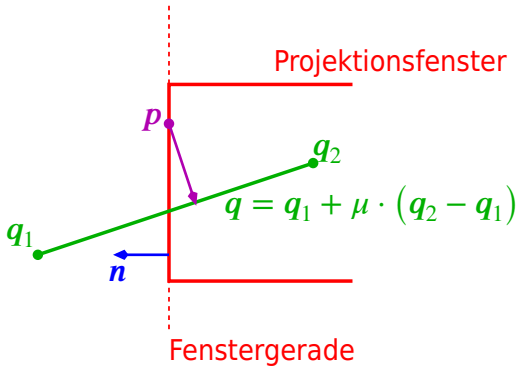
5.2.3 Strecken-Clipping nach Cyrus, Beck, Liang und Barsky

Idee: die Strecke sukzessive durch Schneiden mit den Fenstergeraden verkürzen, wie bei Cohen/Sutherland

aber: Schnittpunkte aus der Parameterdarstellung der Strecke gewinnen,

denn: **Wenn der Parameterwert des Schnittpunkts bekannt ist, muss der Schnittpunkt selbst oft gar nicht mehr explizit berechnet werden!**



Seien

- p ein beliebiger (aber fester) Punkt auf der Fenstergeraden,
- n ein **nach außen** weisender Normalenvektor zur Fenstergeraden.

Ein Punkt $q = q_1 + \mu\,(q_2 - q_1)$ der Geraden $q_1 q_2$ liegt genau dann auf der Fenstergeraden, wenn

$$\begin{aligned} & q - p \;\perp\; n \\ \Leftrightarrow\; & n^T \cdot (q_1 + \mu\,(q_2 - q_1) - p) = 0 \\ \Leftrightarrow\; & n^T \cdot (q_1 - p) + \mu \cdot n^T \cdot (q_2 - q_1) = 0 \\ \Leftrightarrow\; & \mu = \frac{n^T \cdot (q_1 - p)}{-n^T \cdot (q_2 - q_1)}\,. \end{aligned}$$

Der Schnittpunkt wird klassifiziert als:

In, falls $-n^T \cdot (q_2 - q_1) > 0$

(d. h. wenn man von q_1 nach q_2 läuft, geht man über die Fenstergerade in die Halbebene, in der auch das Projektionsfenster liegt)

Out, falls $-n^T \cdot (q_2 - q_1) < 0$

⇒ Ist q ein In-Schnittpunkt, so kann der Abschnitt $\overline{q_1 q}$ der Strecke weggeworfen werden, bei einem Out-Schnittpunkt der Abschnitt $\overline{q q_2}$.

Algorithmus 5.7:

Algorithmus Cyrus, Beck, Liang, Barsky

$[\mu_{In}; \mu_{Out}] := [0;1]$ // μ_{In} und μ_{Out} begrenzen den übrig gebliebenen Teil der Strecke
$\Delta u := u_2 - u_1$
$\Delta v := v_2 - v_1$
// der Reihe nach an den „O"-, „N"-, „W"- und „S"-Seiten des Projektionsfensters abschneiden
wenn noch_etwas_übrig($u_1 - \overline{u}$, $-\Delta u$, μ_{In}, μ_{Out})
  wenn noch_etwas_übrig($v_1 - \overline{v}$, $-\Delta v$, μ_{In}, μ_{Out})
    wenn noch_etwas_übrig($\underline{u} - u_1$, Δu, μ_{In}, μ_{Out})
      wenn noch_etwas_übrig($\underline{v} - v_1$, Δv, μ_{In}, μ_{Out})
        /* die Strecke ist – zumindest teilweise – sichtbar;
           berechne die Endpunkte des sichtbaren Teils */
        wenn $\mu_{Out} < 1$

$$q_2 := q_1 + \mu_{Out} \cdot \underbrace{\begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}}_{=q_2 - q_1}$$

        wenn $\mu_{In} > 0$

$$q_1 := q_1 + \mu_{In} \cdot \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}$$

Algorithmus 5.8:

Algorithmus noch_etwas_übrig(Z, N, μ_{In}, μ_{Out})

/* führt das durch Zähler Z und Nenner N gegebene Clipping an einer Fenstergeraden durch und
   modifiziert μ_{In} oder μ_{Out} entsprechend.
   Zurückgegeben wird, ob jetzt noch ein Teil der Strecke sichtbar ist. */
wenn $N > 0$ // In-Schnittpunkt

$$\mu_{In} := \max\left\{\mu_{In}, \underbrace{\frac{Z}{N}}_{\mu}\right\}$$

  noch_etwas_übrig := $(\mu_{In} \le \mu_{Out})$
sonst wenn $N < 0$ // Out-Schnittpunkt
  $\mu_{Out} := \min\left\{\mu_{Out}, \frac{Z}{N}\right\}$
  noch_etwas_übrig := $(\mu_{In} \le \mu_{Out})$
sonst // Nenner $N = 0$: Strecke parallel zur Fenstergeraden
  noch_etwas_übrig := $(Z \le 0)$
    // Strecke liegt innerhalb/außerhalb der Halbebene, in der auch das Fenster liegt