedno rešenje ove vrste problema je da se za vektore pravih uvek uzimaju normalizovani vektori (poglavlje 4.1.1).

Drugo moguće rešenje je da umesto kriterijuma (31) koristi kriterijum

$$|D(\vec{p}, \vec{q})| < \epsilon |\vec{p}| |\vec{q}|, \quad (32)$$

u kom se uzimaju u obzir dužine vektora $\vec{p}$ i $\vec{q}$. On je ekvivalentan metodi upotrebe normalizovanih vektora.

Pošto se u kriterijumu (32) računaju korenovi, najčešće se koristi ekvivalentan uslov:

$$|D(\vec{p}, \vec{q})|^2 < \epsilon^2 |\vec{p}|^2 |\vec{q}|^2.$$

8.4 Pseudokod funkcije koja računa presečnu tačku dve prave

Podrazumevamo da je tip `Point` ima dve koordinate: x i y . Njega koristimo i za tačke i za vektore.

```
int NadjiPreseke(Point P, Point pp, Point Q, Point qq, Point& X)
{
    epsilon2 = 0.000001;
    float s, pq2, dpqpp;
    Point pq = Q-P;
    float dppqq = pp.x * qq.y - pp.y * qq.x;
    float dppqq2 = dppqq * dppqq;
    float pp2 = pp.x * pp.x + pp.y * pp.y;
    float qq2 = qq.x * qq.x + qq.y * qq.y;
    if (dppqq2 > epsilon2 * pp2 * qq2){
        // prave se seku
        s = (pq.x * p.y - pq.y * p.x) / dppqq;
        X = P + s * pp;
        return 1;
    }
    // prave su paralelne ili se poklapaju
    pq2 = pq.x * pq.x + pq.y * pq.y;
    dpqpp = pq.x * pp.y - pq.y * pp.x;
    dpqpp2 = dpqpp * dpqpp;
    if (dpqpp2 > epsilon2 * pq2 * pp2){
        // prave su paralelne
        return 0;
    }
    // prave se poklapaju
    return 2;
}
```

8.1 Napisati funkcije **ParUEksp** i **EkspUPar** koje prevode parametarski oblik prave u eksplicitni i obrnuto.

8.2 Napisati C funkciju **NadjiPresekePar** koja vraća vrednost integer, a kao argumente uzima tačke i vektore pravca dveju pravih i pokazivač presečne tačke X . Funkcija treba da vrati 0 ako nema preseka, 1 ako postoji jedinstvena presečna tačka X i 2 ako se prave poklapaju. Ukoliko postoji jedinstvena presečna tačka, u X treba upisati njene koordinate.

Testirati funkciju na primerima iz skripte i sopstvenim primerima. Posebnu pažnju posvetiti specijalnim slučajevima.

8.3 Napisati C funkciju **NadjiPresekeImp** kao u prethodnom zadatku, ali za prave zadate implicitno njihovim koeficijentima a, b, c, a_1, b_1, c_1 .

9 Interesantni problemi u ravni

Za rešavanje problema koji slede koristićemo jezik "C" i u njemu strukture Tacka, Prava, Krug i Duz. Primetite da strukturu Tacka koristimo i za tacke i za vektore pošto su oni predstavljeni na isti način - dvema koordinatama.

```
typedef struct {
    float x;
    float y;
} Tacka;
```

```
typedef struct {
    Tacka start;
    Tacka pravac;
} Prava;
```

```
typedef struct {
    Tacka centar;
    float r;
} Krug;
```

```
typedef struct {
    Tacka start;
    Tacka end;
} Duz;
```

Napomena: funkcije u sledećim zadacima vraćaju rezultat po adresi koja je poslednji argument. Po pravilu, ako je taj rezultat jedinstven, funkcija je tipa `void`. Ako rezultat ne mora biti jedinstven biti funkcija je tipa `int` i ona vraća broj rešenja. Ta rešenja se upisuju niz, čiji je pokazivač (tj. adresa prvog člana niza) poslednji argument funkcije.

9.1 a) Napisati funkciju

```
void Normala ( Tacka M, Prava p, Prava *n)
```

koja računa normalu n iz tačke M na pravu p .

b) Napisati funkciju

```
void PodnozjeNormale ( Tacka M, Prava p, Tacka *N)
```

koja računa podnozje normale N iz tačke M na pravu p .

9.2 Deklarisana je promenljiva presek kao niz tačaka dužine 2, tj. sa `Tacka presek[2]`. Napisati funkciju

```
int PresekPravaKrug ( Krug k, Prava p, Tacka presek)
```

koja presečne tačke kruga k i prave p upisuje u niz tačaka `presek` (tj. $T[0] = \text{presek}_1, T[1] = \text{presek}_2$, ako one postoje). Celobrojna vrednost funkcije treba da bude 0, 1 ili 2 u zavisnosti od broja presečnih tačaka.

9.3 Napisati funkciju

```
void PravaPodUglom ( Prava p, Tacka Q, float alfa, Prava *q)
```

koja vraća pravu q koja sadrži datu tačku Q , a sa datom pravom p zaklapa dati orjentisani ugao α dat u stepenima.

9.4 Napisati funkciju

```
void PravaNaRastojanju ( Prava p, float d, Prava q)
```

koja vraća niz od dve prave q_1 i q_2 koje su paralelne pravoj p i na rastojanju $d \neq 0$ od nje.

9.5 Napisati funkciju

```
int PresekKrugova ( Krug k1, Krug k2, Tacka presek)
```

koja upisuje presečne tačke krugova k_1 i k_2 u niz tačaka presek. Funkcija treba da vrati broj presečnih tačaka ta dva kruga.

9.6 Napisati funkciju

```
int UpisanKrug ( Tacka A, Tacka B, Tacka C, Krug *k)
```

koja vraća krug k upisan oko trougla ABC . Funkcija treba da vrati vrednost 1 ako krug postoji, tj. ako su tačke nekolinearne i 0, ako su tačke kolinearne, pa krug ne postoji.

9.7 Napisati funkciju

```
int OpisanKrug ( Tacka A, Tacka B, Tacka C, Krug *k)
```

koja vraća krug k opisan oko trougla ABC . Funkcija treba da vrati vrednost 1 ako krug postoji, tj. ako su tačke nekolinearne i 0, ako su tačke kolinearne, pa krug ne postoji.

9.8 Napisati funkciju

```
int Tangenta ( Tacka A, Krug k, Prava tangenta)
```

koja vraća niz tangenti na opisan krug k iz tačke A . Funkcija treba da vrati broj tangenti.

;

9.9 Napisati funkciju

```
int ZajednickeTangente ( Krug k1, Krug k2, Prava tangenta)
```

koja vraća niz zajedničkih tangenti krugova k_1 i k_2 (može ih biti najviše 4). Funkcija treba da vrati broj tangenti.

9.10 Napisati funkciju

```
int Krug2tackeR ( Tacka A, Tacka B, float r, Krug k)
```

koja vraća niz krugova koji sadrže tačke A i B i imaju poluprečnik r . Funkcija treba da vrati broj takvih krugova (najviše 2).

9.11 Napisati funkciju

```
int KrugTangentaTackaR ( Prava t, Tacka A, float r, Krug k)
```

koja vraća niz krugova koji sadrže tačke A , dodiruju pravu t i imaju poluprečnik r . Funkcija treba da vrati broj takvih krugova.