Schnittpunkte

Mario Konrad
Mario.Konrad@gmx.net

16. November 2006

Inhaltsverzeichnis

1 Zweier Geraden: Allgemeiner Fall in 2D

1.1 Herleitung

Schnittpunkt zweier Geraden, definiert durch je zwei Vektoren:

$$\overrightarrow{g} = \overrightarrow{a} + u \cdot (\overrightarrow{b} - \overrightarrow{a}) \tag{1}$$

$$\overrightarrow{h} = \overrightarrow{c} + t \cdot (\overrightarrow{d} - \overrightarrow{c}) \tag{2}$$

Nun sollen u und t gesucht werden:

$$a_1 + u \cdot (b_1 - a_1) = c_1 + t \cdot (d_1 - c_1) \tag{3}$$

$$a_2 + u \cdot (b_2 - a_2) = c_2 + t \cdot (d_2 - c_2) \tag{4}$$

Wenden wir uns zuerst u zu:

$$a_1 + u \cdot (b_1 - a_1) - c_1 = t \cdot (d_1 - c_1)$$
$$a_2 + u \cdot (b_2 - a_2) - c_2 = t \cdot (d_2 - c_2)$$

Um u zu erhalten, eliminieren wir t durch Gleichsetzen:

$$[a_1 + u \cdot (b_1 - a_1) - c_1] \cdot (d_2 - c_2) = [a_2 + u \cdot (b_2 - a_2) - c_2] \cdot (d_1 - c_1)$$

Substituion: $p = (d_1 - c_1)$, $q = (d_2 - c_2)$, $r = (b_1 - a_1)$, $s = (b_2 - a_2)$:

$$\begin{aligned}
(a_1 + u \cdot r - c_1) \cdot q &= (a_2 + u \cdot s - c_2) \cdot p \\
a_1 \cdot q + u \cdot r \cdot q - c_1 \cdot q &= a_2 \cdot p + u \cdot s \cdot p - c_2 \cdot p \\
u \cdot r \cdot q - u \cdot s \cdot p &= -a_1 \cdot q + c_1 \cdot q + a_2 \cdot p - c_2 \cdot p \\
u \cdot (r \cdot q - s \cdot p) &= q \cdot (c_1 - a_1) + p \cdot (a_2 - c_2) \\
u &= \frac{q \cdot (c_1 - a_1) + p \cdot (a_2 - c_2)}{r \cdot q - s \cdot p}
\end{aligned}$$

Substitution auflösen:

$$u = \frac{(d_2 - c_2) \cdot (c_1 - a_1) + (d_1 - c_1) \cdot (a_2 - c_2)}{(b_1 - a_1) \cdot (d_2 - c_2) - (b_2 - a_2) \cdot (d_1 - c_1)}$$

$$\Longrightarrow \qquad u = \frac{(d_1 - c_1) \cdot (a_2 - c_2) - (d_2 - c_2) \cdot (a_1 - c_1)}{(d_2 - c_2) \cdot (b_1 - a_1) - (d_1 - c_1) \cdot (b_2 - a_2)} \tag{5}$$

Um t zu finden verfahren wir analog zu oben, ausgehend von den Formeln x und y.

$$a_1 + u \cdot (b_1 - a_1) = c_1 + t \cdot (d_1 - c_1)$$
$$a_2 + u \cdot (b_2 - a_2) = c_2 + t \cdot (d_2 - c_2)$$

daraus wird:

$$u \cdot (b_1 - a_1) = c_1 + t \cdot (d_1 - c_1) - a_1$$
$$u \cdot (b_2 - a_2) = c_2 + t \cdot (d_2 - c_2) - a_2$$

Elimination von u durch Gleichsetzen:

$$[c_1 + t \cdot (d_1 - c_1) - a_1] \cdot (b_2 - a_2) = [c_2 + t \cdot (d_2 - c_2) - a_2] \cdot (b_1 - a_1)$$

Substitution: $p = (b_1 - a_1)$, $q = (b_2 - a_2)$, $r = (d_1 - c_1)$, $s = (d_2 - c_2)$

$$\begin{aligned}
(c_1 + t \cdot r - a_1) \cdot q &= (c_2 + t \cdot s - a_2) \cdot p \\
c_1 \cdot q + t \cdot r \cdot q - a_1 \cdot q &= c_2 \cdot p + t \cdot s \cdot p - a_2 \cdot p \\
t \cdot r \cdot q - t \cdot s \cdot p &= c_2 \cdot p - a_2 \cdot p - c_1 \cdot q + a_1 \cdot q \\
t \cdot (r \cdot q - s \cdot p) &= p \cdot (c_2 - a_2) - q \cdot (c_1 - a_1) \\
t &= \frac{p \cdot (c_2 - a_2) - q \cdot (c_1 - a_1)}{r \cdot q - s \cdot p}
\end{aligned}$$

Substitution auflösen:

$$t = \frac{(b_1 - a_1) \cdot (c_2 - a_2) - (b_2 - a_2) \cdot (c_1 - a_1)}{(d_1 - c_1) \cdot (b_2 - a_2) - (d_2 - c_2) \cdot (b_1 - a_1)}$$

$$\Longrightarrow \qquad t = \frac{(b_1 - a_1) \cdot (a_2 - c_2) - (b_2 - a_2) \cdot (a_1 - c_1)}{(d_2 - c_2) \cdot (b_1 - a_1) - (d_1 - c_1) \cdot (b_2 - a_2)} \tag{6}$$

Nun kann der Schnittpunkt P errechnet werden, unter der Verwendung von u oder t. Sollte der Nenner bei der Berechnung von u oder t gleich 0 sein, dann liegen die Geraden parallel zueinander und somit ist kein Schnittpunkt vorhanden.

$$\vec{p} = \vec{a} + u \cdot (\vec{b} - \vec{a}) \qquad \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + u \cdot \begin{pmatrix} b_1 - a_1 \\ b_2 - a_2 \end{pmatrix} \tag{7}$$

oder:

$$\vec{p} = \vec{c} + t \cdot (\vec{d} - \vec{c}) \qquad \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + t \cdot \begin{pmatrix} d_1 - c_1 \\ d_2 - c_2 \end{pmatrix} \tag{8}$$

1.2 Umsetzung in C

Optimierter C Code (floating point Operationen):

```
typedef double[2] Vector;

Vector a = { ... }, b = { ... };
Vector c = { ... }, d = { ... };
Vector p = { 0.0, 0.0 };

double d2c2 = d[1]-c[1];
double d1c1 = d[0]-c[0];
double b1a1 = b[0]-a[0];
double b2a2 = b[1]-a[1];
double a1c1 = a[0]-c[0];
double a2c2 = a[1]-c[1];
double den = d2c2 * b1a1 - d1c1 * b2a2;
double u = (d1c1 * a2c2 - d2c2 * a1c1) / den;
double t = (b1a1 * a2c2 - b2a2 * a1c1) / den;
```

Die Berechnung von $\vec{p}$:

```
p[0] = a[0] + u * b1a1;
p[1] = a[1] + u * b2a2;
```

2 Zweier Geraden: Vereinfachung 2D

2.1 Herleitung

Folgende Vereinfachung kann gemacht werden, wenn $\overrightarrow{c}$ gleich dem Nullvektor wird ($\overrightarrow{c} = \overrightarrow{0}$), d.h. die zweite Gerade durch den Ursprung geht. Somit gilt: $c_i = 0$.

$$\overrightarrow{g} = \overrightarrow{a} + u \cdot (\overrightarrow{b} - \overrightarrow{a}) \tag{9}$$

$$\overrightarrow{h'} = t \cdot \overrightarrow{d} \tag{10}$$

$$u = \frac{(d_1 - c_1) \cdot (a_2 - c_2) - (d_2 - c_2) \cdot (a_1 - c_1)}{(d_2 - c_2) \cdot (b_1 - a_1) - (d_1 - c_1) \cdot (b_2 - a_2)}$$

$$\Longrightarrow \qquad u = \frac{d_1 \cdot a_2 - d_2 \cdot a_1}{d_2 \cdot (b_1 - a_1) - d_1 \cdot (b_2 - a_2)} \tag{11}$$

$$t = \frac{(b_1 - a_1) \cdot (a_2 - c_2) - (b_2 - a_2) \cdot (a_1 - c_1)}{(d_2 - c_2) \cdot (b_1 - a_1) - (d_1 - c_1) \cdot (b_2 - a_2)}$$

$$\Longrightarrow \qquad t = \frac{a_2 \cdot (b_1 - a_1) - a_1 \cdot (b_2 - a_2)}{d_2 \cdot (b_1 - a_1) - d_1 \cdot (b_2 - a_2)} \tag{12}$$

2.2 Umsetzung in C

Der optimierte C Code:

```
typedef double[2] Vector;

Vector a = { ... }, b = { ... };
Vector d = { ... };
Vector p = { 0.0, 0.0 };

double b1a1 = b[0]-a[0];
double b2a2 = b[1]-a[1];

double den = d[1] * b1a1 - d[0] * b2a2;
double u = (d[0] * a[1] - d[1] * a[0]) / den;
double t = (a[1] * b1a1 - a[0] * b2a2) / den;
```

Die Berechnung von $\vec{p}$:

```
p[0] = a[0] + u * b1a1;
p[1] = a[1] + u * b2a2;
```

oder auch:

```
p[0] = t * c[0];
p[1] = t * c[1];
```

3 Einer Geraden und einer Fläche (3D)

3.1 Herleitung

Gegeben sind eine Gerade:

$$\vec{g} = \vec{d} + u \cdot (\vec{e} - \vec{d}) \tag{13}$$

und eine Fläche:

$$\vec{f} = \vec{a} + \mu \cdot (\vec{b} - \vec{a}) + \lambda \cdot (\vec{c} - \vec{a}) \qquad \vec{n} = (\vec{b} - \vec{a}) \times (\vec{c} - \vec{a}) \tag{14}$$

Die Ebenengleichung lautet:

$$A \cdot x + B \cdot y + C \cdot z + D = 0 \tag{15}$$

mit

$$\vec{n} = \begin{pmatrix} A \\ B \\ C \end{pmatrix} = (\vec{b} - \vec{a}) \times (\vec{c} - \vec{a})$$

Die Komponente D kann auch mit weniger Aufwand als der obigen Determinante gefunden werden:

$$-D = \vec{n} \circ \vec{g} \quad \text{mit } \vec{g} \text{ als Punkt auf der Ebene, z.B. } \vec{a}$$

$$\Longrightarrow \quad D = (\vec{n} \circ \vec{a}) = -\sum_{i=1}^{3} n_i \cdot a_i = -(A \cdot a_1 + B \cdot a_2 + C \cdot a_3)$$

Die Koordinaten (x, y, z) in der Ebenengleichung beschreibt einen Punkt auf der Ebene. Um den Schnittpunkt der Geraden (13) mit der Ebene zu finden, muss folgendes gelten:

$$A \cdot g_1 + B \cdot g_2 + C \cdot g_3 + D = 0$$

eingesetzt die Gleichung der Geraden:

$$A \cdot [d_1 + u \cdot (e_1 - d_1)] + B \cdot [d_2 + u \cdot (e_2 - d_2)] + C \cdot [d_3 + u \cdot (e_3 - d_3)] + D = 0$$
$$A \cdot d_1 + A \cdot u \cdot (e_1 - d_1) + B \cdot d_2 + B \cdot u \cdot (e_2 - d_2) + C \cdot d_3 + C \cdot u \cdot (e_3 - d_3) + D = 0$$
$$u \cdot [A \cdot (e_1 - d_1) + B \cdot (e_2 - d_2) + C \cdot (e_3 - d_3)] = -A \cdot d_1 - B \cdot d_2 - C \cdot d_3 - D$$

$$\Rightarrow \quad u = -\frac{A \cdot d_1 + B \cdot d_2 + C \cdot d_3 + D}{A \cdot (e_1 - d_1) + B \cdot (e_2 - d_2) + C \cdot (e_3 - d_3)} = -\frac{\left(\vec{n} \circ \vec{d}\right) + D}{\vec{n} \circ \left(\vec{e} - \vec{d}\right)} \tag{16}$$

Der Schnittpunkt wird dann mit Hilfe der Geradengleichung (13) berechnet. Sollte der Nenner gleich 0 sein, dann ist die Gerade parallel zur Fläche und somit kein Schnittpunkt vorhanden.

3.2 Umsetzung in C

Der optimierte C Code:

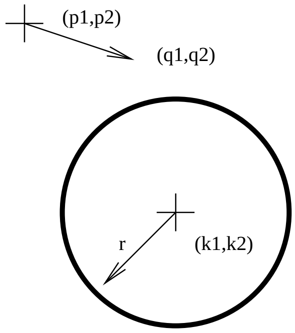
```
typedef double[3] Vector;

Vector a = { ... }, b = { ... }, c = { ... };
Vector d = { ... }, e = { ... };
Vector p = { ... };

Vector ed;
Vector ba, ca;

ed[0] = e[0]-d[0];
ed[1] = e[1]-d[1];
ed[2] = e[2]-d[2];

ba[0] = b[0]-a[0]; ba[1] = b[1]-a[1]; ba[2] = b[2]-a[2];
ca[0] = c[0]-a[0]; ca[1] = c[1]-a[1]; ca[2] = c[2]-a[2];

double A = ba[1]*ca[2]-ba[2]*ca[1];
double B = ba[2]*ca[0]-ba[0]*ca[2];
double C = ba[0]*ca[1]-ba[1]*ca[0];
double D = -(A*a[0] + B*a[1] + C*a[2]);
double den = (A*ed[0] + B*ed[1] + C*ed[2]);
double u = -(A*d[0] + B*d[1] + C*d[2] + D) / den;
```

Schnittpunkt:

```
p[0] = d[0] + u * ed[0];
p[1] = d[1] + u * ed[1];
p[2] = d[2] + u * ed[2];
```

4 Einer Geraden und einem Kreis 2D

Gegeben ist ein Kreis $k(\vec{m}, r)$ mit dem laufenden Punkt $\vec{k}$ und eine Gerade, definiert durch zwei Vektoren: $\vec{g} = \vec{p} + u \cdot \vec{q}$.



4.1 Herleitung

Der Kreis kann auch durch eine Gleichung beschrieben werden:

$$(k_1 - x_1)^2 + (k_2 - x_2)^2 = r^2$$

Der Schnittpunkt, falls es ihn gibt, befindet sich an der Stelle:

$$(k_1 - p_1 - u \cdot q_1)^2 + (k_2 - p_2 - u \cdot q_2)^2 - r^2 = 0 \tag{17}$$

Auflösen nach u als Normalform einer quadratischen Gleichung:

$$u^2 \underbrace{\sum_{i=1}^{2} q_i^2}_{a} + u \cdot \underbrace{2 \sum_{i=1}^{2} q_i (k_i - p_i)}_{b} + \underbrace{\sum_{i=1}^{2} (k_i - p_i)^2 - r^2}_{c} = 0$$

und wir erhalten eine quadratische Geleichung. Diese kann gelöst werden durch:

$$u_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

mit

$$a = \sum_{i=1}^{2} q_i^2 \qquad = q_1^2 + q_2^2 \tag{18}$$

$$b = 2 \sum_{i=1}^{2} q_i (k_i - p_i) \qquad = 2q_1 (k_1 - p_1) + 2q_2 (k_2 - p_2) \tag{19}$$

$$c = -r^2 + \sum_{i=1}^{2} (k_i - p_i)^2 = (k_1 - p_1)^2 + (k_2 - p_2)^2 - r^2 \tag{20}$$

Entscheidend ist die Diskriminante (der Term unter der Quadratwurzel) da sie entscheidet wieviele Schnittpunkte existieren:

$$b^2 - 4 \cdot a \cdot c \begin{cases} < 0 & : \text{kein Schnittpunkt, Passante} \\ = 0 & : \text{ein Schnittpunkt, Tangente} \\ > 0 & : \text{zwei Schnittpunkte, Sekante} \end{cases}$$

Diskriminante (bereits Eingesetzt und Umgeformt):

$$d = 4r^2 \left(q_1^2 + q_2^2 \right) - 4 \left[q_1 \left(k_2 - p_2 \right) + q_2 \left(k_1 - p_1 \right) \right]^2$$

Da die Berechnung von d in dieser Form mehr Rechenoperationen erfordert als die Berechnung der Komponenten a, b, c und dann daraus d, werden wir diese Form nicht verwenden. Wir schreiben daher einfach:

$$d = b^2 - 4ac \tag{21}$$

Die drei genannten Fälle müssen nun unterschieden werden.

4.1.1 Kein Schnittpunkt - Die Passante

Da $d < 0$ gibt es keinen Schnittpunkt.

4.1.2 Ein Schnittpunkt - Die Tangente

Da $d = 0$ existiert genau ein Schnittpunkt an der Stelle

$$\vec{s} = \vec{p} + u \cdot \vec{q} \qquad \text{mit} \qquad u = \frac{-b}{2 \cdot a} \tag{22}$$

dabei sind a und b wie unter (18) und (19) beschrieben.

4.1.3 Zwei Schnittpunkte - Die Sekante

Da $d > 0$ so schneidet die Gerade den Kreis an zwei Stellen

$$u_1 = \frac{-b + \sqrt{d}}{2a} \qquad u_2 = \frac{-b - \sqrt{d}}{2a} \tag{23}$$

mit a, b und d wie unter (18), (19) und (4.1) beschrieben.

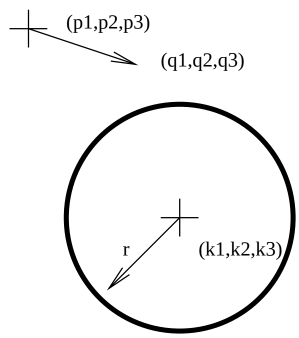
4.2 C Source Code

```
#define sqr(a)  ((a)*(a))

typedef double Vector[2];

Vector p = { ... };
Vector q = { ... };
```

```
Vector k = { ... };
double r = ... ;

double a, b, c, d;
double u1 = 0.0, u2 = 0.0;
Vector s = { k[0]-p[0], k[1]-p[1] };

a = sqr(q[0]) + sqr(q[1]);
b = 2.0 * (q[0] * s[0] + q[1] * s[1]);
c = sqr(s[0]) + sqr(s[1]) - sqr(r);
d = sqr(b) - 4.0 * a * c;

if (d < 0.0)
{
    printf("Passante\n");
}
else if (d == 0.0)
{
    u1 = u2 = (-b) / (2.0 * a);
    printf("Tangente, u=%f\n", u1);
}
else // d > 0.0
{
    u1 = (-b + sqrt(d)) / (2.0 * a);
    u2 = (-b - sqrt(d)) / (2.0 * a);
    printf("Sekante, u1=%f u2=%f\n", u1, u2);
}
```

5 Einer Geraden und einer Kugel 3D

Gegeben ist eine Kugel $s(\vec{m}, r)$ mit dem laufenden Punkt $\vec{s}$ und eine Gerade, definiert durch zwei Vektoren: $\vec{g} = \vec{p} + u \cdot \vec{q}$



5.1 Herleitung

Analog zum Kreis kann die Kugel durch eine Gleichung beschrieben werden:

$$(k_1 - x_1)^2 + (k_2 - x_2)^2 + (k_3 - x_3)^2 = r^2$$

Der Schnittpunkt, falls es ihn gibt ist demnach:

$$\vec{x} = \vec{g} \implies \sum_{i=1}^{3} (k_i - g_i)^2 = r^2 \tag{24}$$

$$\sum_{i=1}^{3} (k_i - p_i - u \cdot q_i)^2 = r^2$$

$$\sum_{i=1}^{3} u^2 q_i^2 - 2 \cdot u \cdot q_i (k_i - p_i) + (k_i - p_i)^2 = r^2$$

Auflösen nach u als Normalform einer quadratischen Gleichung:

$$u^2 \underbrace{\sum_{i=1}^{3} q_i^2}_{a} - u \cdot \underbrace{2 \sum_{i=1}^{3} q_i (k_i - p_i)}_{b} + \underbrace{r^2 + \sum_{i=1}^{3} (k_i - p_i)^2}_{c} = 0 \tag{25}$$

Diese Gleichung kann wiederum nach u gelöst werden durch:

$$u_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

mit

$$a = \sum_{i=1}^{3} q_i^2 = q_1^2 + q_2^2 + q_3^2 \tag{26}$$

$$b = -2 \sum_{i=1}^{3} q_i \left(k_i - p_i\right) = -2 \left[q_1 \left(k_1 - p_1\right) + q_2 \left(k_2 - p_2\right) + q_3 \left(k_3 - p_3\right)\right] \tag{27}$$

$$c = -r^2 + \sum_{i=1}^{3} \left(k_i - p_i\right)^2 = \left(k_1 - p_1\right)^2 + \left(k_2 - p_2\right)^2 + \left(k_3 - p_3\right)^2 - r^2 \tag{28}$$

Zu Beachten ist der unscheinbare Schritt von 2D auf 3D. Die Berechnung der Komponenten a, b und c ist in beiden Fällen gleich, es werden immer alle Komponenten der Vektoren mit einbezogen (2D sind es 2, in 3D sind es 3).

Analog zu oben betrachten wir die Diskriminante um die Fallunterscheidung zu machen und die Anzahl der Schnittpunkte (0, 1, 2) zu finden:

$$b^2 - 4ac \begin{cases} < 0 & : \text{keinen Schnittpunkt} \\ = 0 & : \text{einen Schnittpunkt} \\ > 0 & : \text{zwei Schnittpunkte} \end{cases}$$

In diesem Fall macht es keinen Sinn mehr die Determinante auszumultiplizieren und Umzuformen, da zu viele Komponenten ins Spiel kommen. Darum schreiben wir einfach:

$$d = b^2 - 4ac \tag{29}$$

a, b und c sind unter (26), (27) und (28) definiert.

5.1.1 Keinen Schnittpunkt

Da $d < 0$ gibt es keinen Schnittpunkt.

5.1.2 Einen Schnittpunkt

Da $d = 0$ gibt es genau einen Schnittpunkt an der Stelle

$$\vec{s} = \vec{p} + u \cdot \vec{q} \qquad \text{mit} \qquad u = \frac{-b}{2a} \tag{30}$$

dabei sind a und b wie unter (26) und (27) beschrieben.

5.1.3 Zwei Schnittpunkte

Da $d > 0$ gibt es genau zwei Schnittpunkte an den Stellen

$$\vec{s_1} = \vec{p} + u_1 \cdot \vec{q} \qquad \text{und} \qquad \vec{s_2} = \vec{p} + u_2 \cdot \vec{q}$$
$$u_1 = \frac{-b + \sqrt{d}}{2a} \qquad u_2 = \frac{-b - \sqrt{d}}{2a} \tag{31}$$

mit a, b und d wie unter (26), (27) und (29) beschrieben.

5.2 C Source Code

```
typedef double Vector[3];

Vector p = { ... };
Vector q = { ... };
Vector k = { ... };
double r = ... ;

dobuel a, b, c, d;
double u1 = 0.0, u2 = 0.0;

Vector s = { p[0]-k[0], p[1]-k[1], p[2]-k[2] };

double a = sqr(q[0]) + sqr(q[1]) + sqr(q[2]);
double b = -2.0 * (q[0]*s[0] + q[1]*s[1] + q[2]*s[2]);
double c = sqr(s[0]) + sqr(s[1]) + sqr(s[2]) - sqr(r);
double d = sqr(b) - 4.0 * a * c;

if (d < 0.0)
{
    printf("Keinen Schnittpunkt\n");
}
else if (d == 0.0)
{
    u1 = u2 = b / (2.0 * a);
    printf("Einen Schnittpunkt, u=%f\n", u1);
}
else
{
    u1 = (-b + sqrt(d)) / (2.0 * a);
    u2 = (-b - sqrt(d)) / (2.0 * a);
    printf("Zwei Schnittpunkte, u1=%f u2=%f\n", u1, u2);
}
```