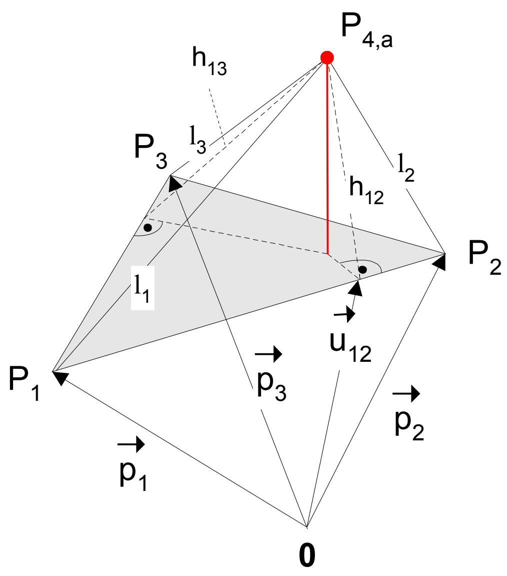
Triangulation

geg.: 3 Punkte P_1, P_2, und P_3, spezifiziert durch Ortsvektoren $\vec{p}_1$, $\vec{p}_2$, $\vec{p}_3$

ges.: Die Punkte $P_{4,a}$ und $P_{4,b}$, die von den Punkten P_1, P_2 und P_3 die Entfernungen l_1, l_2 und l_3 aufweisen.

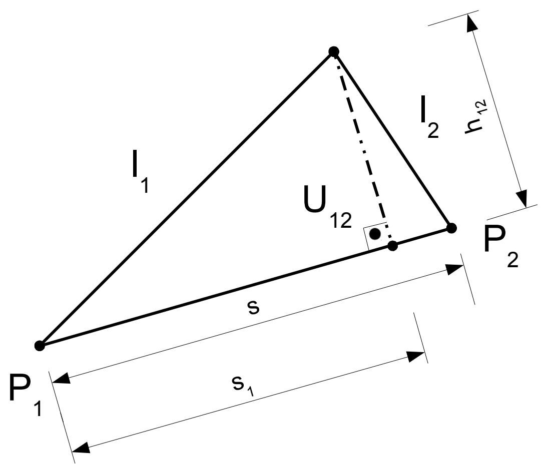


$$s^2=|\vec{p}_2-\vec{p}_1|^2$$
$$h_{12}^2=l_1^2-s_1^2\,;\;h_{12}^2=l_2^2-(s-s_1)^2$$

$$=> l_1^2-s_1^2=l_2^2-(s-s_1)^2$$
$$=> l_1^2=l_2^2-s^2+2\,s\,s_1$$

$$=> s_1=\frac{l_1^2-l_2^2+s^2}{2s}$$
$$=> \kappa:=\frac{s_1}{s}=\frac{l_1^2-l_2^2+s^2}{2\,s^2}$$
$$\vec{n}_{12}=\vec{p}_2-\vec{p}_1$$
$$\vec{u}_{12}=\vec{p}_1+\kappa\,\vec{n}_{12}=\vec{p}_1+\frac{l_1^2-l_2^2+s^2}{2\,s^2}(\vec{p}_2-\vec{p}_1)$$

$$=> \vec{u}_{12}=\vec{p}_1+\kappa\,\vec{n}_{12}=\vec{p}_1+\frac{l_1^2-l_2^2+|p_2-p_1|^2}{2|p_2-p_1|^2}(\vec{p}_2-\vec{p}_1)$$



Ebene 1-2 wird somit definiert zu:

$$E_{1-2}:\vec{n}_{12}\vec{r}-d_{12}=0$$
$$d_{12}=\vec{n}_{12}\vec{u}_{12}$$

Entsprechend gilt für Ebene 1-3

$$E_{1-3}:\vec{n}_{13}\vec{r}-d_{13}=0$$
$$d_{13}=\vec{n}_{13}\vec{u}_{13}$$

Die Schnittgerade der beiden Ebenen lautet:

$$g:\vec{\rho}+\lambda\cdot\vec{v}$$
$$\vec{v}=\vec{n}_{12}\times\vec{n}_{13}$$

Der Vektor $\vec{\rho}$ ergibt sich aus der Lösung des LGS

$$\begin{bmatrix} n12_x & n12_y & n12_z \\ n13_x & n13_y & n13_z \end{bmatrix}\cdot\vec{\rho}=\begin{bmatrix} d_{12} \\ d_{13} \end{bmatrix}$$

Nun sind die Werte für die Punkte $P_{4,a}$ und $P_{4,b}$ gesucht, die die Abstände l_1, l_2 und l_3 zu den Punkten P_1, P_2 und P_3 haben.

$$|\vec{\rho}+\lambda_{1/2}\cdot\vec{v}-\vec{p}_1|=l_1^2$$
$$\vec{\rho}-\vec{p}_1=:\vec{u}$$

Lösen der quadratischen Gleichung:

$$u_x^2+u_y^2+u_z^2+2\cdot\lambda\left(u_x\cdot v_x+u_y\cdot v_y+u_z\cdot v_z\right)+\lambda^2\left(v_x^2+v_y^2+v_z^2\right)=l_1^2$$

$$=>|\vec{u}|^2+2\cdot\lambda\cdot\vec{u}\cdot\vec{v}^T+\lambda^2\cdot|\vec{v}|^2:=l_1^2$$

$$=>\lambda_{1/2}=-\frac{\vec{u}\cdot\vec{v}^T}{|\vec{v}|^2}\pm\sqrt{\left(\frac{\vec{u}\cdot\vec{v}^T}{|\vec{v}|^2}\right)^2-\frac{\left(|\vec{u}|^2-l_1^2\right)}{|\vec{v}|^2}}$$

$$\left\{\begin{matrix}P_{4,a}\\P_{4,b}\end{matrix}\right\}=\vec{\rho}+\lambda_{1/2}\cdot\vec{v}$$

SciLab-Realisierung:

```
function [p4a,p4b]=triangu3D(p1,p2,p3,l1,l2,l3);
// [p4a,p4b]=triangu3D(p1,p2,p3,l1,l2,l3);
//
// Given are three points in 3D, specified by the local vectors p1, p2 and p3
// Now the points p4a and p4b should be calculated, which have distance l1 to point p1,
// l2 to point p2 and l3 to point p3
// Instead of solving a nonlinear equation system, an algorithm based upon linear algebra is used.

flag0=0;
p4a=[];
p4b=[];

n12=p2-p1;                     // Vector from p1 to p2 (normal vector of plane 12)
n13=p3-p1;                     // Vector from p1 to p3 (normal vector of plane 13)
S12=sqrt(sum((p2-p1) .^2));    // distance between point p1 and p2
S13=sqrt(sum((p3-p1) .^2));    // distance between point p1 and p3
S23=sqrt(sum((p3-p2) .^2));    // distance between point p3 and p2

if l1+l2<S12, flag0=1; end
if l1+l3<S13, flag0=1; end
if l2+l3<S23, flag0=1; end

// Determine position of intersection of plane 12 with line p1-p2
k12=0.5 .*(1 +((l1 ./S12) .^2) -((l2 ./S12) .^2));
// Point of intersection Plane p12 with line p1-p2
u12=p1 + k12 .*n12;

// Determine position of intersection of plane 13 with line p1-p3
k13=0.5 .*(1 +((l1 ./S13) .^2) -((l3 ./S13) .^2));
// Point of intersection Plane p13 with line p1-p3
u13=p1 + k13 .*n13;

d12=n12*(u12.');               // Scalar value of plane 12 equation in normal form
d13=n13*(u13.');               // Scalar value of plane 13 equation in normal form

rg=[(n12(2)*n13(3)-n12(3)*n13(2)), -(n12(1)*n13(3)-n12(3)*n13(1)), (n12(1)*n13(2)-n12(2)*n13(1))];
if rg==[0,0,0], flag0=1; end

A=[n12;n13]; B=[d12;d13];
rho=(A\B).';
// intersection line of planes 1-2 and 1-3: rho + lambda * rg
// On this intersection line, the points p4 with distance l(i) to points p(i) can be found
// Now solving for lambda1 and lambda2
u=rho - p1;

z1=u*(rg.')/sum(rg .^2);
z2=(sum(u .^2) - (l1 .^2)) ./sum(rg .^2);
qval=(z1 .^2) - z2;
if qval<0; flag0=1; end
lam1=-z1 + sqrt( (z1 .^2) - z2);
lam2=-z1 - sqrt( (z1 .^2) - z2);

p4a=rho + lam1 .*rg; p4b=rho + lam2 .*rg;

if flag0~=0,
        p4a=[];
        p4b=[];
end

endfunction;
```