11. Základy výpočetní geometrie

BI-EP1
Efektivní programování 1

Katedra teoretické informatiky
Fakulta informačních technologií
České vysoké učení technické v Praze

ZS 2018/2019
Ing. Martin Kačer, Ph.D.

Výpočetní geometrie

- Algoritmy pracující s objekty v prostoru
- Obvykle Eukleidovský prostor
  - 2D
  - 3D
  - 4D a více ???
- Topologie
- …

Využití výpočetní geometrie

- Počítačová grafika
- Modelování
- Vizualizace
- Robotika
- Geografické systémy

(dnes ale ještě nic z toho)

Geometrie x Algoritmy

- Výpočetní geometrie samozřejmě staví na geometrii jako takové
  - Pravidla a poučky platí …
  - … a opravdu se používají

- Algoritmizace ale může být problém
  - **Proč?**

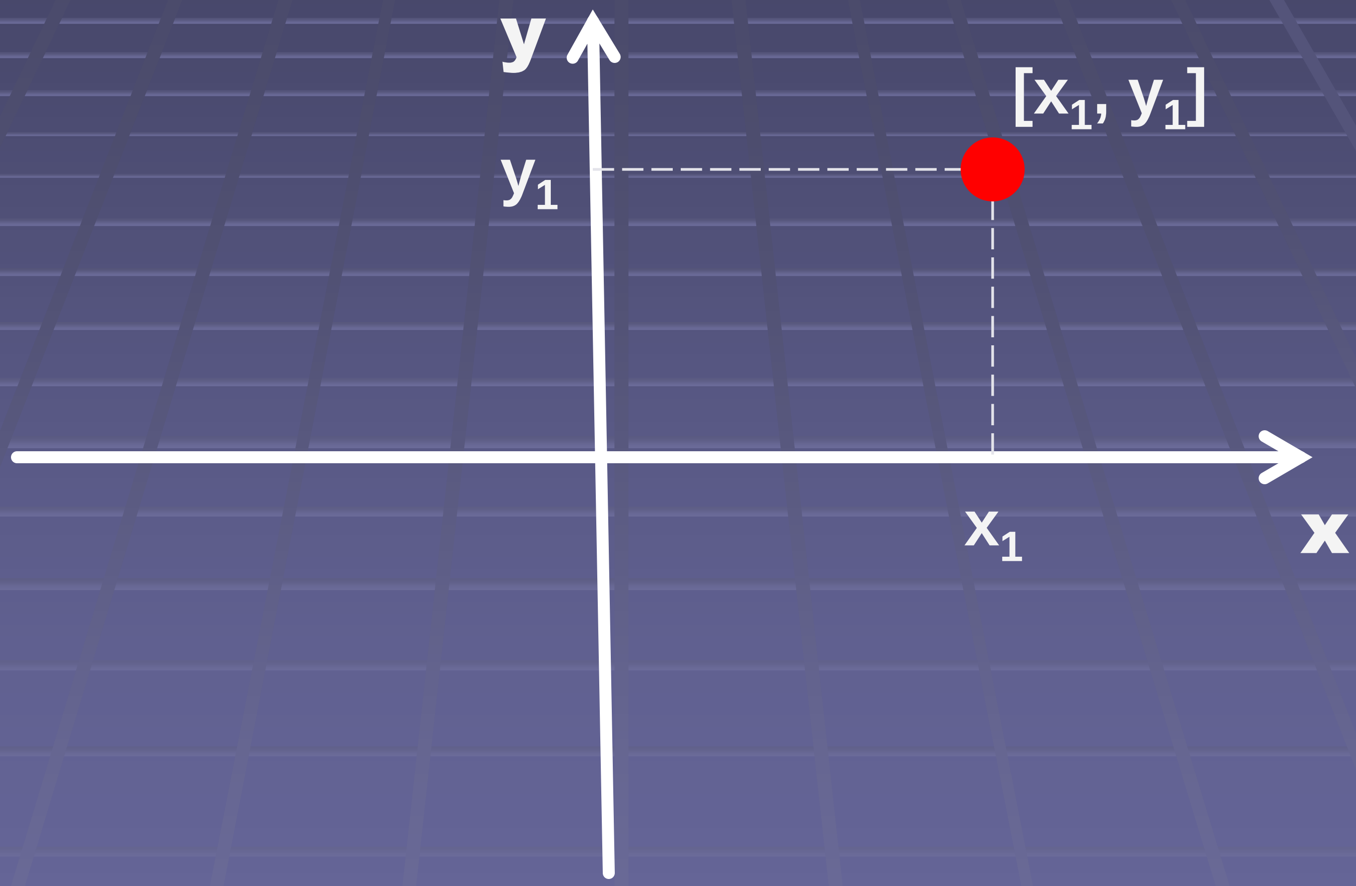
Komplikace pro algoritmy

- „Je vidět na první pohled“ nefunguje
- Reálná čísla (!)
- Speciální a degenerované případy
- Definiční obory operací
  - dělení
  - goniometrie
- Asymptotická složitost
  - i kvadratické řešení může být pomalé (!!)

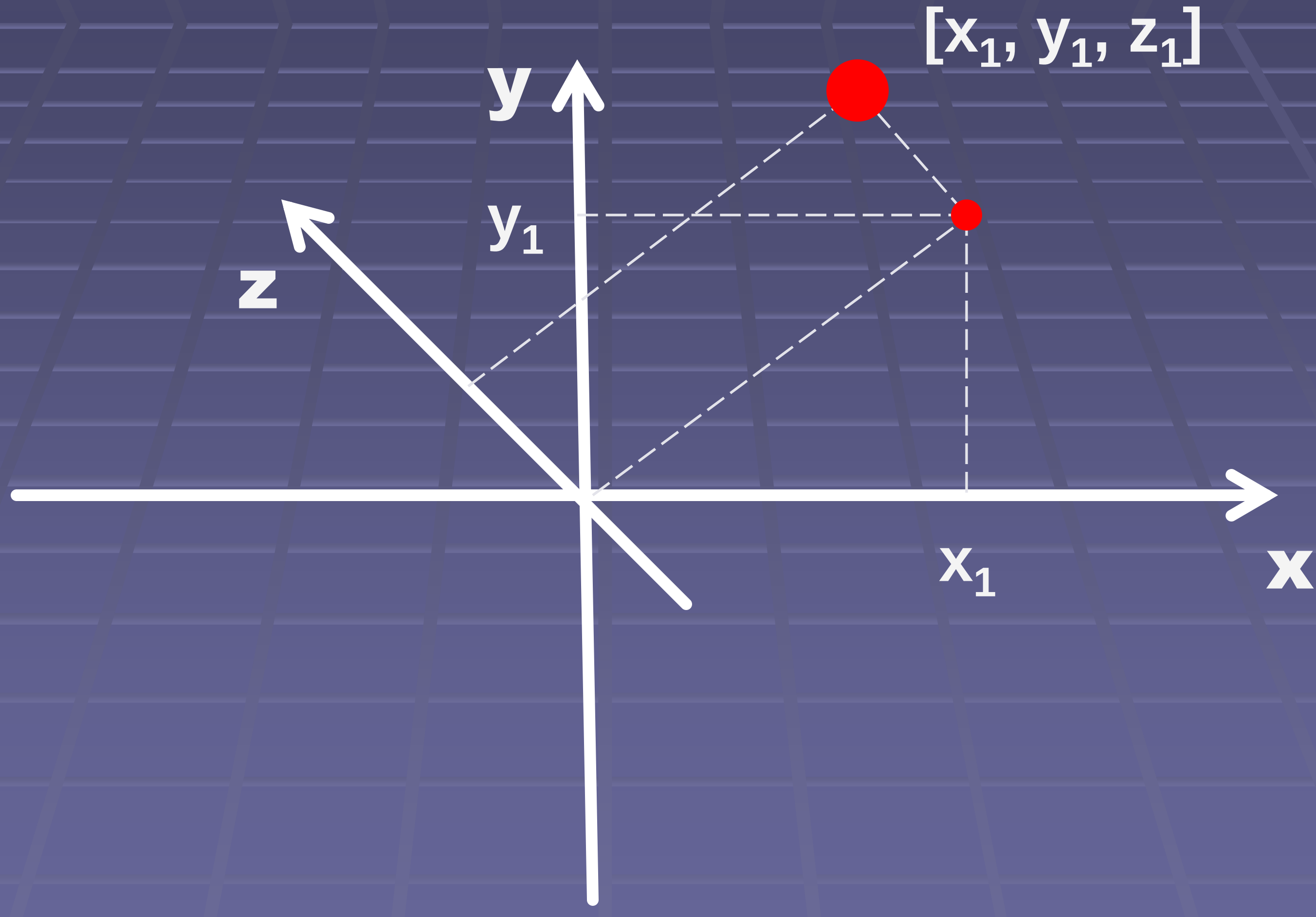
Základy výpočetní geometrie

- I jednoduché poučky nabízejí mnoho možností pro řešení problémů
  - Pythagorova věta
  - Goniometrické funkce
  - Vektorový součin
- Dvojnásob zde platí: **přemýšlet!!!**
  - Snaha najít <u>nejjednodušší</u> řešení

Souřadnicový systém

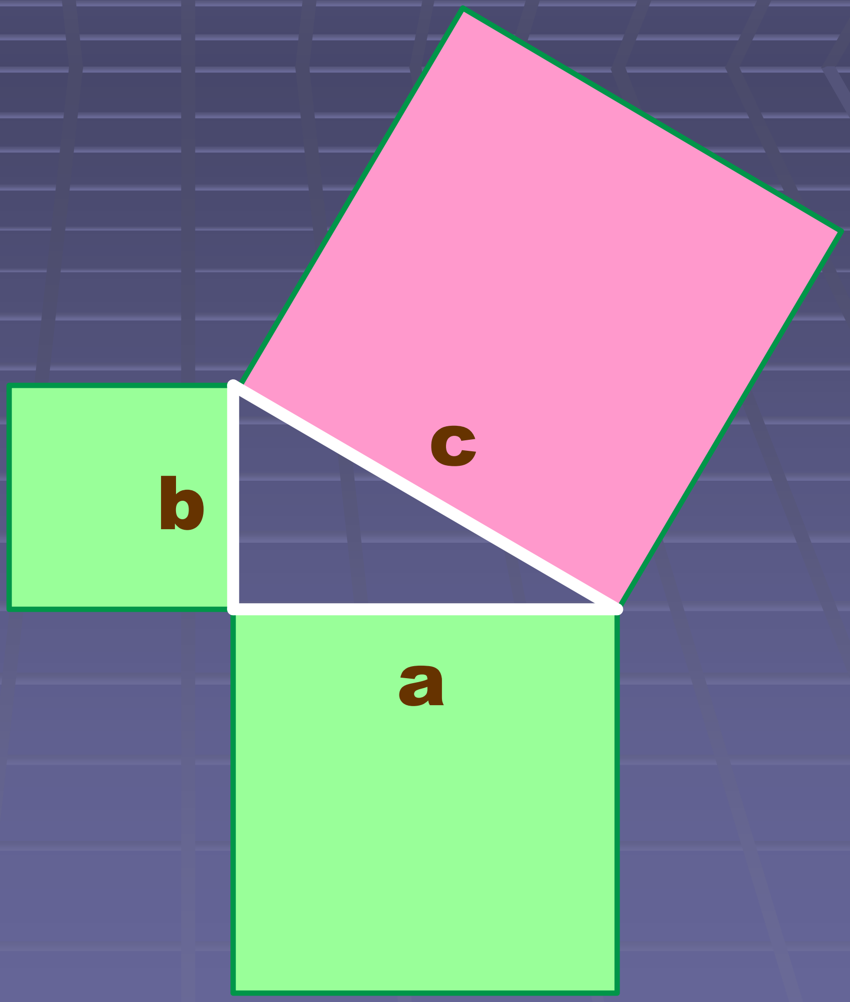


Souřadnicový systém 3D

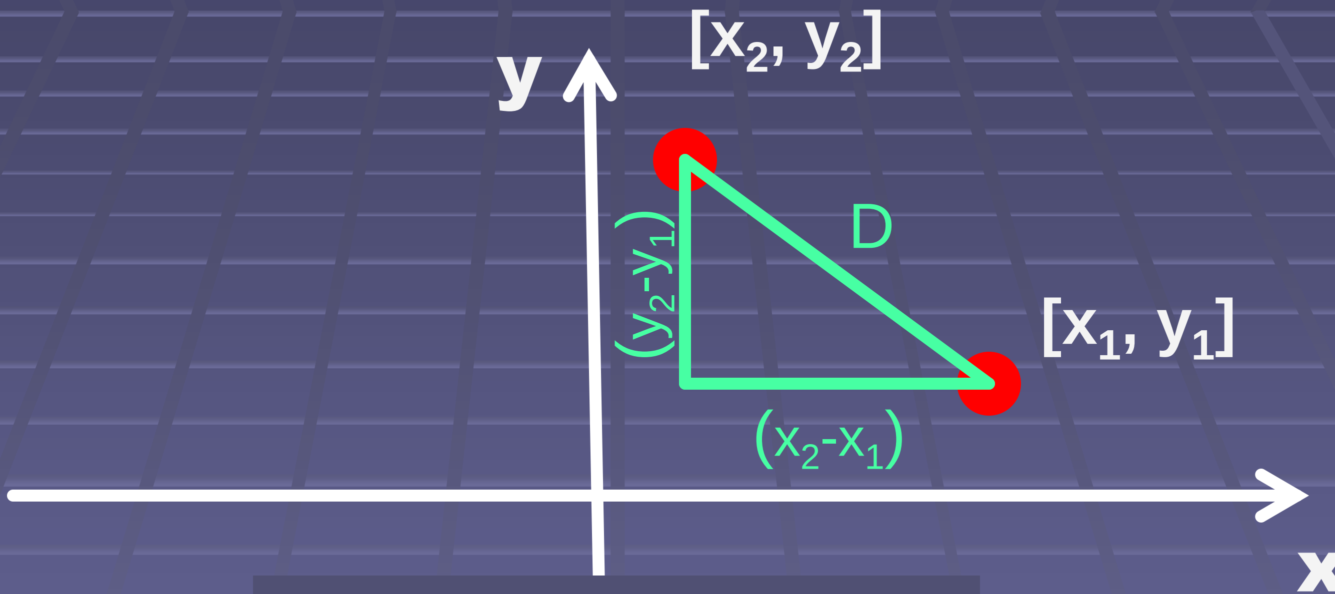


Pythagorova věta

$a^2 + b^2 = c^2$



Vzdálenost bodů



$$D^2=(x_2-x_1)^2 + (y_2-y_1)^2$$

$$D = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

Vzdálenost bodů

- $D = sqrt((x_2-x_1)^2 + (y_2-y_1)^2)$

- Co záporné souřadnice?
- Nebo opačné pořadí bodů?
- A dva stejné body?
  - Nic z toho nám nevadí
  - Vstup pro odmocninu je vždy nezáporný

Na co dávat pozor?

- U <u>každého</u> porovnávání
  - Pamatovat na nepřesnosti
  - Používat „epsilon“

```
if (a >= b) ...

if (a + EPS > b) ...
```

Na co dávat pozor?

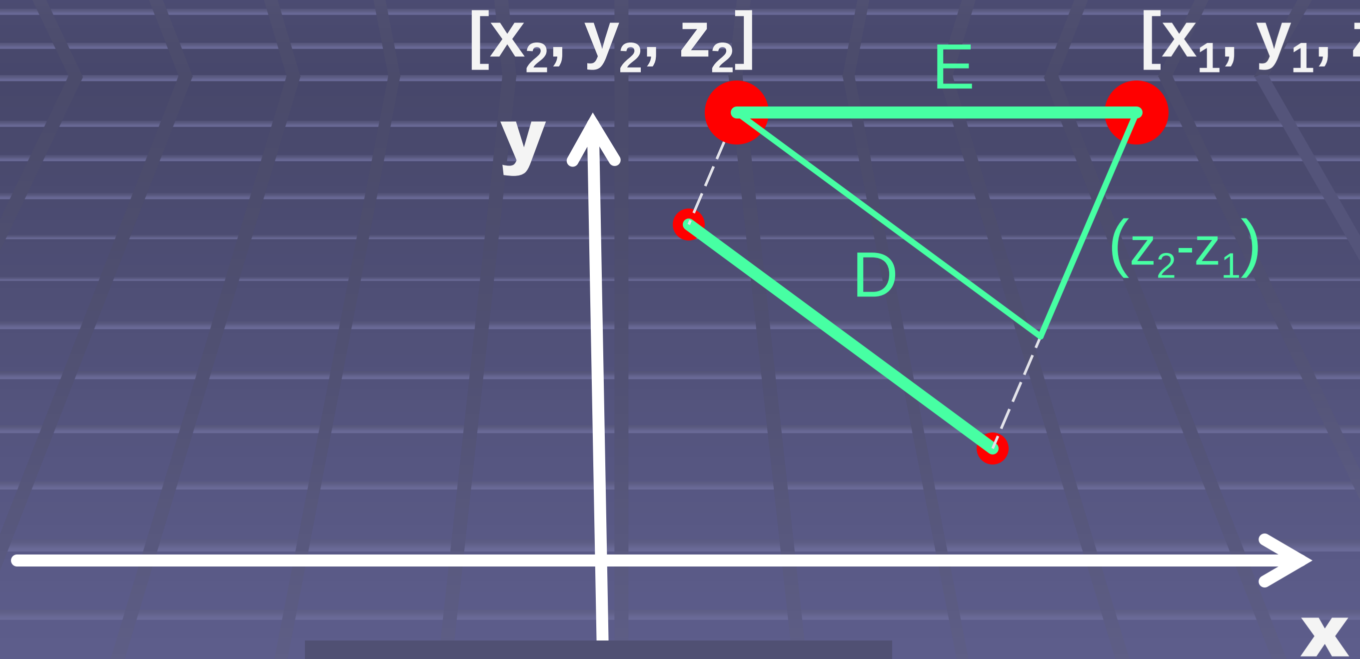
- U <u>každého</u> dělení
  - Nemůže být jmenovatel nula?
  - A co blízko nule?

- U <u>každé</u> odmocniny
  - Nemůže být vstup záporný?
  - Ani „skoro“?

Na co dávat pozor?

- Goniometrické funkce:
  - tan(x) – výsledek může být nedefinován
  - atan(x) – často je vstupem podíl
    - použijte raději atan2
- Další problémy?
  - Vždy přemýšlet „kde se to může zvrtnout“
  - x,y,z a indexy – pozor při copy&paste

Vzdálenost bodů 3D



$$E^2 = D^2 + (z_2-z_1)^2$$

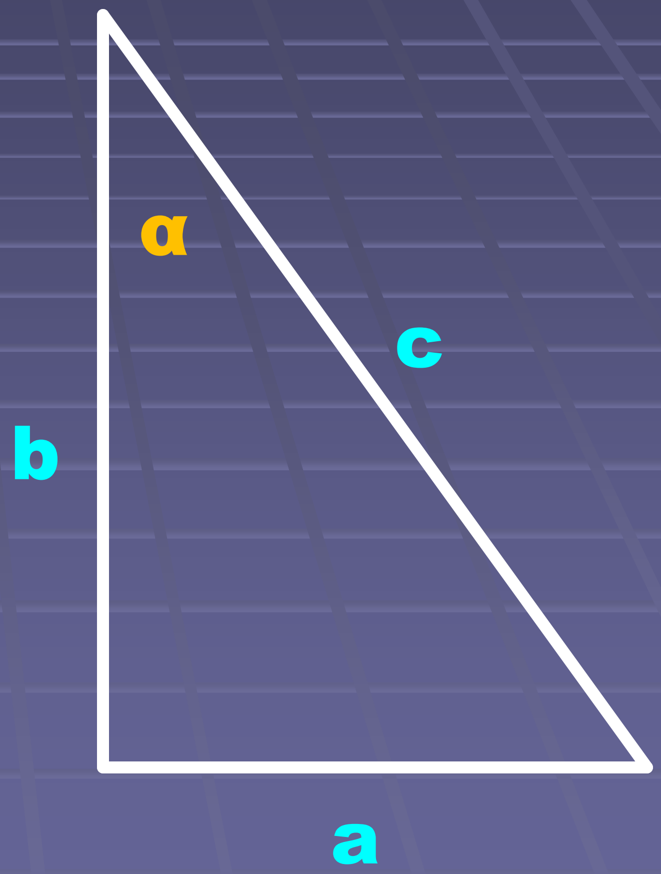
$$E^2 = (x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2$$

$$E = \mathrm{sqrt}((x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2)$$

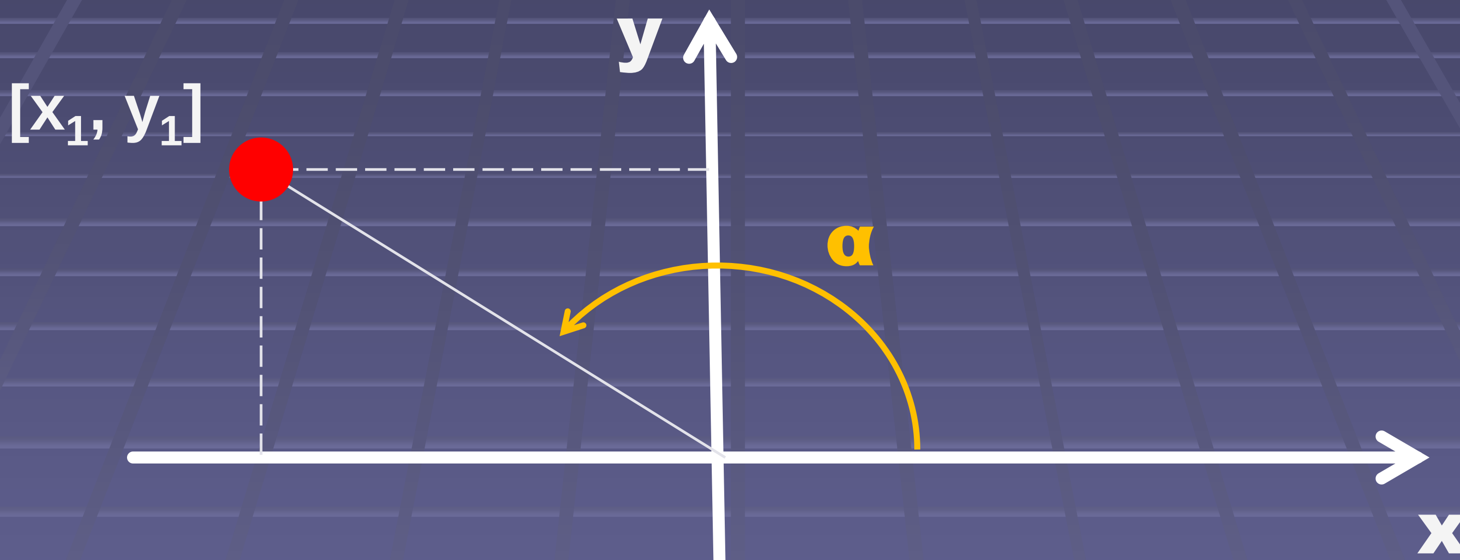
Goniometrické funkce

- sin(α) = a/c
- cos(α) = b/c
- tan(α) = a/b

- α = Math.asin(a/c)
- α = Math.atan(a/b)
  - radiány!!



Goniometrické funkce



$\tan(\alpha) = y_1 / x_1$

$\alpha = \text{Math.atan}(\ y_1 / x_1\ )$?

Goniometrická funkce atan2

- double alpha = Math.atan2(y, x)
- Ekvivalentní alpha = Math.atan(y/x)
  - Nevadí „dělení nulou“
  - Umí rozlišit kvadranty
- Převod kartézských souřadnic na polární
  - Jak převádět tam a zpět?

Polární souřadnice

```
double x, y;
x = r * cos(theta);
y = r * sin(theta);
```

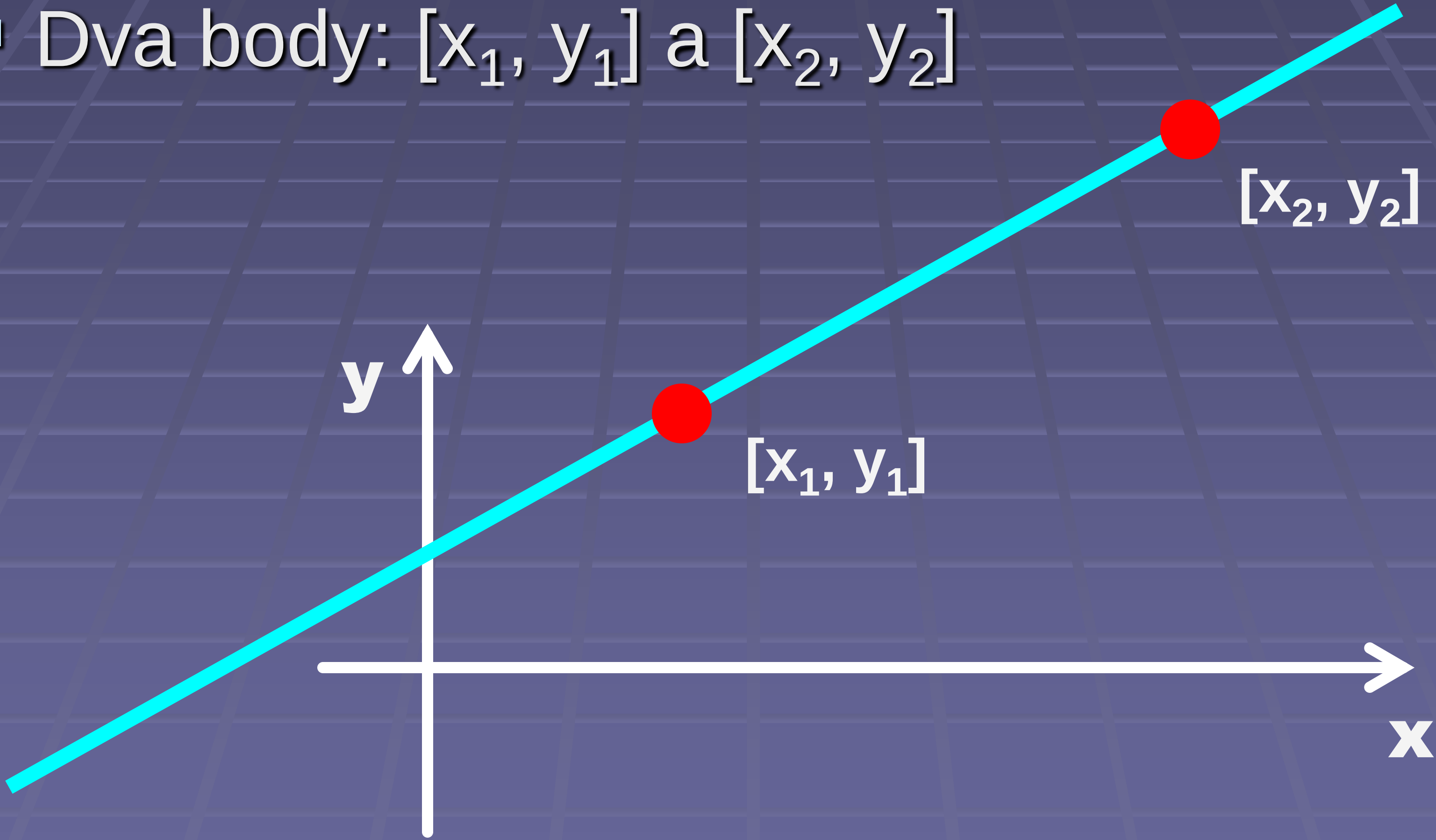
```
double r, theta;
r = Math.sqrt(x*x + y*y);
theta = Math.atan2(y, x);
```

Úsečky a přímky

- Reprezentace dvěma body
  - $[x_1, y_1]$ $[x_2, y_2]$
- Reprezentace rovnicí
  - x = x1 + t.dx , y = y1 + t.dy   (parametrická)
  - y = d.x + e   (směrnicová)
  - a.x + b.y + c = 0   (obecná)

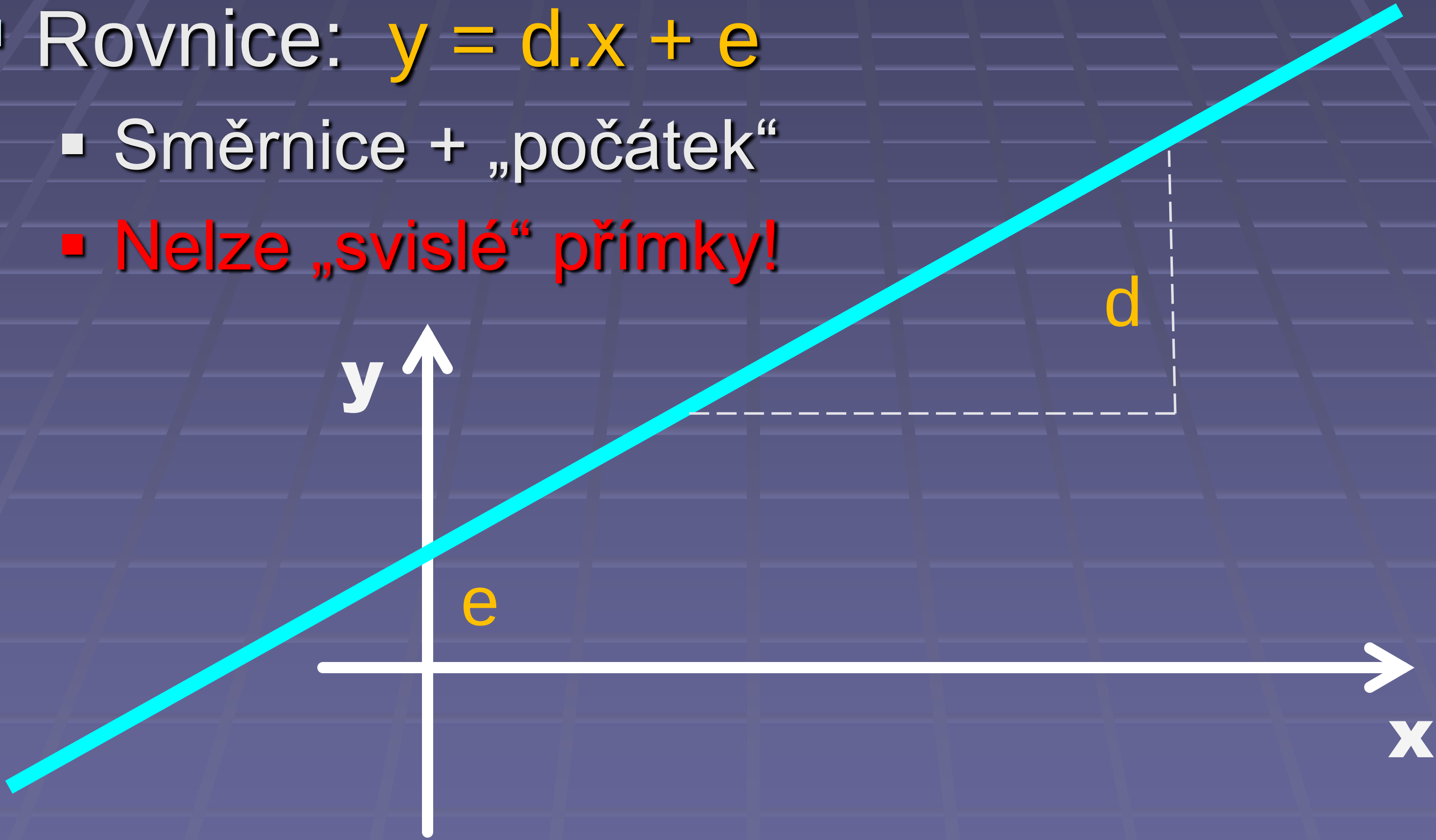
Reprezentace přímky

- Dva body: $[x_1, y_1]$ a $[x_2, y_2]$



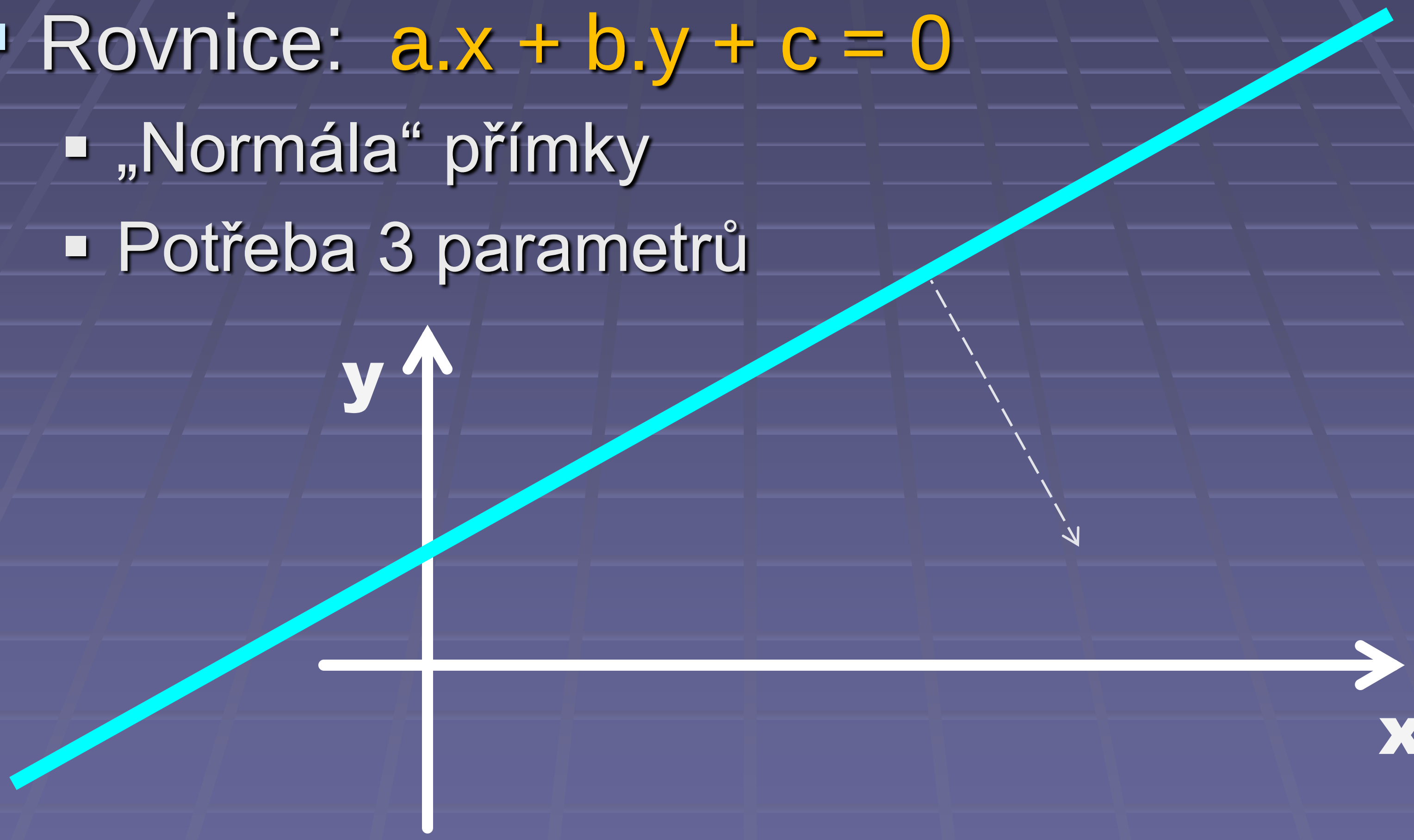
Reprezentace přímky

- Rovnice: $y = d.x + e$
  - Směrnice + „počátek“
  - Nelze „svislé“ přímky!

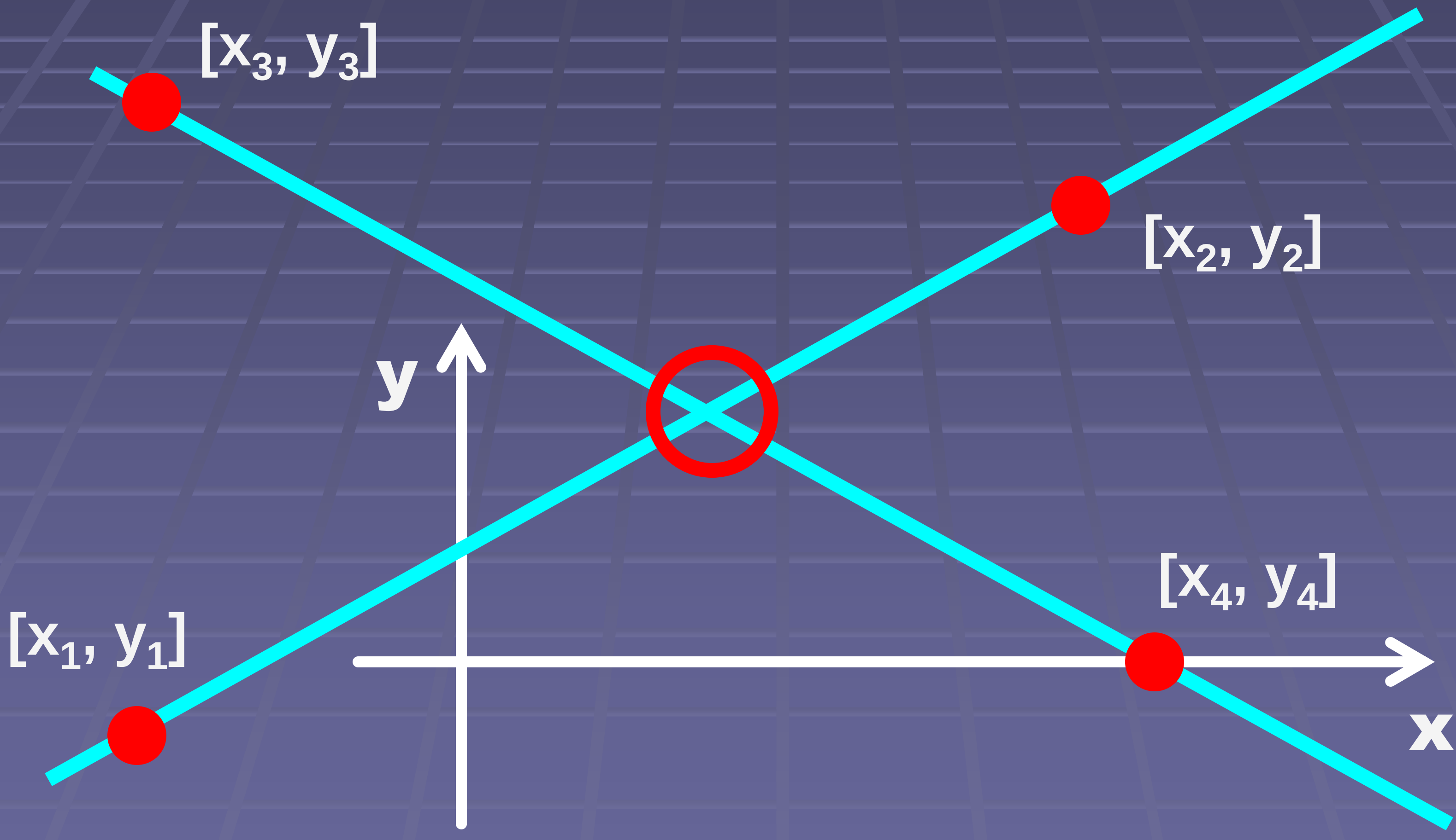


Reprezentace přímky

- Rovnice: $a.x + b.y + c = 0$
  - „Normála“ přímky
  - Potřeba 3 parametrů



Průsečík dvou přímek



Průsečík přímek – rovnice

- První přímka
  - $x = x_1 + t . (x_2 - x_1)$
  - $y = y_1 + t . (y_2 - y_1)$
- Druhá přímka
  - $x = x_3 + t . (x_4 - x_3)$
  - $y = y_3 + t . (y_4 - y_3)$
- Průsečík
  - $x_1 + t_1 . (x_2 - x_1) = x_3 + t_2 . (x_4 - x_3)$
  - $y_1 + t_1 . (y_2 - y_1) = y_3 + t_2 . (y_4 - y_3)$

Průsečík přímek – rovnice

- Průsečík
  - $x_1 + t_1 . (x_2 - x_1) = x_3 + t_2 . (x_4 - x_3)$
  - $y_1 + t_1 . (y_2 - y_1) = y_3 + t_2 . (y_4 - y_3)$
- => Soustava dvou rovnic
  - Lze řešit pomocí determinantů

Průsečík přímek – řešení

- $x_1 + t_1 . (x_2 - x_1) = x_3 + t_2 . (x_4 - x_3)$
- $y_1 + t_1 . (y_2 - y_1) = y_3 + t_2 . (y_4 - y_3)$

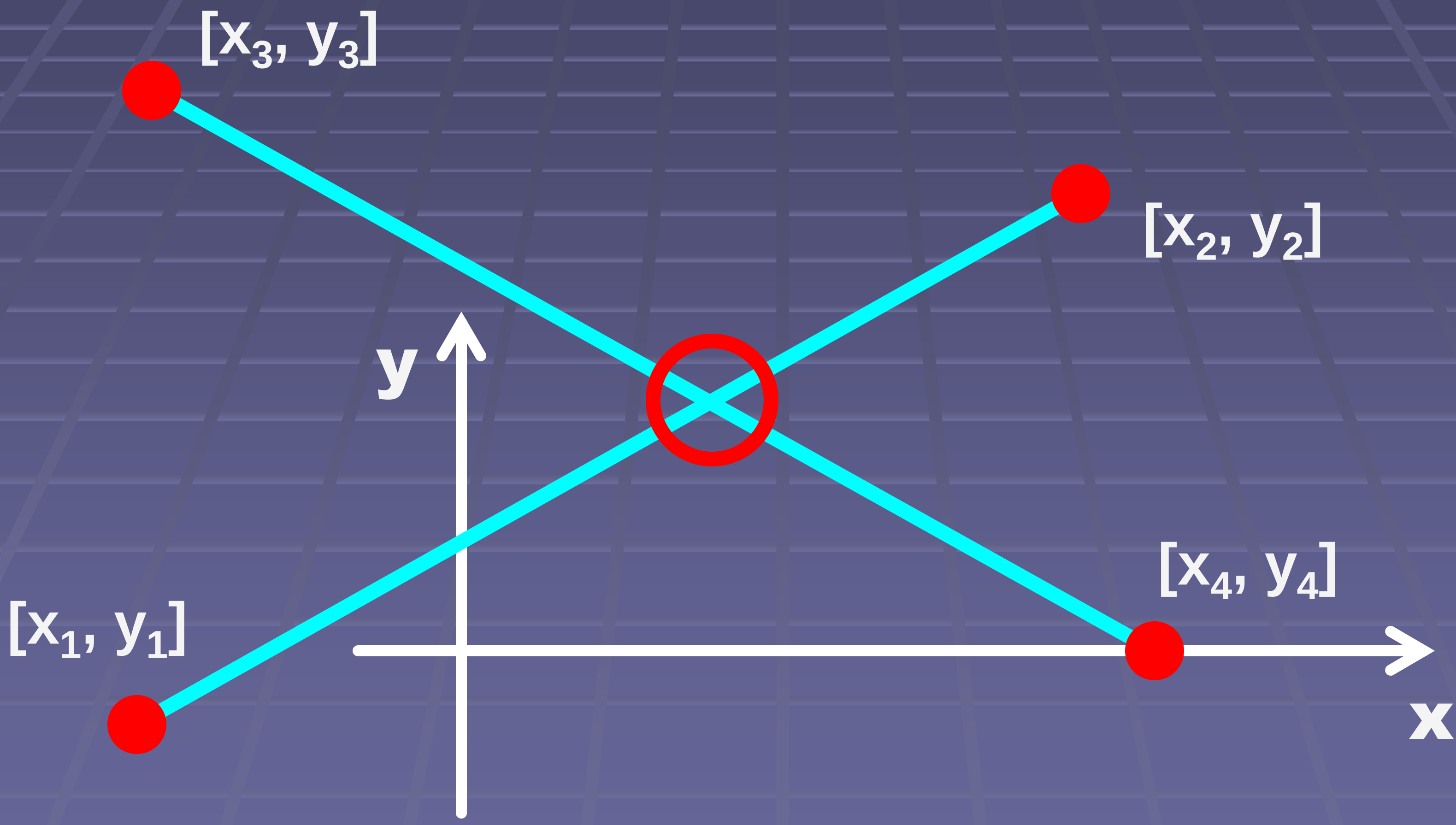
- Determinanty rovnic:
  - $D_0 = (y_4 - y_3).(x_2 - x_1) – (x_4 - x_3).(y_2 - y_1)$
  - $D_1 = (x_4 - x_3).(y_1 - y_3) – (y_4 - y_3).(x_1 - x_3)$
  - $D_2 = (x_2 - x_1).(y_1 - y_3) – (y_2 - y_1).(x_1 - x_3)$

  - $t_1 = D_1 / D_0$
  - $t_2 = D_2 / D_0$

Průsečík – nulový jmenovatel

- Determinant $D_0 = 0$
  - Přímky mají stejnou směrnici (rovnoběžné)
- $D_1 \neq 0 \neq D_2$
  - Rovnoběžky
- $D_1 = D_2 = 0$
  - Stejná přímka

Průsečík dvou úseček



Průsečík úseček

- Jako u přímek
- Navíc test, zda je bod uvnitř úseček
  - $0 \leq t_1 \leq 1$
  - $0 \leq t_2 \leq 1$
- Ale co když jsou rovnoběžné?
  - $D_0 = 0$

Průsečík úseček – degradace

- $D_0 = 0$
  - Úsečky jsou rovnoběžné (či kolineární)
- $D_1 \neq 0 \neq D_2$
  - Různé rovnoběžky => bez průsečíku
- $D_1 = D_2 = 0$
  - Kolineární. Co s tím?

Průsečík úseček – kolineární

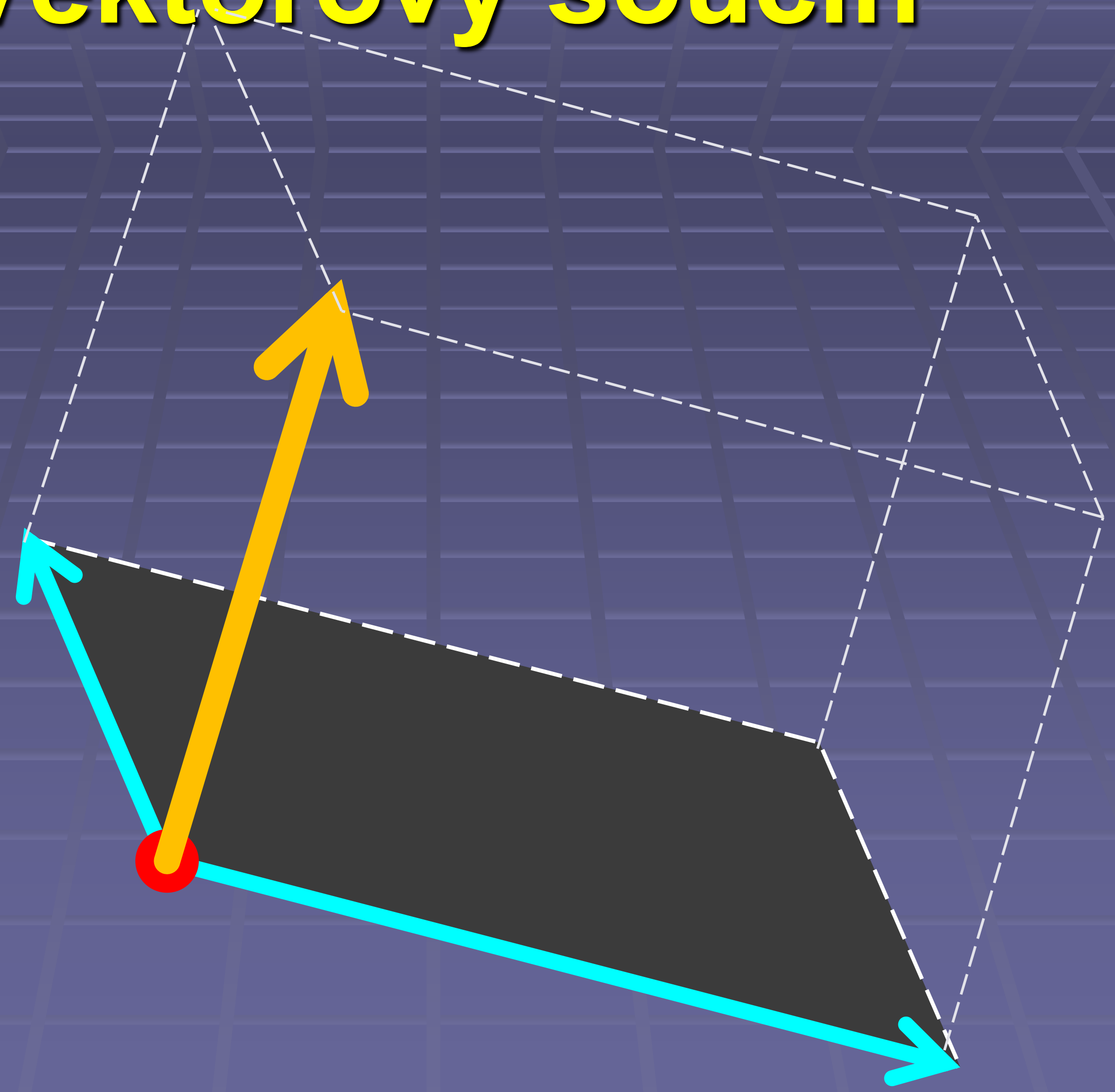
- Porovnáme, zda se překrývají
- Test intervalů
- Stačí vzít jednu souřadnici (např. x)

- Ale co když $x_1 = x_2 = x_3 = x_4$?
  - Další degradace
  - Použijeme y

Průsečík úseček – závěr

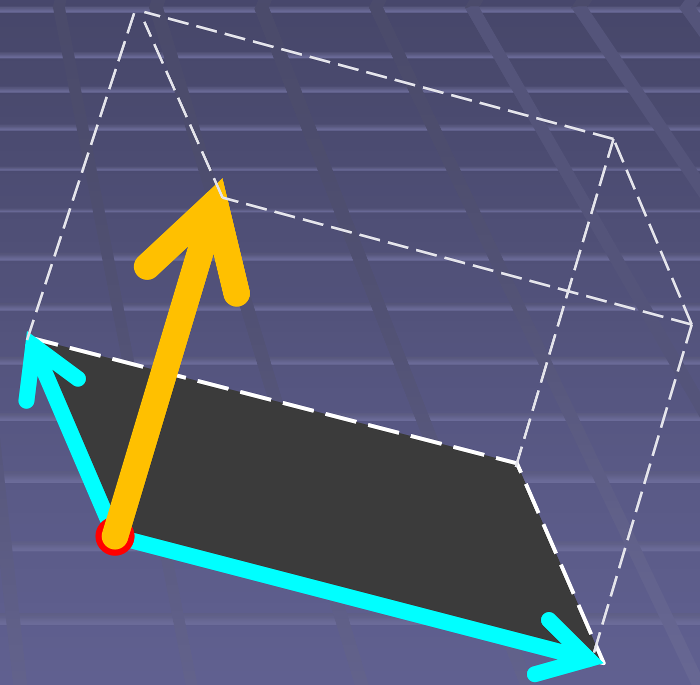
- Ošetřování speciálních případů
- Na co dalšího bychom měli dát pozor?
  - Úsečky nulové délky

Vektorový součin



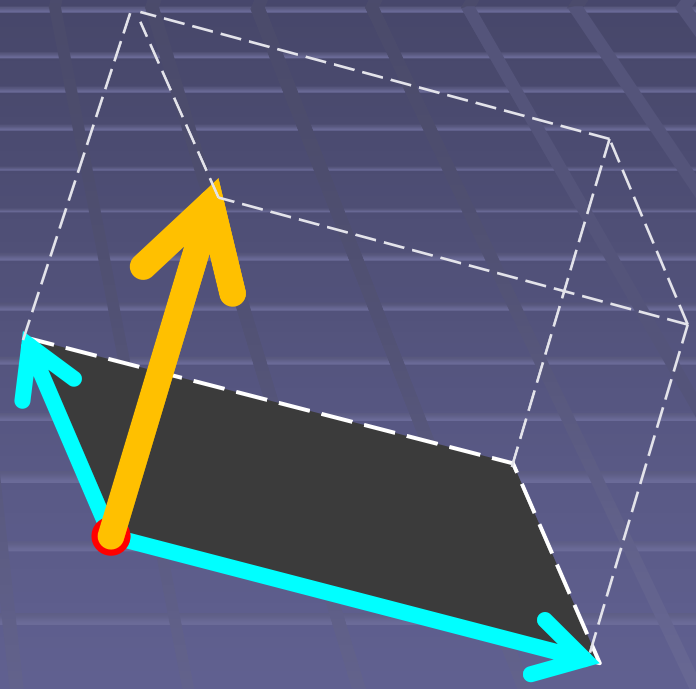
Vektorový součin

- Pouze ve 3D
- Velikost
  - Plocha rovnoběžníka
- Směr
  - Kolmý na oba původní
  - „Znaménko“ dle pořadí



Vektorový součin

- $[x_1, y_1, z_1] \times [x_2, y_2, z_2] = [x_S, y_S, z_S]$



- $x_S = y_1.z_2 - y_2.z_1$
- $y_S = x_2.z_1 - x_1.z_2$
- $z_S = x_1.y_2 - x_2.y_1$

Vektorový součin ve 2D

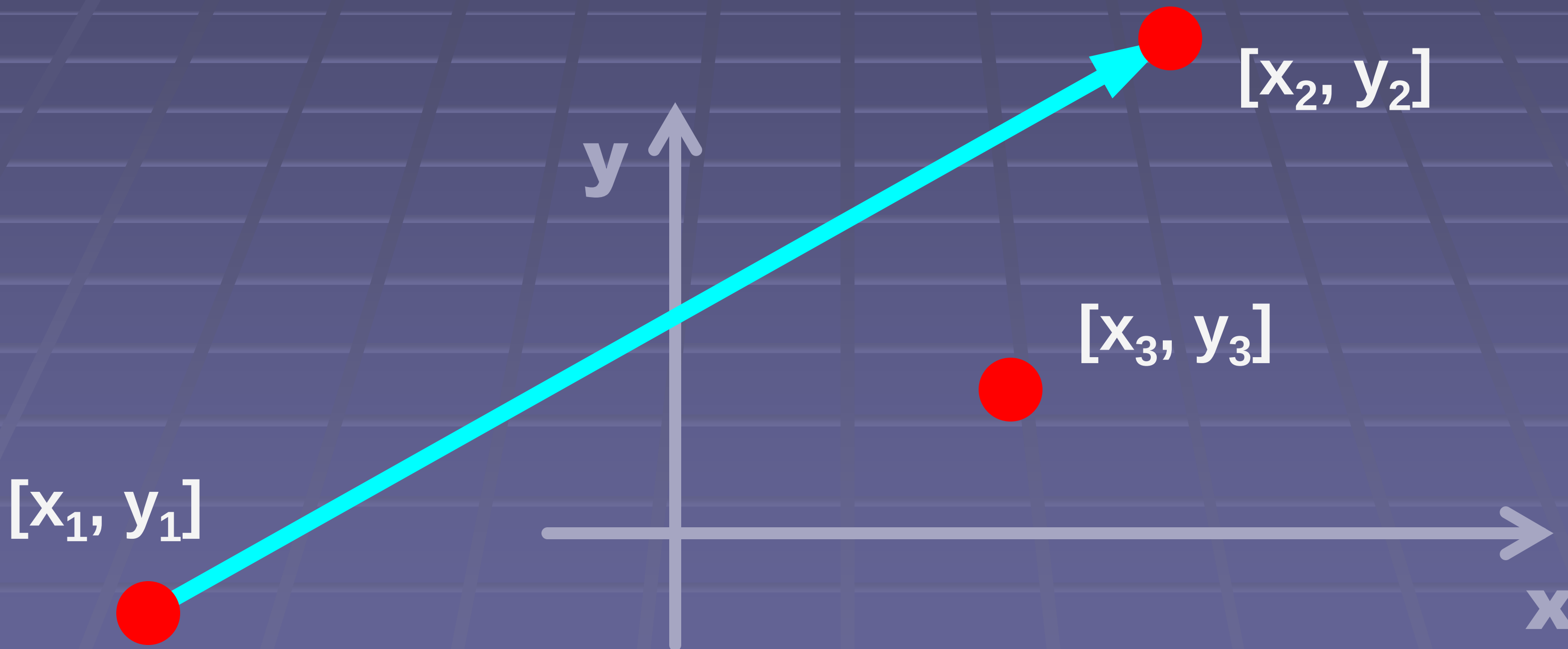
- $z_1 = z_2 = 0$

- $x_S = y_1.z_2 – y_2.z_1 = 0$
- $y_S = x_2.z_1 – x_1.z_2 = 0$
- $z_S = x_1.y_2 – x_2.y_1 = z$

- => ve 2D vyjde jen jedna souřadnice!
  - A občas vůbec nepotřebujeme reálná čísla! ☺

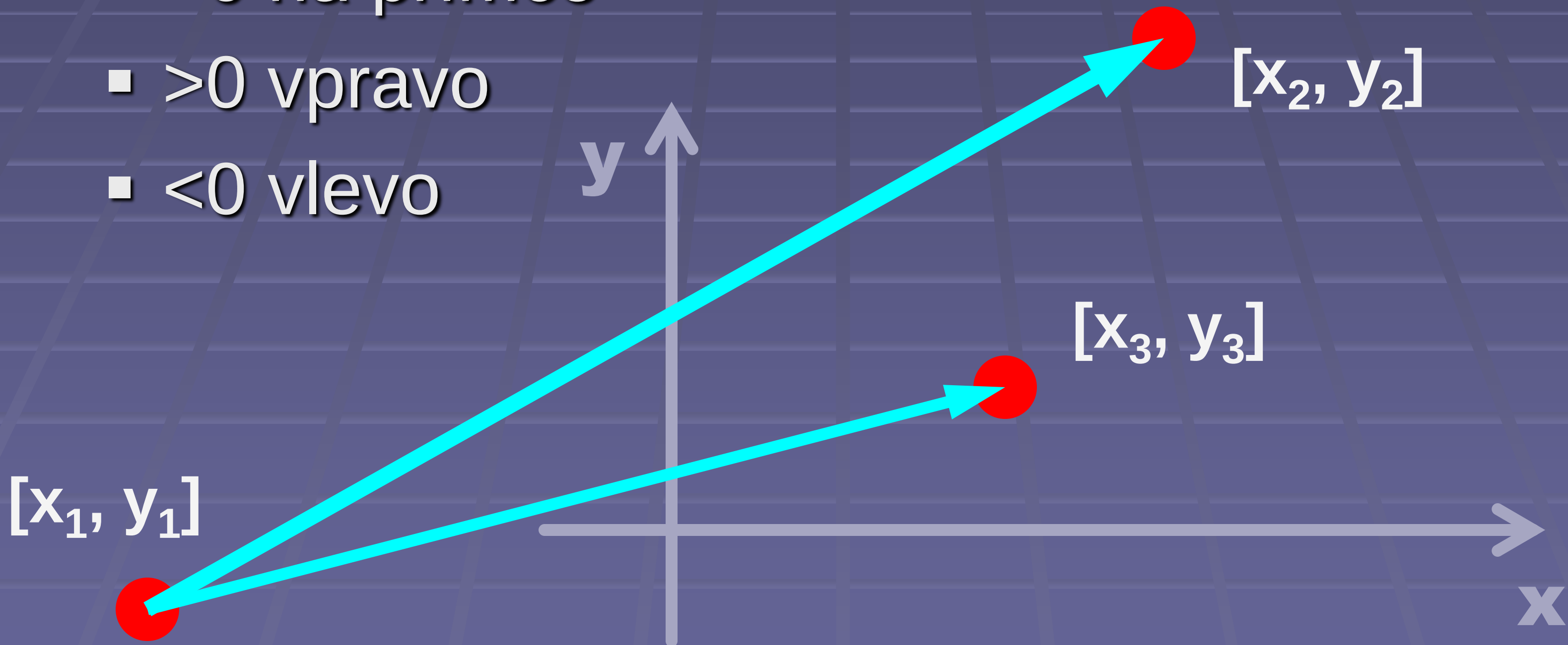
Zjištění poloroviny

- Na které straně je bod od přímky?



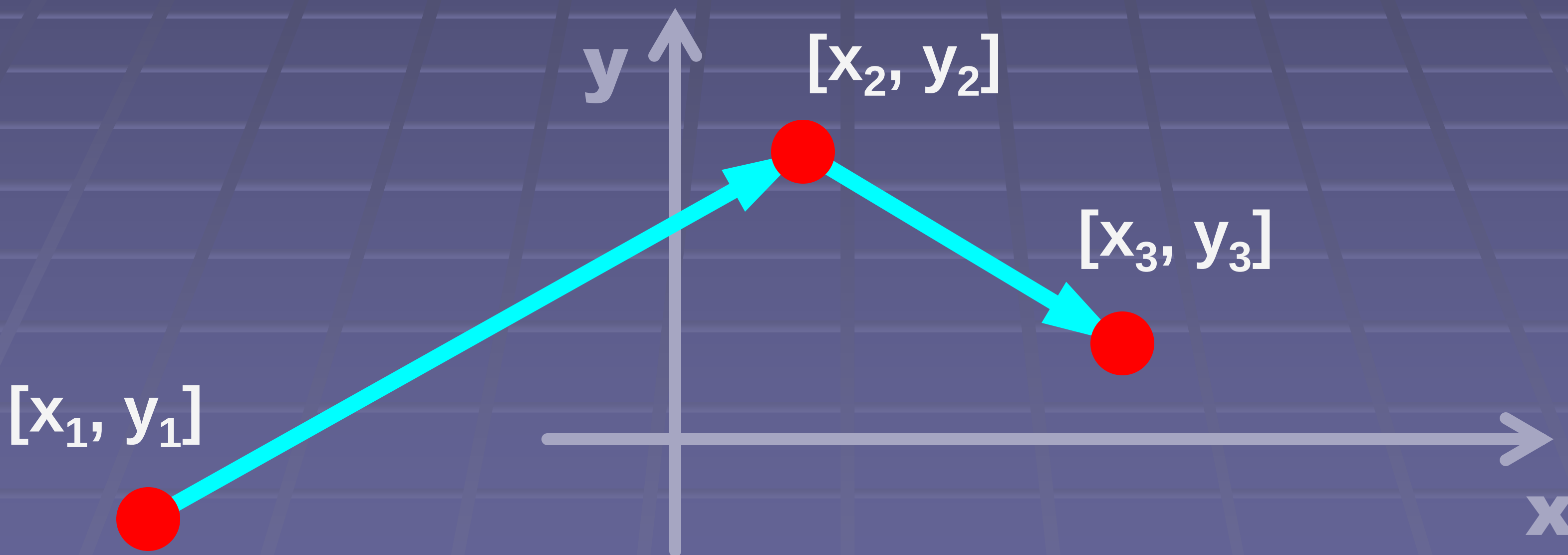
Zjištění poloroviny

- Součin $[x_3-x_1, y_3-y_1] \times [x_2-x_1, y_2-y_1]$
  - =0 na přímce
  - >0 vpravo
  - <0 vlevo



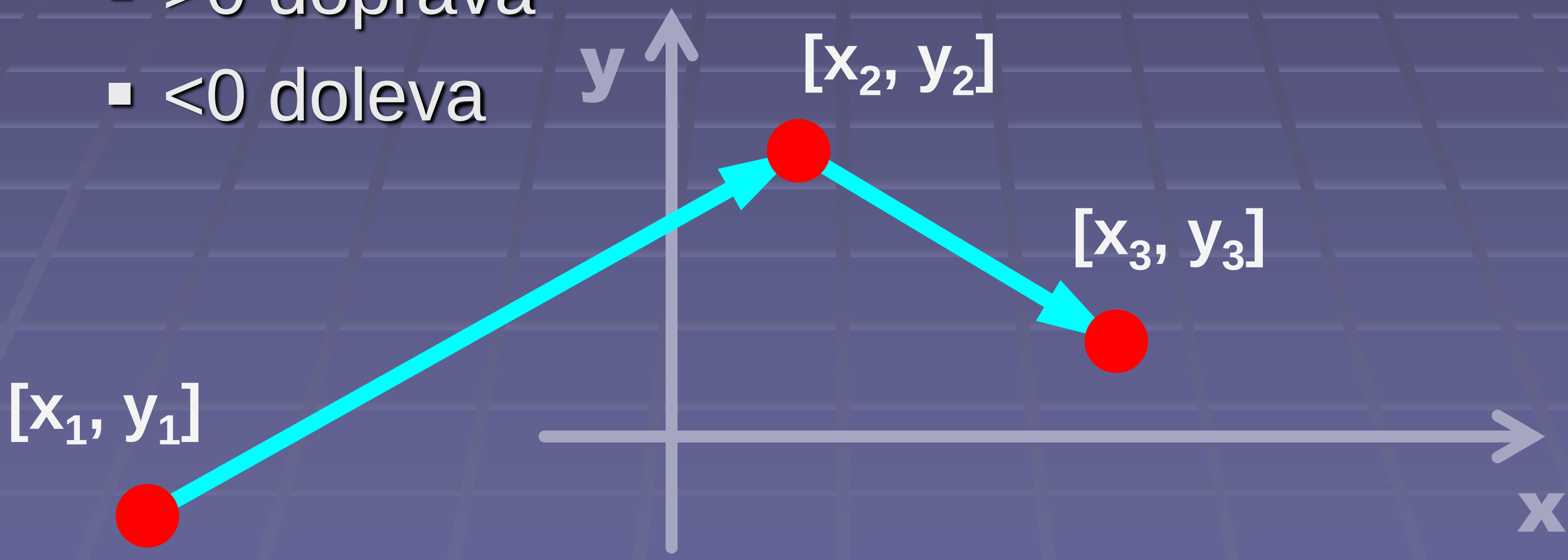
Zjištění směru

- Na kterou stranu „zatočila“ lomená čára?



Zjištění směru

- Součin $[x_3-x_2, y_3-y_2] \times [x_2-x_1, y_2-y_1]$
  - =0 rovně (či zpět)
  - >0 doprava
  - <0 doleva

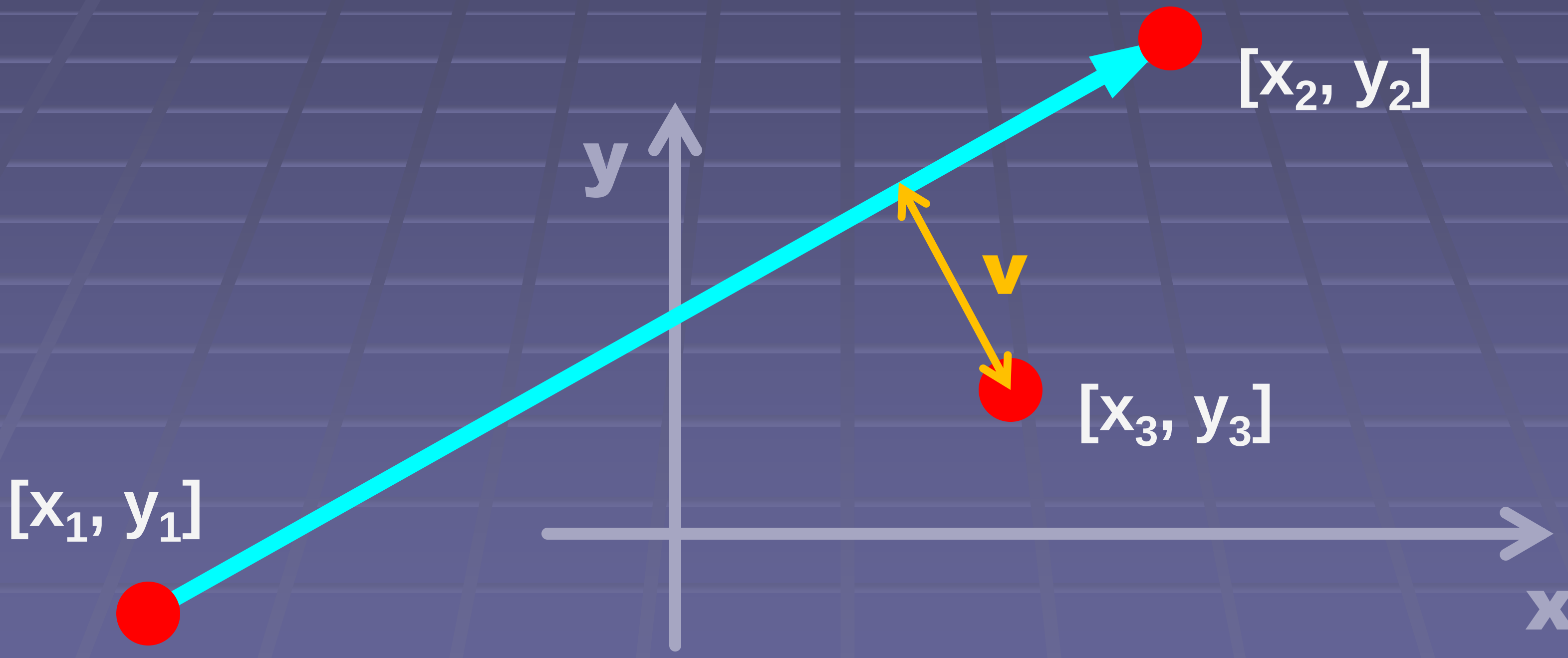


Zjištění směru

- Příklady použití
  - Ověření konvexnosti
  - Zjištění směru („po“ x „proti“ směru ručiček)

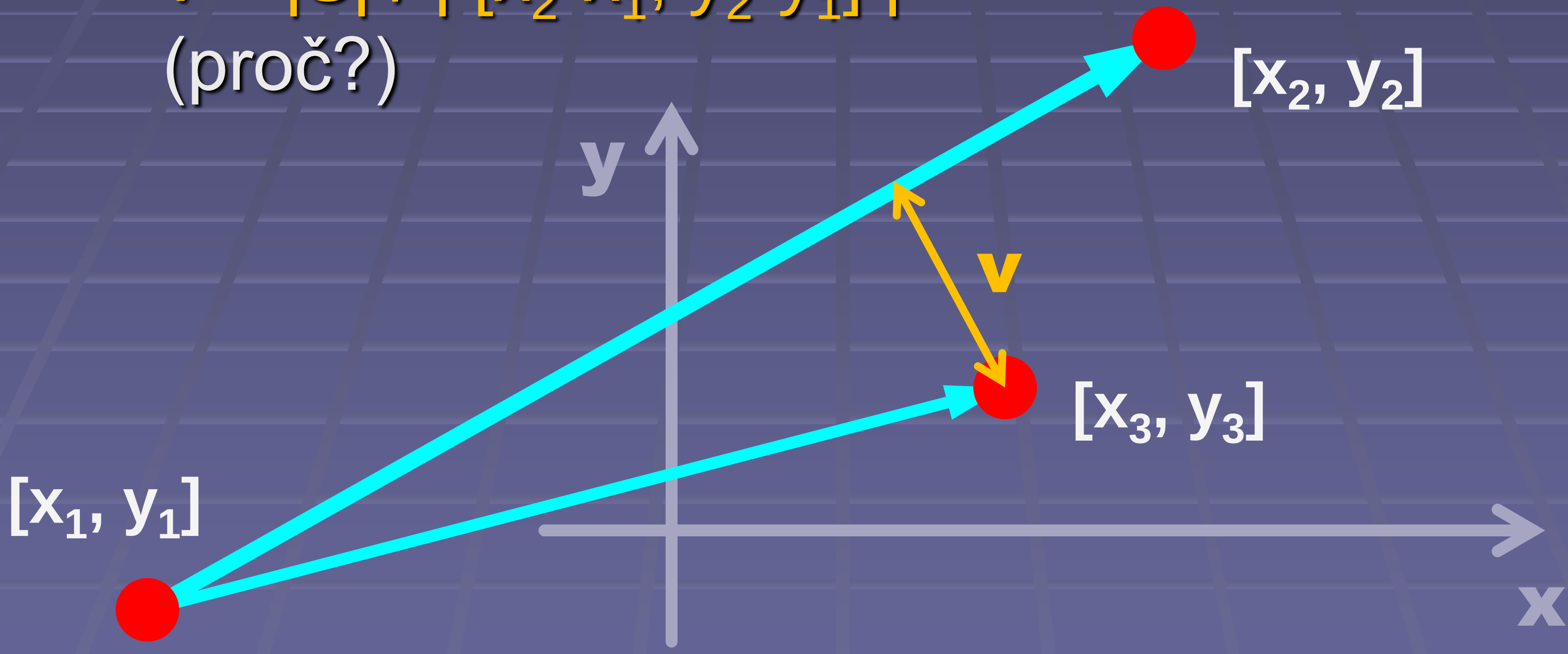
Zjištění vzdálenosti

- Jak daleko je bod od přímky?



Zjištění vzdálenosti

- Součin $S = [x_3-x_1, y_3-y_1] \times [x_2-x_1, y_2-y_1]$
  - $v = |S| / |\,[x_2-x_1, y_2-y_1]\,|$ (proč?)

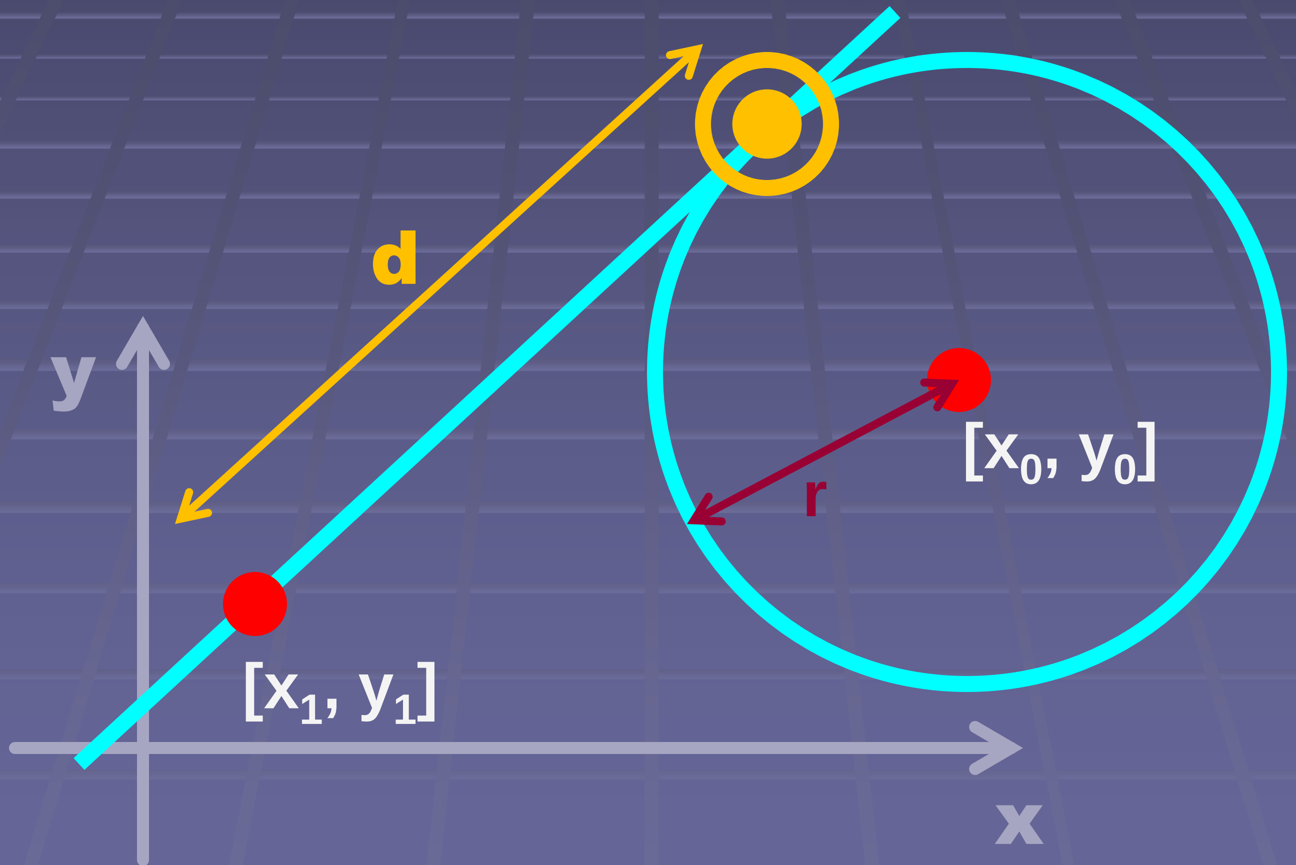


Další příklady

- Přímka vs. kružnice:
  - Délka tětivy
  - „Délka“ tečny

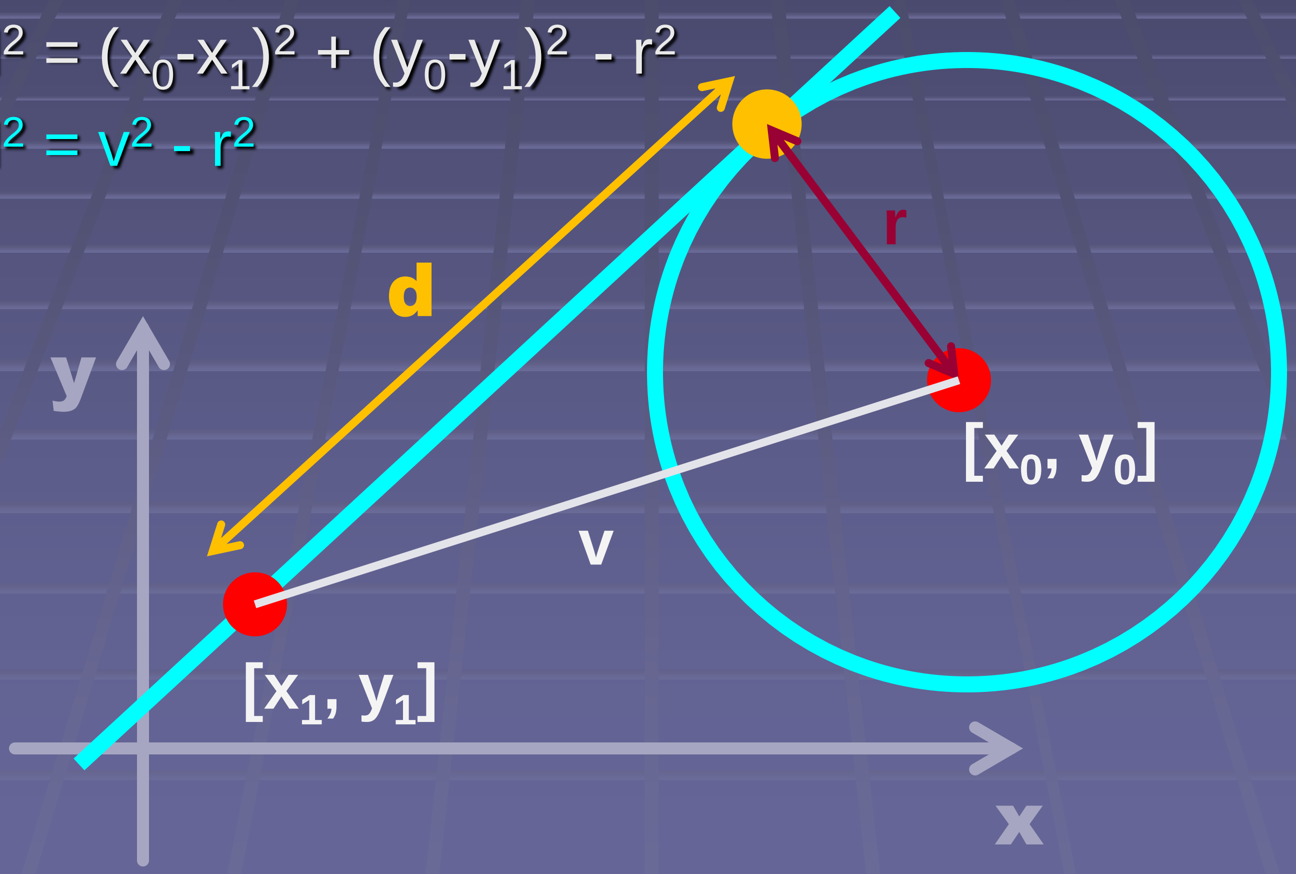
Vzdálenost na tečně

- Vzdálenost od bodu k dotyku s kružnicí



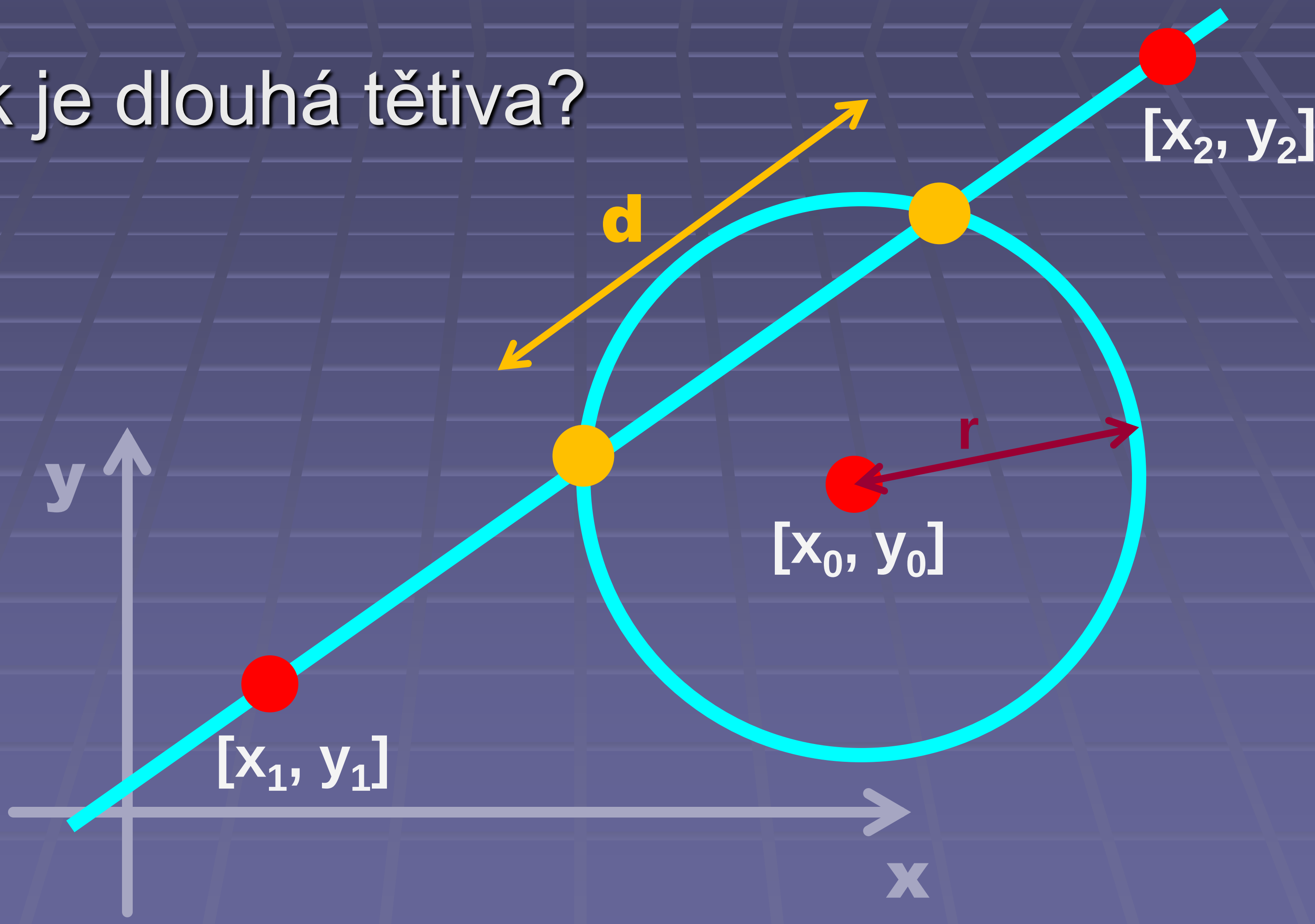
Vzdálenost na tečně

- Pravoúhlý trojúhelník => Pythagorova věta
  - $d^2 = (x_0-x_1)^2 + (y_0-y_1)^2 - r^2$
  - $d^2 = v^2 - r^2$



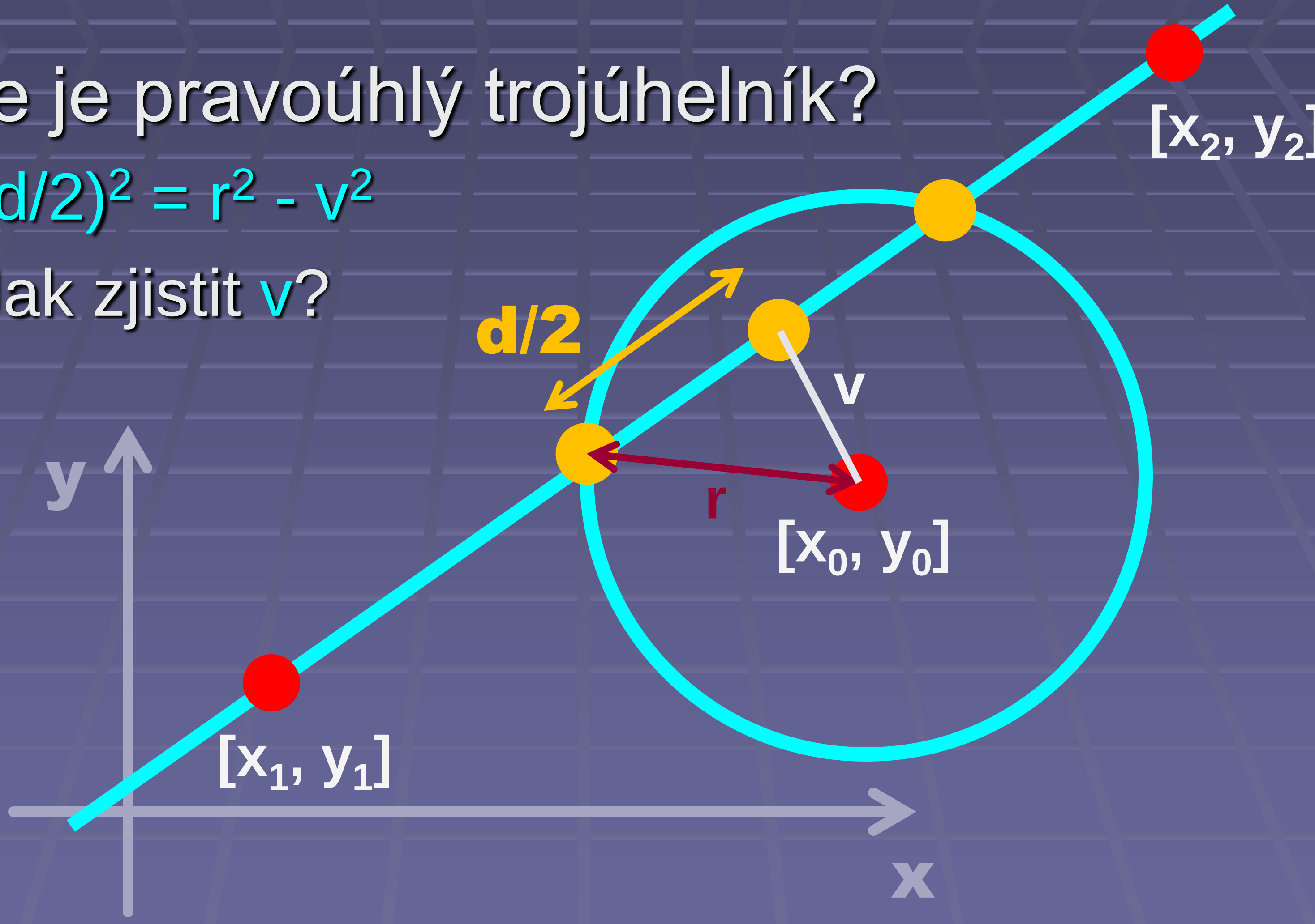
Délka tětivy

- Jak je dlouhá tětiva?



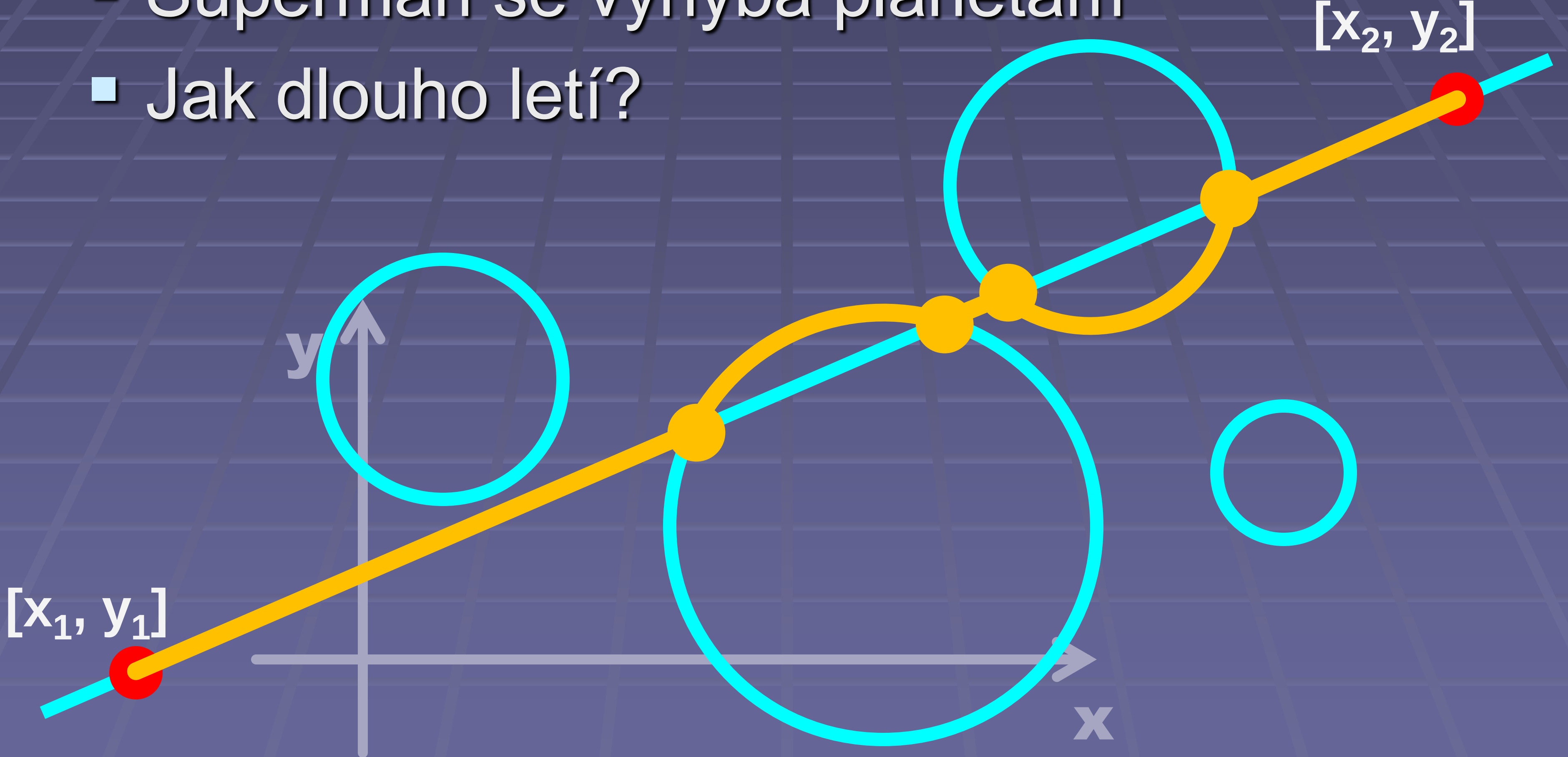
Délka tětivy

- Kde je pravoúhlý trojúhelník?
  - $(d/2)^2 = r^2 - v^2$
  - Jak zjistit v?



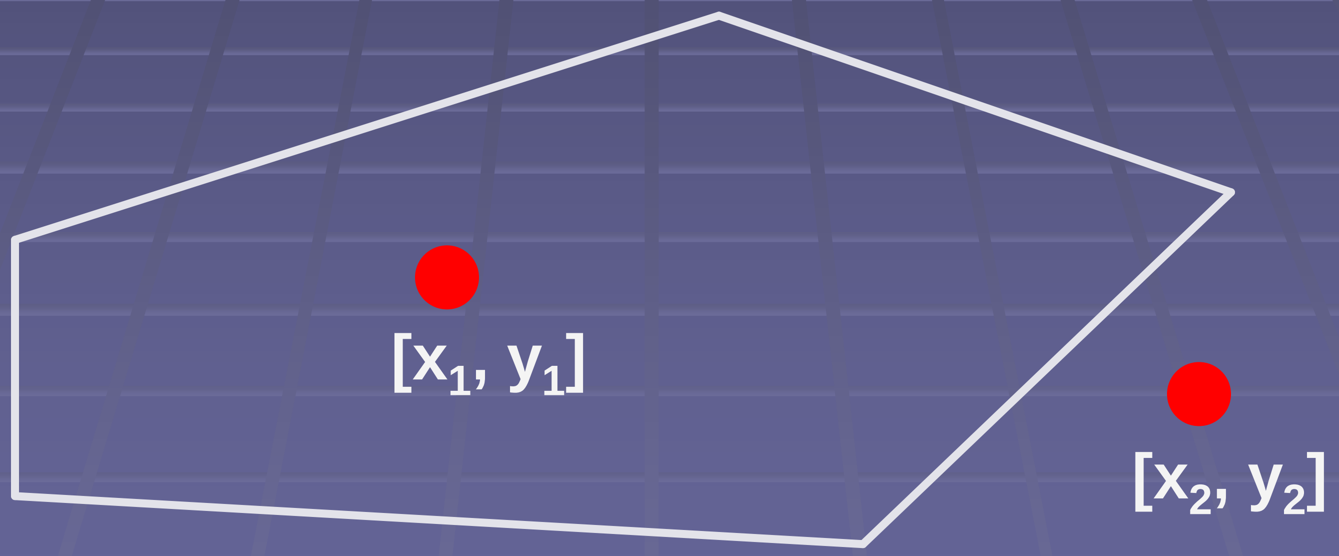
Úloha: Superman

- Superman se vyhýbá planetám
- Jak dlouho letí?



Úloha: Vnitřek obrazce

- Konvexní mnohoúhelník
- Poznat, zda bod je uvnitř nebo vně

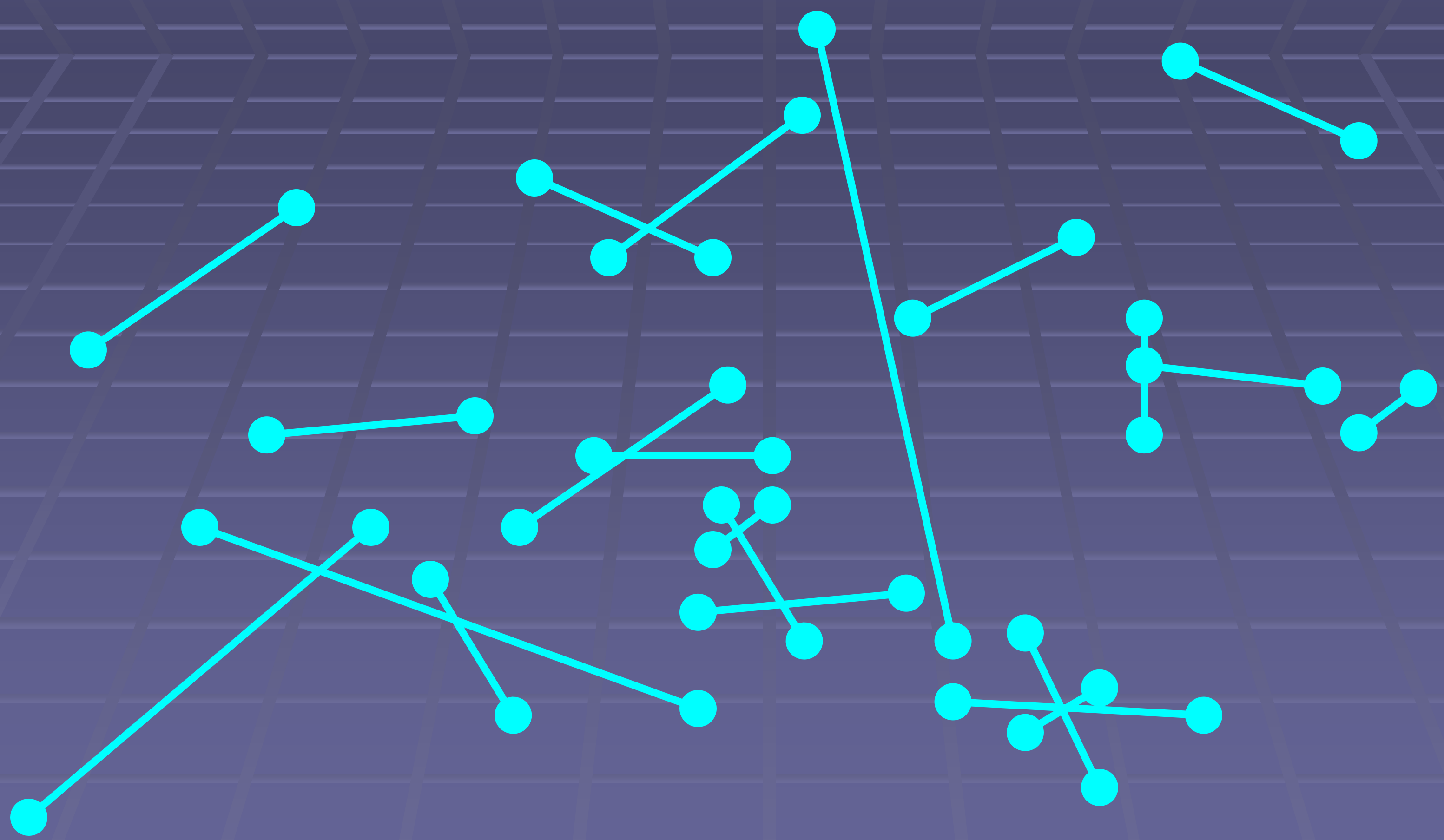


Obecné techniky V.G.

- Pro složitější příklady
  (nebude potřeba u dnešních úloh)

- „Rozděl a panuj“
- „Zametací přímka„

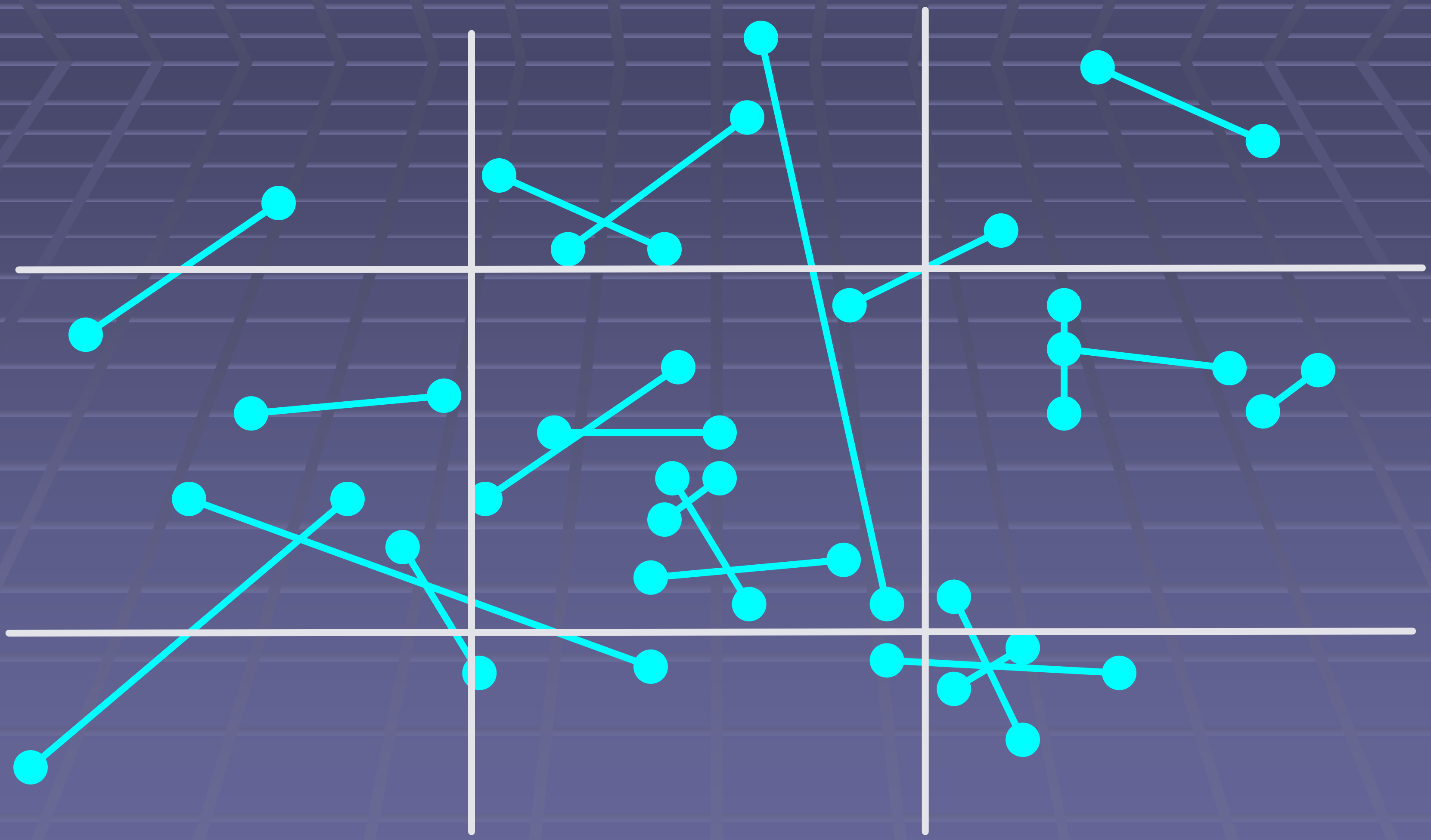
Příklad – průsečíky N úseček



Příklad – průsečíky N úseček

- Přímočaré řešení
- Porovnat každou úsečku s každou další
  - Složitost?
  - $O(n^2)$
- Pro velké počty chceme zrychlit
  - Rozdělení na oblasti
  - Řešení každé zvlášť

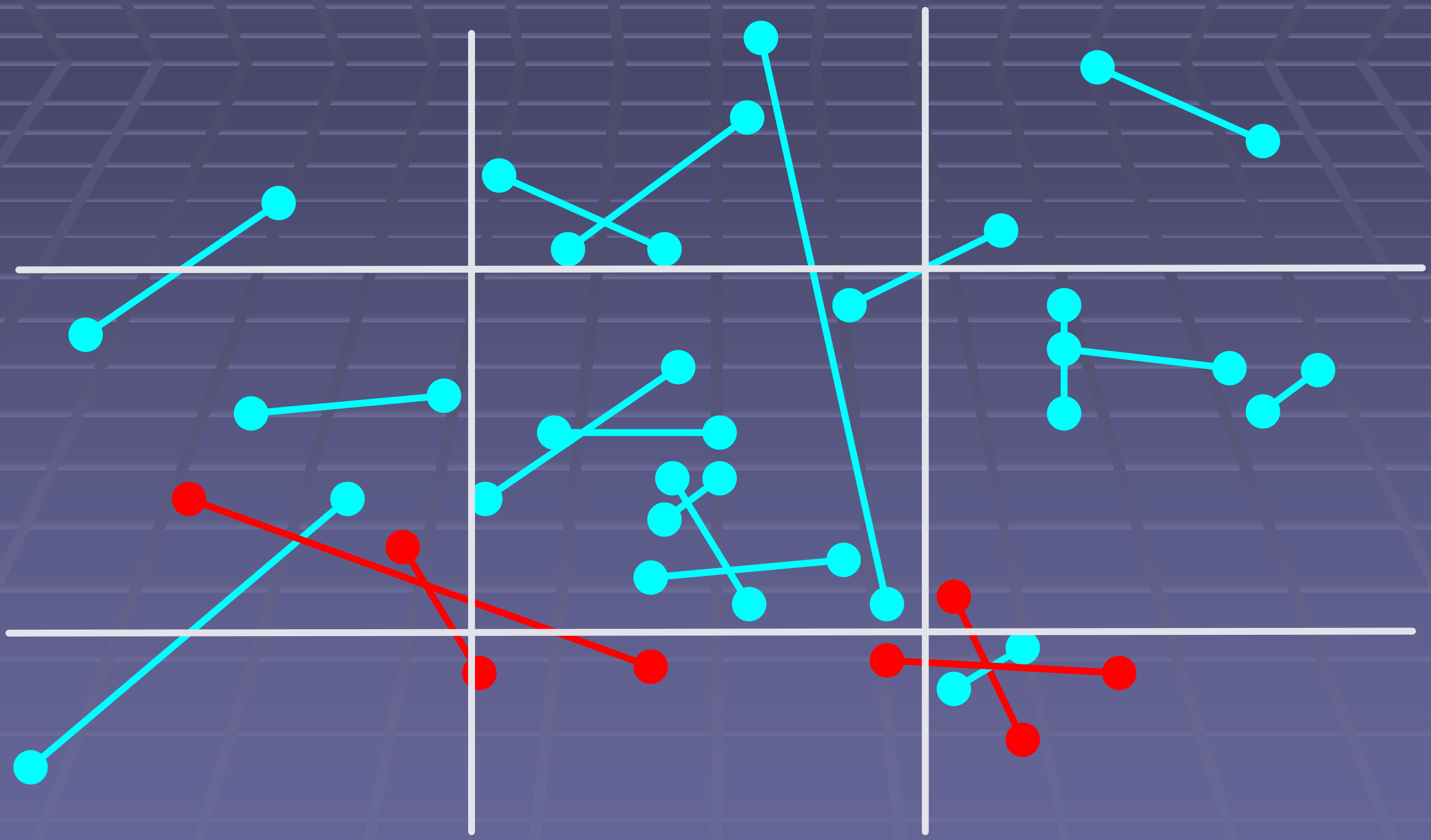
Rozděl a panuj – úsečky



Rozděl a panuj

- Asymptotická složitost
  - Původní počet kroků: n^2
  - Nový počet kroků: $k.(n/k)^2 = n^2 / k$
- Úsečky na hranici oblastí musí být v obou
  - V nejhorším případě nepomáhá
  - Mohou vznikat duplicity => odstranit

Rozděl a panuj – hranice



Zametací přímka

- Seřadíme podle souřadnice X
- Postupujeme „zleva doprava“
- Upravujeme stav
  - V případě úseček množinu „aktivních“

- Nutnost efektivních datových struktur!
  - => Může být o dost náročnější…
  - … ale typicky **O (n . log n)**

Organizační věci

Zápočty

- Limit pro zápočet: **85 bodů**
- Termín udělení: **4.1.2019** odpoledne
  - Podmínka = dostatek bodů
- **Chybí-li body, ozvěte se VČAS!!!**
  - Tj. ihned (!)

Efektivní programování II

- Podobný formát (2+2, kredity 4)
- Menší kapacita (poloviční)
- Témata blízká **BI-AG1** a **BI-AG2**:
  - Grafové algoritmy
  - A další: Řetězce, výpočetní geometrie, …

- *… Výběr primárně dle počtu bodů z EP1*

Jak jste na tom s body?

- Budou doplněny do Moodle:
  - Body za první 3 sady úloh a první týden 4.
  - Bonusové body za CTU Open

- *Zkontrolujte si body a nesrovnalosti hlaste e-mailem*