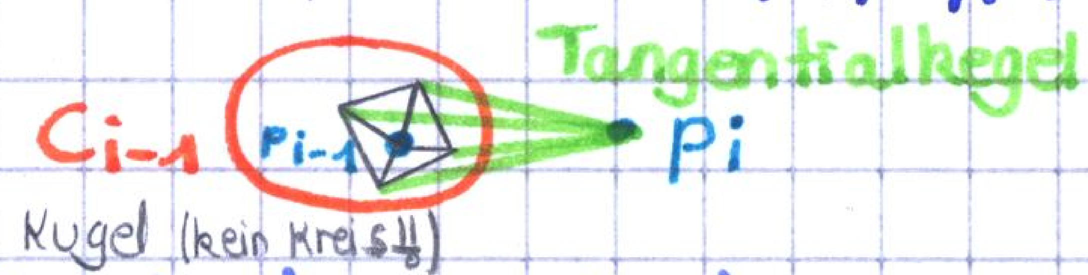
# 6.2.2 Algorithmus I: Inkrementeller Alg. (Sweep)

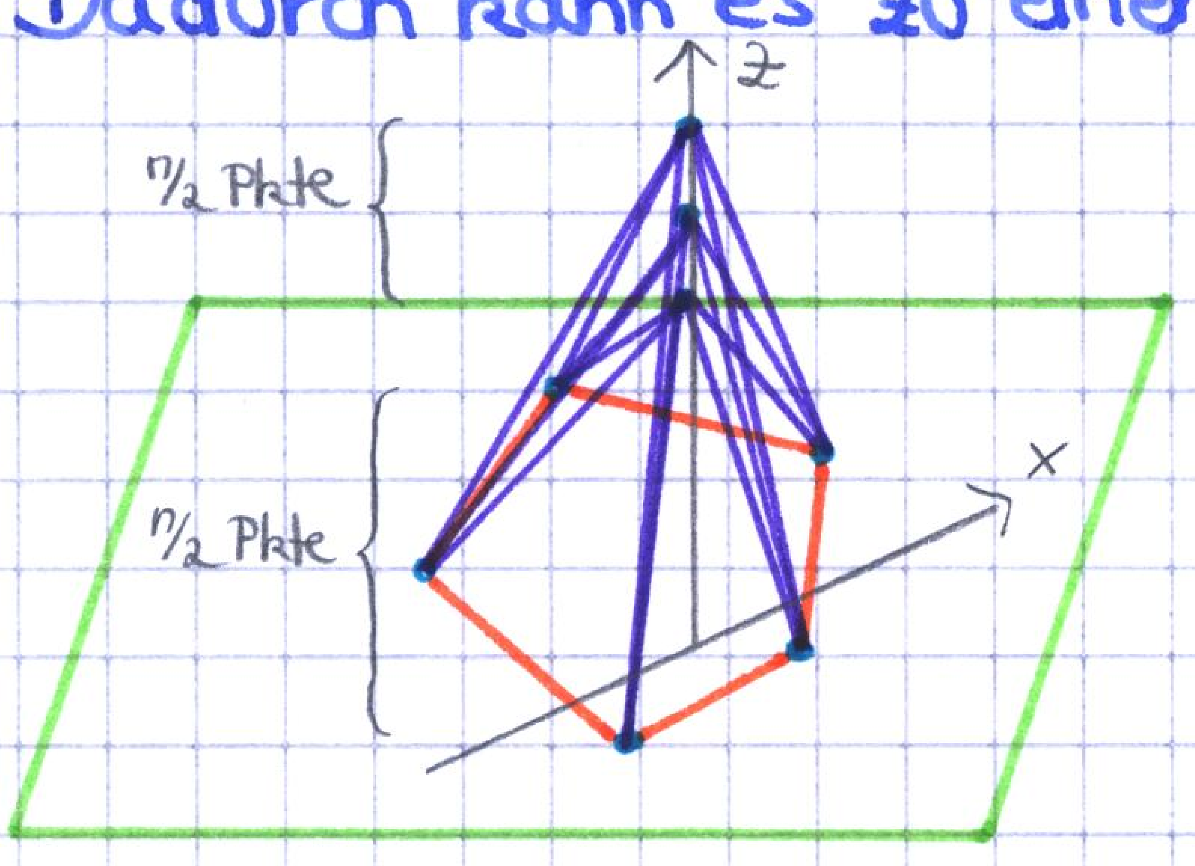
Der Algorithmus verläuft analog zum Algorithmus im $\mathbb{R}^2$.
Siehe: Varianten von Graham's Scan ⇝ Berechnung der gesamten Hülle in einer Phase

- Algorithmus:
    - Sortiere Pkte xyz-lexikogr (bzw konv Hülle)
    - Erstelle planaren Graphen aus den ersten 4 Pktn
    - Bearbeite $p_i$ $i = 5, \dots, n$,
      wobei $C_{i-1} = CH(\{p_1, \dots, p_{i-1}\})$ als Oberflächengraph vorliegt.




  Wir starten im Knoten $p_{i-1}$ und durchsuchen den Graphen nach allen Dreiecken $a, b, c$ mit orientation $(a, b, c, p_i) \geq 0$.
  Das sind alle von $p_i$ sichtbaren Dreiecke.
  Von diesen Dreiecken entfernen wir alle Kanten und alle danach isolierten Knoten. ⇝ Am Rand entstehen Kanten ohne Gegenkanten
  Für jede Kante $(v, w)$ auf dem Rand fügen wir ein Dreieck $(w, v, p_i)$ hinzu. Es entsteht ein tangentialer Kegel mit Spitze $p_i$.

- Laufzeit:
  Im $\mathbb{R}^2$ ist der "neue Rand" bzw die # der eingefügten Tangenten konstant, nämlich 2 !
  Im $\mathbb{R}^3$ müssen wir normalerweise viel mehr als 2 Tangenten konstruieren.
  Dadurch kann es zu einer Laufzeit von $O(n^2)$ kommen. worst case




  - Dieses Problem kann umgangen werden, wenn wir in eine andere Richtung zB nach x anstatt nach z sortieren.
  - Randomnisierung: Sweep in zufällige Richtung
    ⇔ Drehung der Pktmenge um zufälligen Winkel
  
  ⇒ erwartete Laufzeit: $O(n \log n)$

- Vorteil des Algorithmus:
    - einfach
    - praktisch
    - effizient

- Nachteil des Algorithmus:
  $O(n^2)$ Laufzeit im schlechtesten Fall.
  In der Praxis wird der Alg trotzdem eingesetzt, weil er für viele Eingaben doch relativ schnell ist und einfach zu implementieren ist.