

Die Rolle der Geometrie bei Navigationssystemen

Hellmuth Stachel



TECHNISCHE
UNIVERSITÄT
WIEN

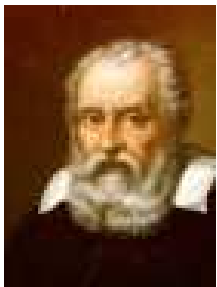


stachel@dmg.tuwien.ac.at — <http://www.geometrie.tuwien.ac.at/stachel>



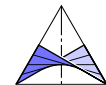
Inhalt

1. Positionierung und Navigation
2. Positionierung mittels Distanzen
3. Distanzen zu Satellitenpositionen
4. Navigation, Algorithmus von Dijkstra



Galileo Galilei:

“Wer die Geometrie begreift, vermag in dieser Welt alles zu verstehen”

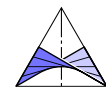


1. Positionierung und Navigation



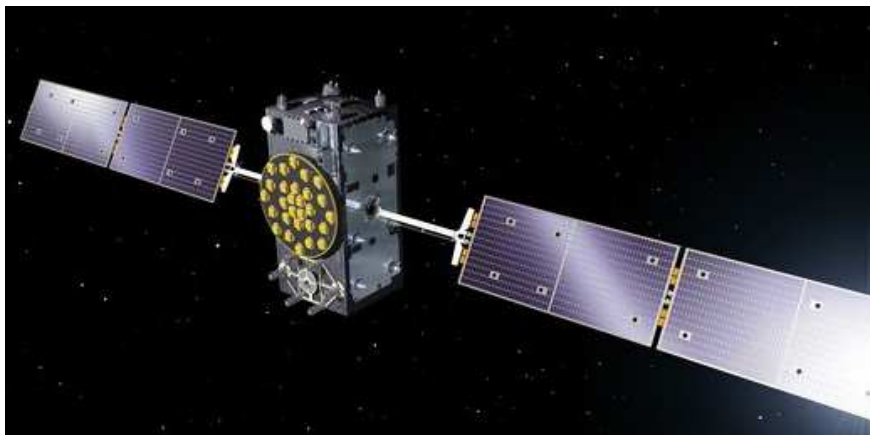
Seit dem Altertum war es eine große wissenschaftliche Herausforderung, sich auf dem Meer zurechtzufinden. Man konnte **Winkel messen** und orientierte sich nach Leuchttürmen, Sternen oder nach der Sonne.

Mit Hilfe des **Sextanten** wurde der **Erhebungswinkel der Sonne** gemessen. Zusammen mit Datum und Uhrzeit konnte man auf die geographische Länge und Breite schließen.



1. Positionierung und Navigation

Inzwischen wurde die Winkelmessung durch die **Distanzmessung** zu Satelliten abgelöst.

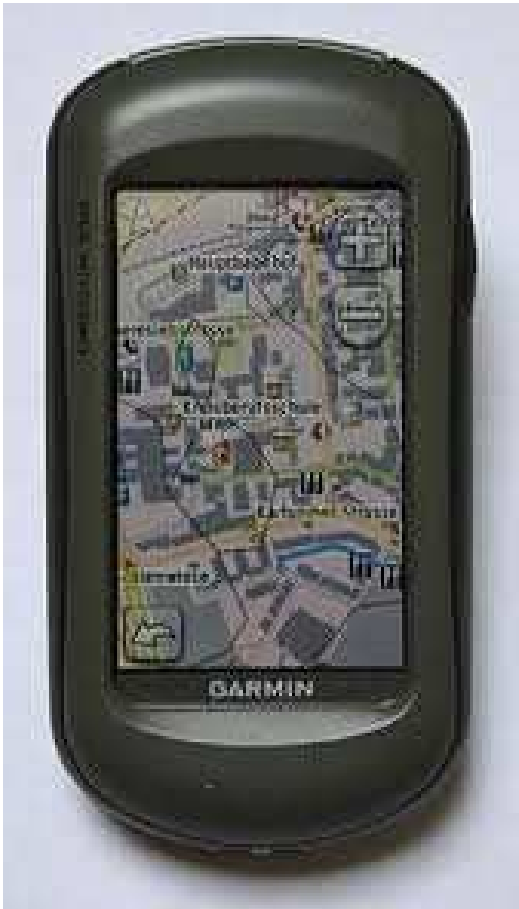


Satellit von Galileo
ca. 730 kg, 23.260 km hoch

Es gibt derzeit vier globale **Satelliten-Navigationssysteme** (GNSS):

1. **GPS** (Global Positioning System), USA
2. **GLONASS** (Globalnaja nawigazionaja sputnikowaja sistema), Russland
3. **BeiDou** 北斗 ('Großer Bär'), China
4. **Galileo**, EU, seit 2016, mit dzt. 18 Satelliten (bald 27 + 3 in Reserve). Es ist dies das einzige zivile System.

1. Positionierung und Navigation



Jedes **“Navi”** (Zielführungssystem) erledigt zwei Aufgaben, die **Positionierung** und die **Navigation**.

Es besteht aus mindestens vier Komponenten:

- einem **GNSS-Empfänger*** (GPS oder mehr),
- einer **digitalen Karte**,
- einem **Navigationcomputer** und
- einem Gerät zur **Eingabe und Anzeige**.

und allenfalls unterstützende Sensoren wie z.B. Rad-sensoren (in Tunnels).

*) GNSS = global navigation satellite system

2. Positionierung mittels Distanzen



Die Position des Empfängers X wird berechnet aus den Distanzen zu vier oder mehr Satelliten S_i , $i = 1, 2, \dots$

Somit wurde in der Navigation die **Winkelmessung** (Triangulation) abgelöst von der **Distanzmessung** (Trilateration).

2. Positionierung mittels Distanzen



einundzwanzig, zweiundzwanzig, ...

Distanzen mittels Laufzeitmessung des **Schalles**; dessen Geschwindigkeit (in der Luft) beträgt **0.340 km/s** (Mach 1 = 1.224 km/h).

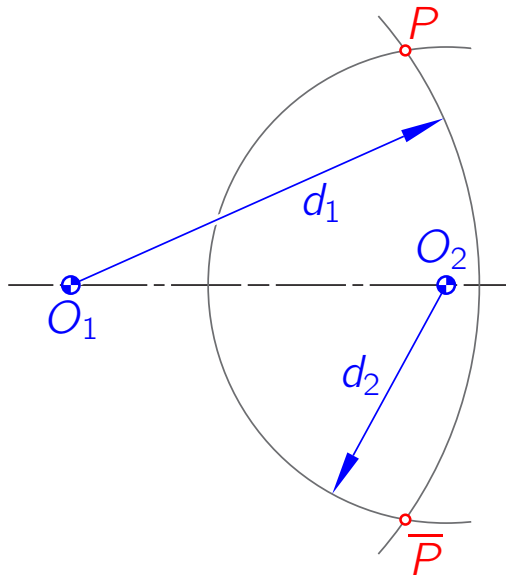
Bei der Satellitennavigation ist es die **Laufzeit des Funksignals** mit der Geschwindigkeit $c = 300.000 \text{ km/s}$.

Dabei betrachten wir die **Zeitansage** über Rundfunk als verbindlich; Funkuhren gelten als höchst präzise.

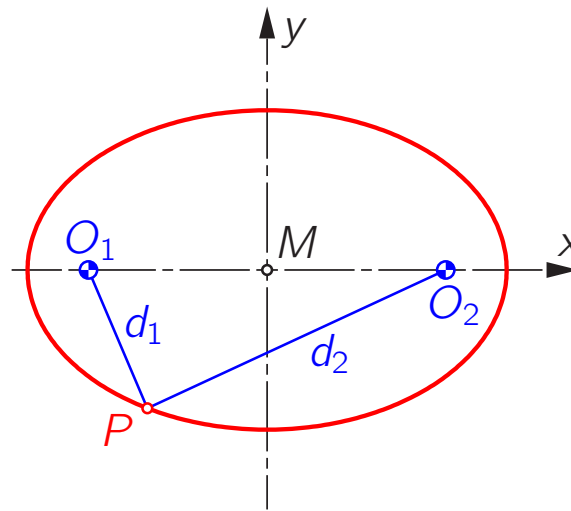
Ein Funksignal von Wien nach Innsbruck benötigt ca. 0.0013s. Das TV-Signal von Wien über Astra (ca. 36.000 km über Äquator) und zurück benötigt rund **0.5 s!**

2. Positionierung mittels Distanzen

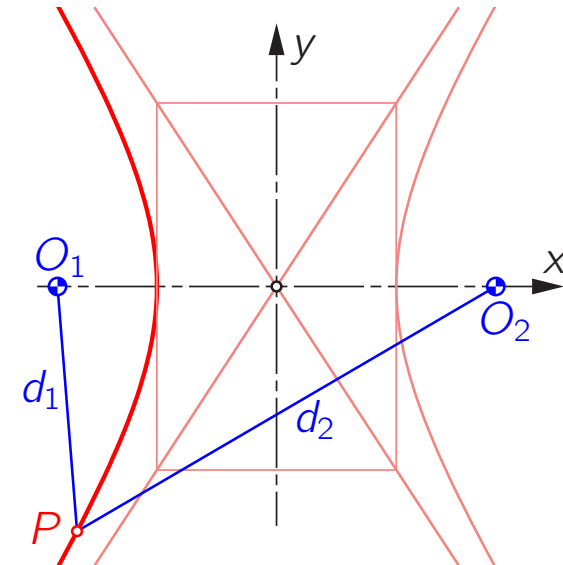
Gegeben: Punkte $O_1 \neq O_2$: **bipolare Koordinaten** (d_1, d_2) mit $d_i = \overline{O_i P}$
 bei $d_1 + d_2 \geq \overline{O_1 O_2}$ und $|d_1 - d_2| \leq \overline{O_1 O_2}$.



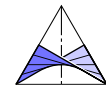
bipolare Koordinaten von P



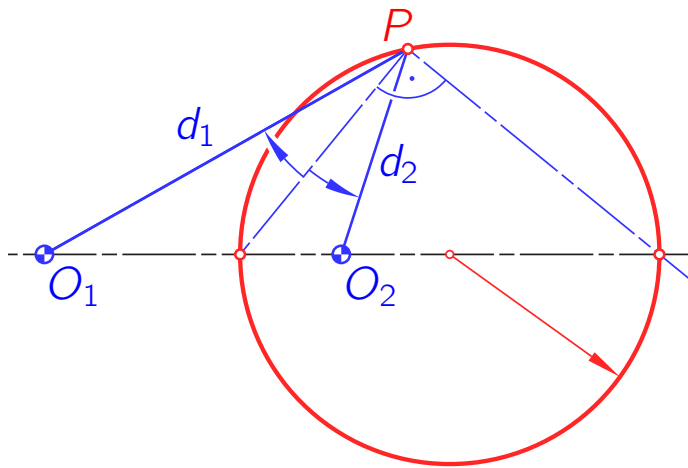
$d_1 + d_2 = \text{konst.}$



$d_2 - d_1 = \text{konst.} \neq 0$

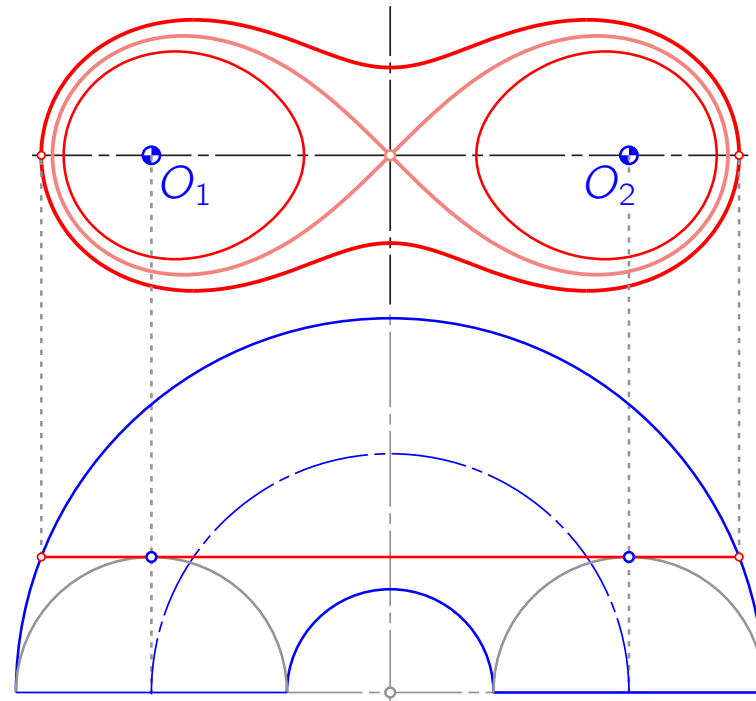


2. Positionierung mittels Distanzen



$d_1 : d_2 = \text{konst.}$
Kreis des Apollonius

Apollonius von Perga (262 – 190 v. Chr.)
heute in Provinz Antalya



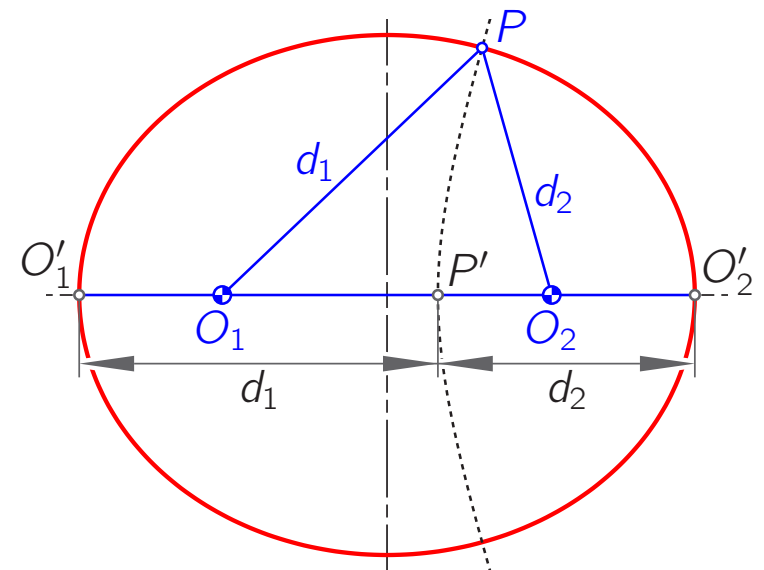
$d_1 \cdot d_2 = \text{konst.}$
Cassinische Kurven
Sonderfall: Bernoullische Lemniskate

2. Positionierung mittels Distanzen

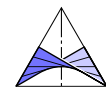
Gegeben: Dreieck $O_1O_2O_3$, **tripolare Koordinaten** (d_1, d_2, d_3) mit $d_i = \overline{O_iP}$
 $\iff F_{\Delta PO_1O_2} + F_{\Delta PO_2O_3} + F_{\Delta PO_3O_1} \geq F_{\Delta O_1O_2O_3}$ sowie Dreiecksungleichungen.

Jacobische Fokaleigenschaft der Quadriken:

Ort der Punkte P , deren tripolare Koordinaten d_i gleich sind den Entfernungen $d_i = \overline{O'_iP'}$ der Ecken eines Dreiecks $O'_1O'_2O'_3$ von einem in der Dreiecksebene gelegenen Punkt P' , ist eine Fläche zweiten Grades.



Rechts: Zweidimensionale Version der Jacobischen Fokaleigenschaft



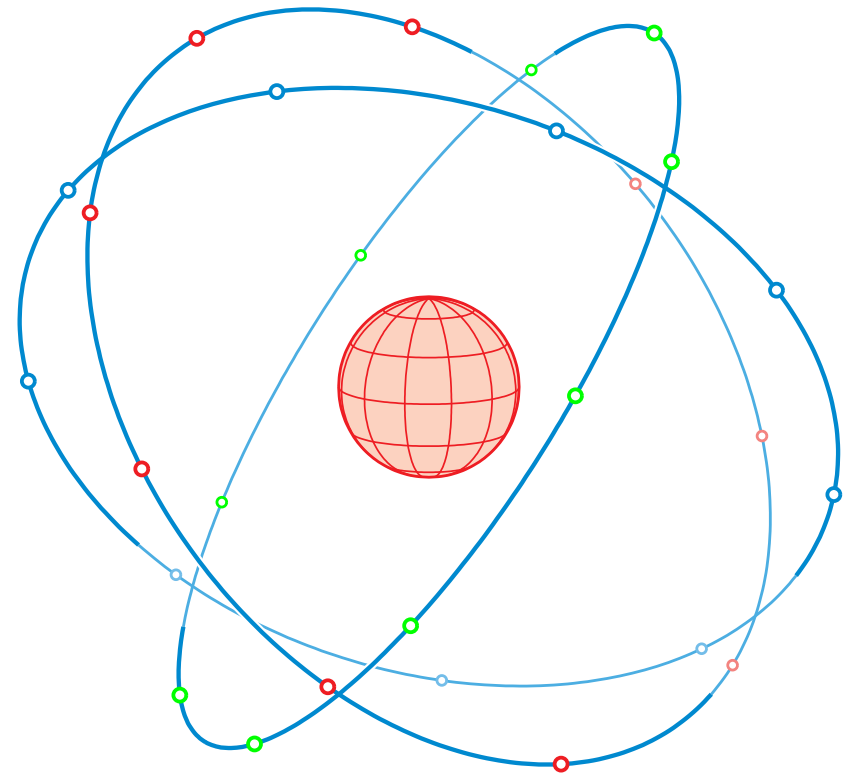
3. Distanzen zu Satellitenpositionen

Galileo Endausbau: 30 Satelliten, je 9 (+1 Reserve) verteilt auf 3 Bahnebenen (Neigungswinkel 56° gegenüber Äquator).

Umlaufzeit ~ 14 h; daher nach 10 Tagen (bzw. $240/14 \sim 17$ Umläufen) Wiederholung relativ zur Erde.

Sie laufen auf einer Kreisbahn in 23.260 km Höhe über der Erde mit einer Geschwindigkeit von 3.693 m/s (\sim Mach 3).

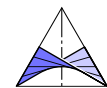
Fast überall (außer in Polnähe) sind 6 bis 8 Satelliten sichtbar.



3. Distanzen zu Satellitenpositionen

Bodenstationen informieren jeden Satellit über seine genauen Bahndaten und Zeitkorrekturen. Pro Satellit gibt es 4 Atomuhren, jeweils auf $1 \cdot 10^{-10}$ s genau; 1 Takt Laufzeit entspricht 3 cm Weg des Funksignals.

Die **Bahndaten** beziehen sich auf die in **WGS 84** (World Geodetic System) festgelegte Approximation der Erde durch ein **Drehellipsoid** mit Halbachsen $a = 6378.137$ km und $b = 6356.752$ km. Das Geoid, die überall zur Gravitation orthogonale Fläche, weicht bis zu ± 100 m von diesem Ellipsoid ab.



3. Distanzen zu Satellitenpositionen

Jede Navigationsmitteilung umfasst neben Kennung des Satelliten eine Zeit- und Positionsangabe (samt aktuellen Korrekturen für Zeitmessung und Bahndaten.)

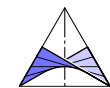
Der Satellit in Position \mathbf{s}_i funkt die Zeitansage "8:00 Uhr", und diese erreicht den Empfänger \mathbf{x} gemäß dessen Uhr mit der Verzögerung t_i , also um $8:00 + t_i$. Die Multiplikation mit der Lichtgeschwindigkeit c ergibt für die **scheinbare** Distanz \tilde{d}_i

$$\|\mathbf{s}_i - \mathbf{x}\| = \tilde{d}_i = c t_i.$$

Fehlerquelle: Die Atomuhren in den Satelliten sind sehr genau synchronisiert; bei den Empfängeruhren ist dies technisch nicht möglich. Geht die Empfängeruhr um t_0 vor, so erscheinen alle Distanzen um dasselbe $d_0 = c t_0$ vergrößert.

Deshalb lautet die **wahre** Distanz

$$d_i = \|\mathbf{s}_i - \mathbf{x}\| = \tilde{d}_i - d_0.$$



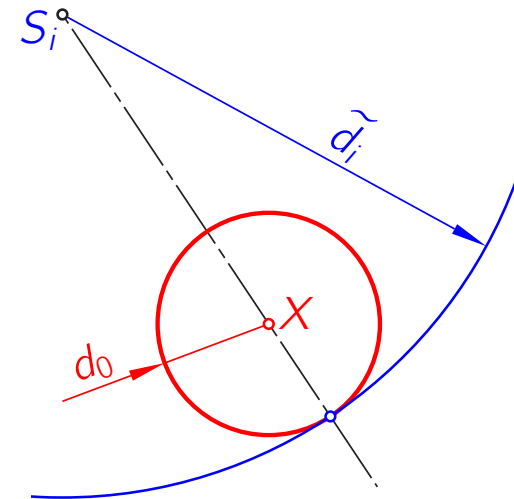
3. Distanzen zu Satellitenpositionen

wahre Distanzen: $\| \mathbf{s}_i - \mathbf{x} \| = \tilde{d}_i - d_0, \quad i = 1, \dots, 4.$

Gegeben: Satellitenpositionen S_1, \dots, S_4 sowie scheinbare Distanzen $\tilde{d}_1, \dots, \tilde{d}_4$.

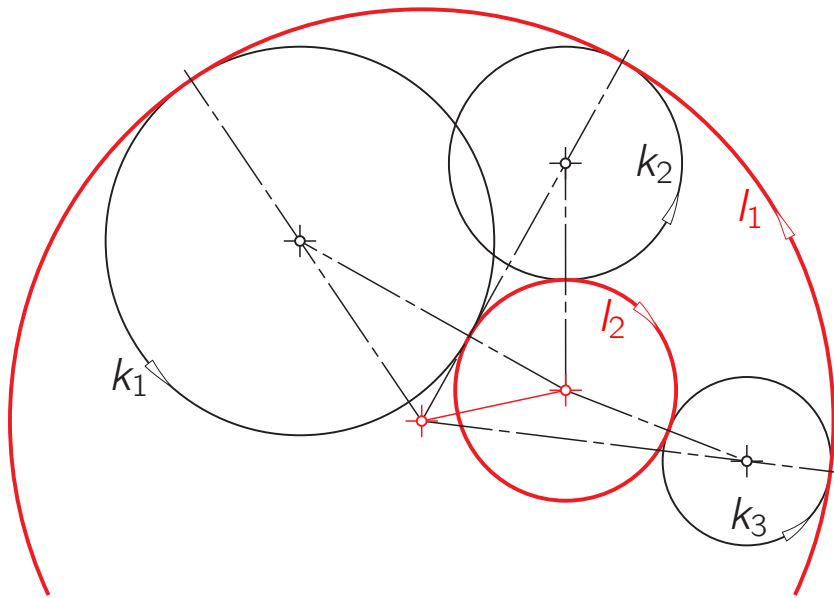
Gesucht: Die Raumkoordinaten des Empfängers X und der durch mangelnde Synchronisation der Empfängeruhr entstandene Distanzfehler $d_0 \lesseqgtr 0$.

Geometrisch: Gesucht ist ein **Kreis** (X, d_0) , welcher vier gegebene Kreise (S_i, \tilde{d}_i) **gleichartig berührt**, d.h., alle von außen oder alle von innen (**Apollonisches Berührproblem**).



3. Distanzen zu Satellitenpositionen

Apollonisches Berührproblem in der Ebene:



Version mit orientierten Kreisen

Gegeben: orientierte Kreise k_1, k_2, k_3

Gesucht: orientierte Berührkreise
(gleichsinnige Berührung).

Es gibt bis zu zwei reelle Lösungen.

Drei nicht-orientierte Kreise können auf $2^3 = 8$ verschiedene Arten orientiert werden, wobei komplementäre Orientierungen komplementäre Lösungen ergeben.

$\implies \exists$ bis zu 8 reelle Lösungen im nicht-orientierten Fall.

3. Distanzen zu Satellitenpositionen

Analytische Positionsbestimmung:

Gegeben: Vier Satellitenpositionen \mathbf{s}_i , $i = 1, \dots, 4$, samt scheinbaren Distanzen \tilde{d}_i .

Gesucht: Drei Koordinaten der Empfängerposition \mathbf{x} sowie der Distanzfehler d_0 .

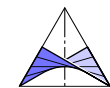
Die vier Unbekannten müssen vier quadratische Gleichungen

$$Q_i(\mathbf{x}, d_0) = (\mathbf{s}_i - \mathbf{x})^2 - (d_i - d_0)^2 = 0$$

lösen, also ausführlich

$$\mathbf{x} \cdot \mathbf{x} - 2(\mathbf{s}_i \cdot \mathbf{x}) + \mathbf{s}_i \cdot \mathbf{s}_i - d_0^2 + 2d_i d_0 - d_i^2 = 0.$$

Die Differenz je zweier Gleichungen ist linear in den Unbekannten \implies bleiben **drei lineare** und **eine quadratische** Gleichung, daher **maximal zwei** reelle Lösungen.



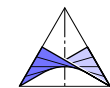
3. Distanzen zu Satellitenpositionen

Praktische Berechnung:

Die Positionsbestimmung wird wiederholt durchgeführt; also gibt es zumeist eine **gute Näherungslösung** \mathbf{x}^* für \mathbf{x} und d_0^* für d_0 . Andererseits stehen meist mehr als vier Satelliten zur Verfügung.

Wir substituieren $\mathbf{x} = \mathbf{x}^* + \mathbf{v}$ und $d_0 = d_0^* + v_0$ in den quadratischen Polynomen $Q_j(\mathbf{x}, d_0)$ und entwickeln diese an der Näherungslösung (\mathbf{x}^*, d_0^*) nach Taylor. Damit bleibt ein **lineares System** für die 'Verbesserungen' \mathbf{v} und v_0 .

- Das geht schneller. Es gibt kein Problem mit der Zweideutigkeit der Lösung.
- Es gibt Standardverfahren zur Bestimmung der **optimalen Lösung** im Fall eines überbestimmten Gleichungssystems, wenn mehr als vier Satelliten zur Verfügung stehen.



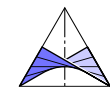
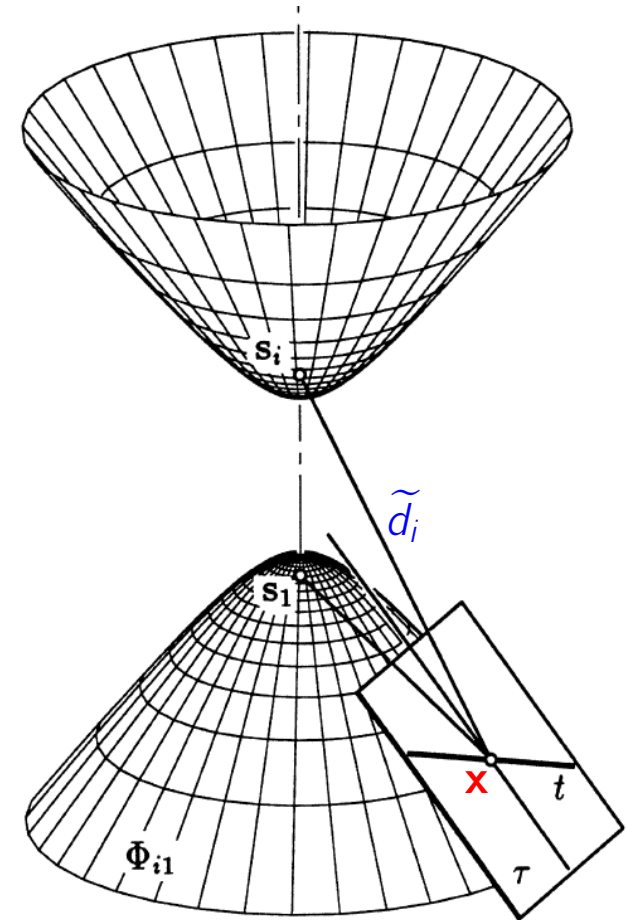
3. Distanzen zu Satellitenpositionen

Dieses approximative Verfahren wird ungenau, wenn die zwei Lösungen (des Apollonischen Problems) nahe beieinanderliegen. Wann fallen diese zusammen?

Wegen $\| \mathbf{s}_j - \mathbf{x} \| = \tilde{d}_j - d_0$ ist für den Empfänger \mathbf{x} die **Differenz der Abstände** von S_1 und S_i für $i = 2, 3, 4$ bekannt, nämlich $|\tilde{d}_i - \tilde{d}_1|$.

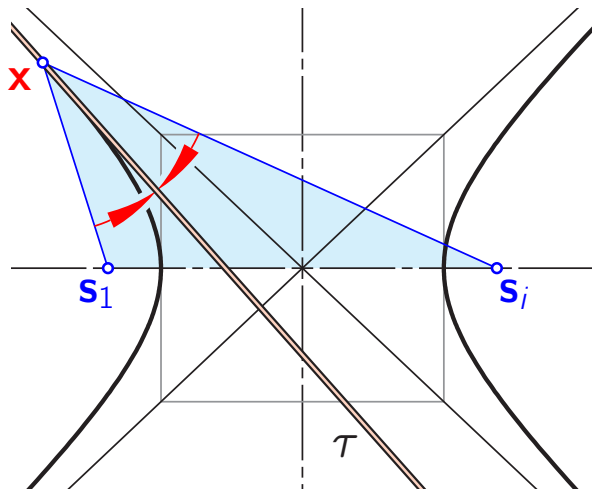
Also liegt \mathbf{x} auf einem **zweischaligen Drehhyperboloid** mit Brennpunkten S_j und S_1 . Somit ist \mathbf{x} ein Schnittpunkt dreier Drehhyperboloide.

In \mathbf{x} **fallen 2 Lösungen zusammen** \iff die Hyperboloide haben in \mathbf{x} **eine gemeinsame Tangente t** .



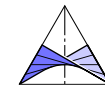
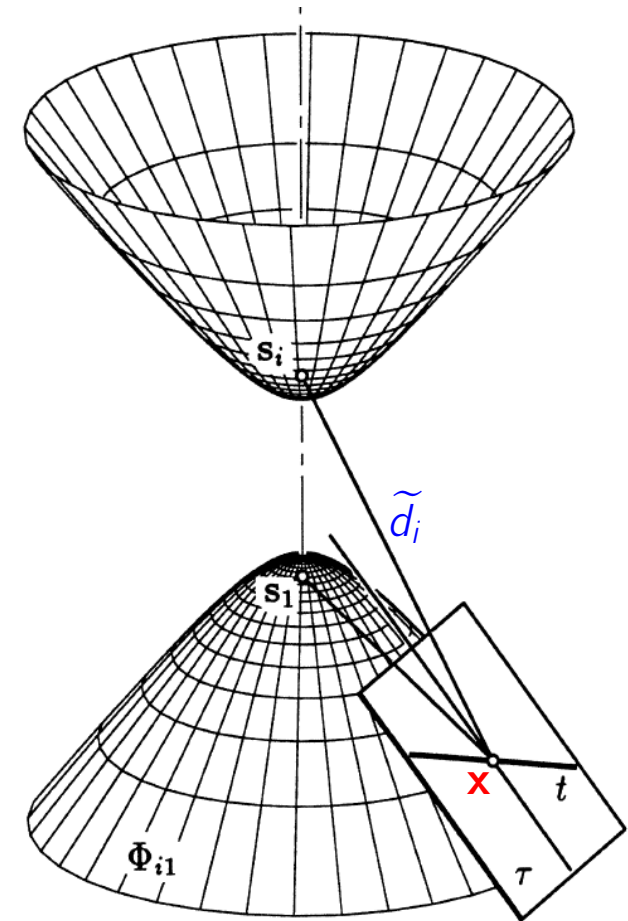
3. Distanzen zu Satellitenpositionen

In \mathbf{x} fallen 2 Lösungen zusammen \iff die Hyperboloide haben in \mathbf{x} eine gemeinsame Tangente t .



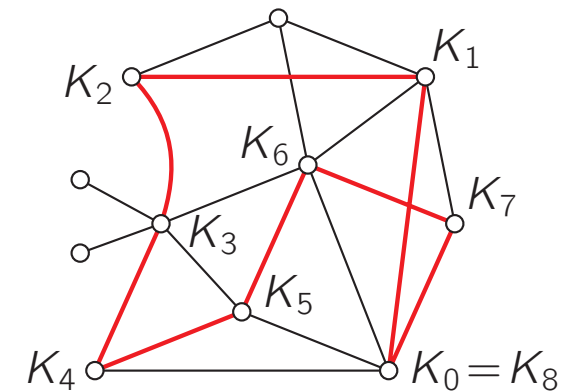
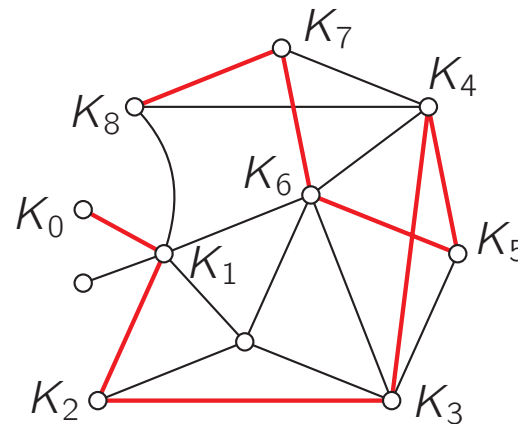
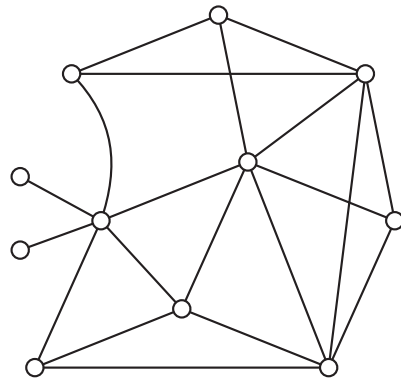
Die Verbindungsgeraden von \mathbf{x} mit \mathbf{s}_1 und \mathbf{s}_j gehen durch Spiegelung an der durch t gehenden Tangentialebene τ ineinander über.

2 Lösungen fallen zusammen \iff die Winkel zwischen t und den Verbindungsgeraden von \mathbf{x} und \mathbf{s}_j für $j = 1, \dots, 4$ sind gleich.



4. Navigation, Algorithmus von Dijkstra

kurz zur Graphentheorie:



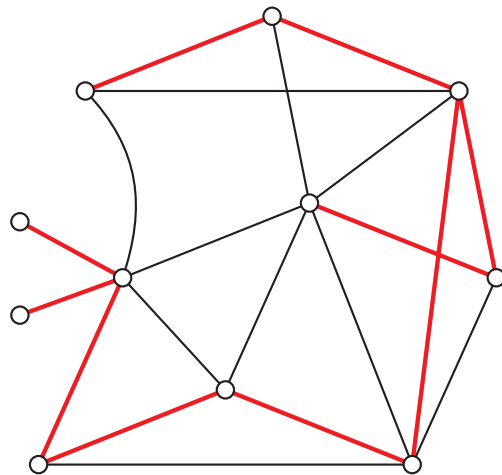
Ein **Graph** umfasst eine *Knotenmenge* \mathcal{V} und eine *Kantenmenge* $\mathcal{E} \subset \mathcal{V}^2$, wobei jede Kante von zwei Knoten begrenzt ist.

Ein **Weg** (von K_0 nach K_8) ist eine aus Kanten bestehende *Verbindung*, die keinen Knoten öfter als einmal enthält.

Ein **Kreis** ist eine aus Kanten zusammengesetzte *geschlossene* Verbindung, die keinen Knoten mehrfach enthält.

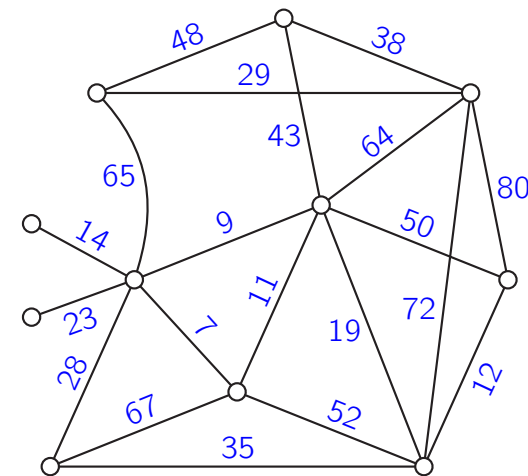
4. Navigation, Algorithmus von Dijkstra

kurz zur Graphentheorie:



Ein **Baum** ist ein zusammenhängender Graph ohne Kreis; genau dann hat er um 1 mehr Knoten als Kanten.

Ein **aufspannender Baum** erreicht alle Knoten eines gegebenen Graphen.



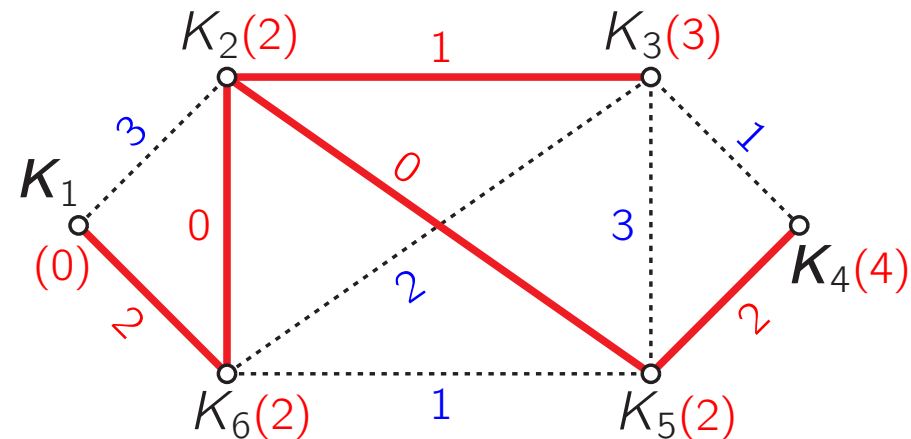
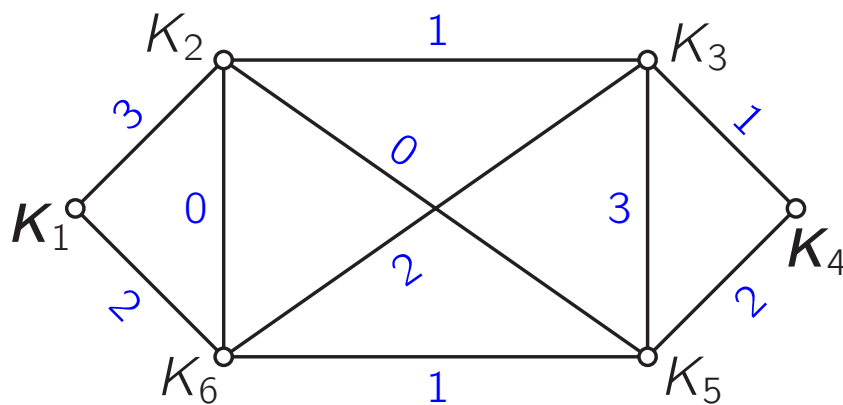
Ist jeder Kante des Graphen ein Wert zugeordnet, so heißt er **bewerteter Graph** oder **Netzwerk**. Ein K_0 mit K_i verbindender Weg mit kleinster Wertesumme heißt **Minimalweg**.

4. Navigation, Algorithmus von Dijkstra

Algorithmus von Edsger W. Dijkstra (1959):

Gegeben: Bewerteter Graph (mit nichtnegativen Bewertungen)

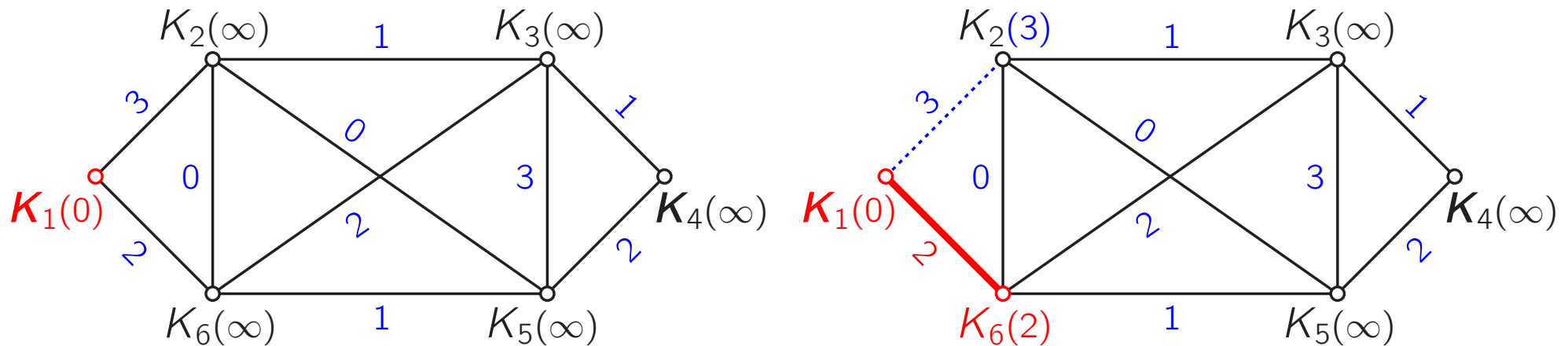
Gesucht: Minimalweg von K_1 bis K_4 , d.h., Weg mit kleinster Wertesumme.



Links ein **bewerteter Graph** und rechts ein **Entfernungsbaum** zum Knoten K_1 .

Die jeweiligen Distanzen $d(K_1, K_i)$ stehen in Klammern.

4. Navigation, Algorithmus von Dijkstra



Initialisierung

Jeder Knoten K_i bekommt einen Wert v_i zugeordnet, zuerst $v_1 = 0$ und $v_i = \infty$ für $i > 1$.

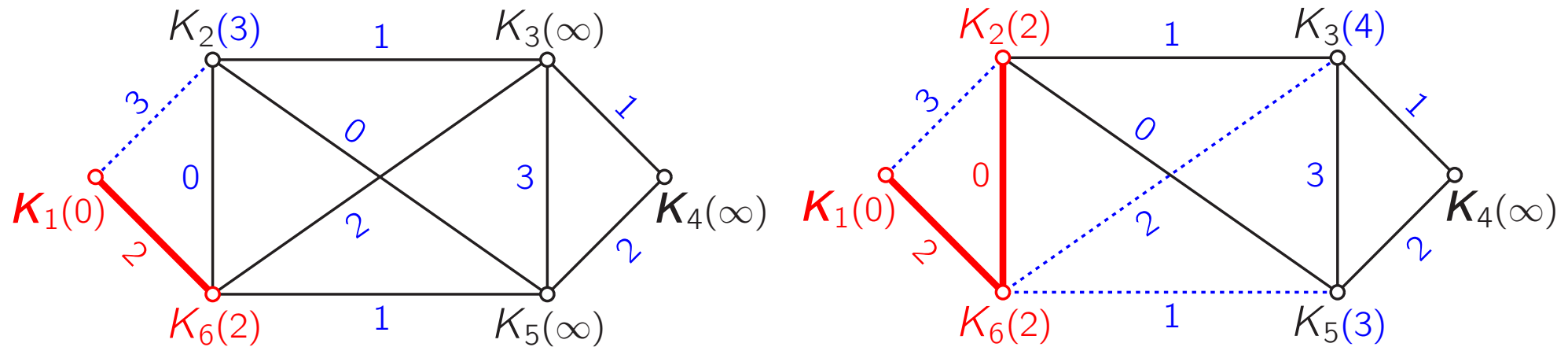
Bestimmt wird ein Teilgraph \mathcal{T} , der am Ende den Minimalweg enthält.

1. Schritt:

Für alle Nachbarn K_j von K_1 korrigieren wir deren Wert v_j auf die Länge der Verbindungskante mit K_1 .

Eine Kante minimaler Länge samt ihren Endpunkten kommt zu \mathcal{T} .

4. Navigation, Algorithmus von Dijkstra



1. Schritt:

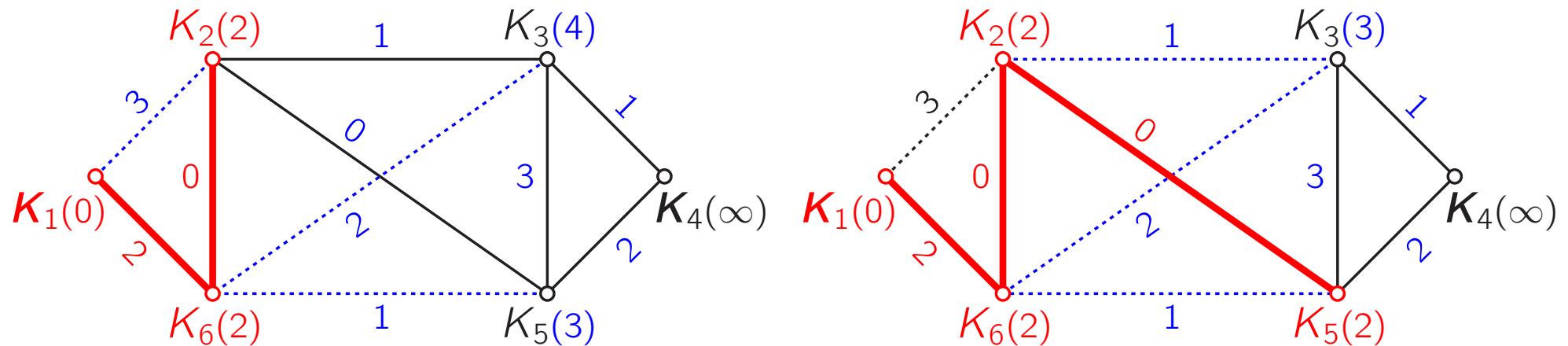
Für alle **Nachbarn** K_j von K_1 **korrigieren** wir deren Wert v_j auf die Länge der Verbindungskante mit K_1 .

Eine Kante **minimaler Länge** samt ihren Endpunkten kommt zu \mathcal{T} .

2. Schritt:

Für **Nachbarn** K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ **korrigiert**, sofern **kleiner** als bisheriges v_j . Ein K_j mit **minimalem** v_j kommt zu \mathcal{T} dazu samt Kante K_iK_j .

4. Navigation, Algorithmus von Dijkstra



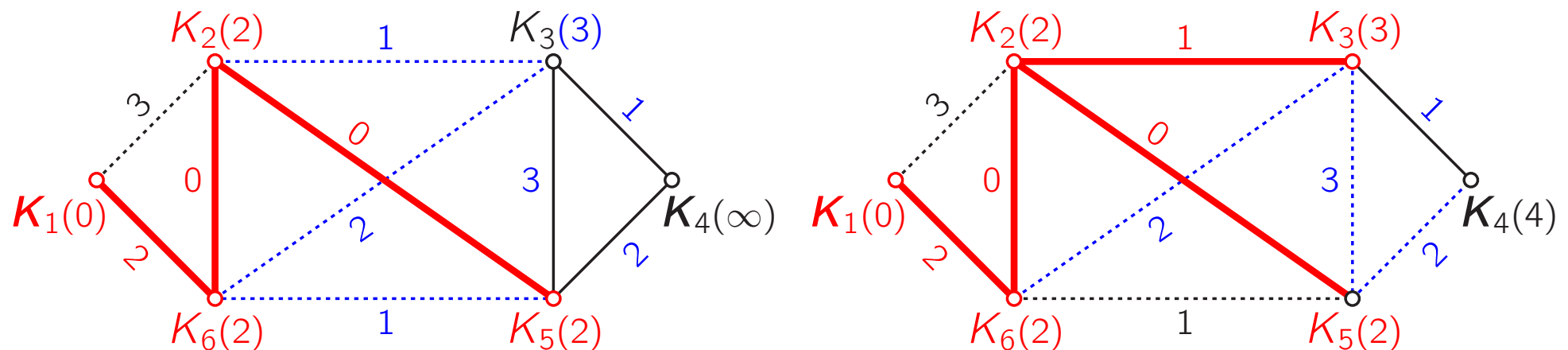
2. Schritt:

Für Nachbarn K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ korrigiert, sofern kleiner als bisheriges v_j . Ein K_j mit minimalem v_j kommt zu \mathcal{T} dazu samt Kante K_iK_j .

3. Schritt:

Für Nachbarn K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ korrigiert, sofern kleiner als bisheriges v_j . Ein K_j mit minimalem v_j kommt zu \mathcal{T} dazu samt Kante K_iK_j .

4. Navigation, Algorithmus von Dijkstra



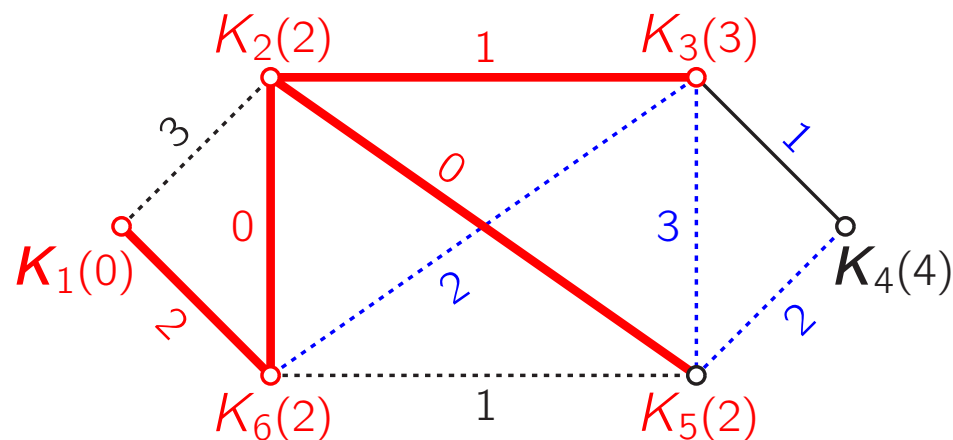
3. Schritt:

Für **Nachbarn** K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ **korrigiert**, sofern **kleiner** als bisheriges v_j . Ein K_j mit **minimalem** v_j kommt zu \mathcal{T} dazu samt Kante K_iK_j .

4. Schritt:

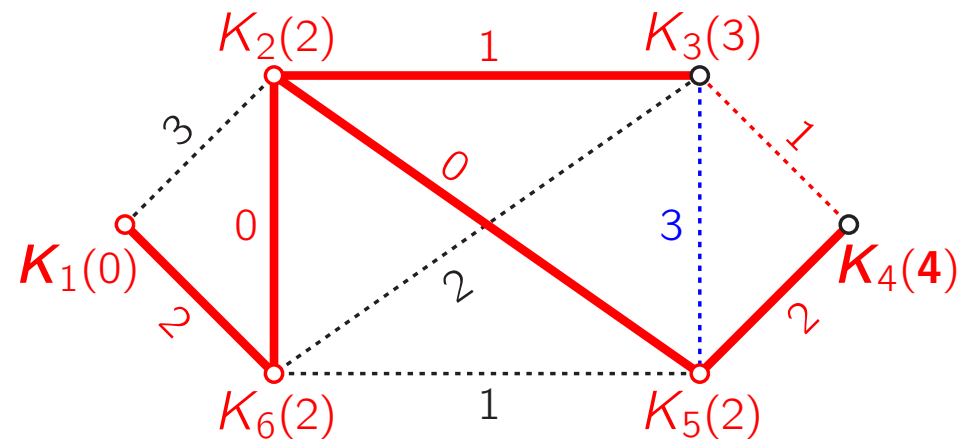
Für **Nachbarn** K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ **korrigiert**, sofern **kleiner** als bisheriges v_j . Ein K_j mit **minimalem** v_j kommt zu \mathcal{T} dazu samt Kante K_iK_j .

4. Navigation, Algorithmus von Dijkstra



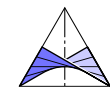
4. Schritt:

Für Nachbarn K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ korrigiert, sofern kleiner als bisheriges v_j . Ein K_j mit minimalem v_j kommt zu \mathcal{T} dazu samt Kante K_iK_j .



5. Schritt:

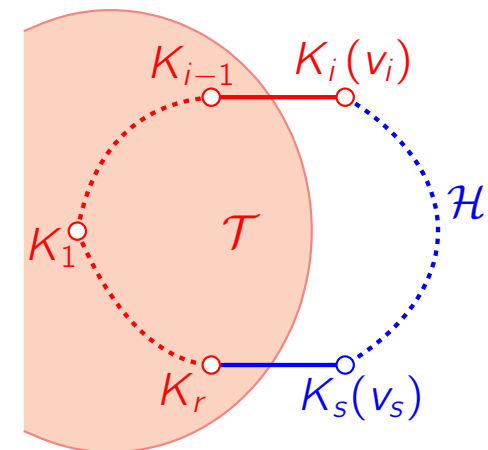
Für Nachbarn K_j der $K_i \in \mathcal{T}$ wird v_j auf die Summe aus v_i und der Distanz $d(K_iK_j)$ korrigiert, sofern diese kleiner als bisheriges v_j ist ...
... bis der Zielknoten K_4 erreicht ist.



4. Navigation, Algorithmus von Dijkstra

Satz: Für jeden Knoten $K_i \in \mathcal{T}$ ist der in dem Baum \mathcal{T} enthaltene Weg von K_1 nach K_i ein **Minimalweg**, und der Wert v_i gibt dessen **Länge** an.

Beweis: Angenommen, an Wegen von K_1 nach K_i gibt es neben jenem aus \mathcal{T} über K_{i-1} noch einen **kürzeren Weg** \mathcal{H} , der bis K_r in \mathcal{T} verläuft und dann außerhalb, von K_s an. Dann ist die Länge von \mathcal{H} die **Summe** aus dem Wert v_s von K_s , wobei $v_s \geq v_i$ nach dem Entscheidungskriterium, und der Länge ≥ 0 des Weges von K_s nach K_i , also insgesamt $\geq v_i$. **Widerspruch!**



Wird der Algorithmus fortgesetzt, bis alle Knoten erreicht sind, entsteht ein **Entfernungsbaum** \mathcal{T} zum Knoten K_1 , also ein aufspannender Baum, wobei für jedes i der Wert v_i des Knotens K_i die Länge des Minimalweges von K_1 und K_i angibt.

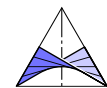


“La Géométrie”
Hôtel de Ville, Paris

Danke für Ihre
Aufmerksamkeit!

Literatur

- T. Arens, R. Busam, F. Hettlich, Ch. Karpfinger, H. Stachel: *Grundwissen Mathematikstudium — Analysis und Lineare Algebra mit Querverbindungen*. Spektrum Akademischer Verlag, Heidelberg 2013, 1172 S., 648 Abbn. u. 52 Tab., ISBN 978-3-8274-2308-5,
Kap. 26: *Elemente der diskreten Mathematik – die Kunst des Zählens*,
Kap. 26.1: *Einführung in die Graphentheorie*, S. 1076–1089.
- Ken Cheng: *Geometry and Navigation*. Springer-Verlag, Berlin Heidelberg 2008, ISBN 978-3-540-75386-5.
- E.W. Dijkstra: *A Note on Two Problems in Connexion with Graphs*. Numerische Mathematik **1**, 269–271 (1959).



- H. Dodel, D. Häupler: *Satellitenavigation*. 2. korr. u. erw. Aufl., Springer-Verlag, Berlin Heidelberg 2010, ISBN 978-3-540-79443-1, 978-3-540-79444-8 (eBook).
- Sh. Izumi: *Sufficiency of simplex inequalities*. Proc. Amer. Math. Soc. **144**, no. 3, 1299–1307 (2016).
- G. Glaeser: *Briefe der Geometrie an die Mathematik*. Informationsblätter der Geometrie **34**/1, 25–33 (2015).
- T. Schüttler: *Satellitenavigation*. Springer Vieweg, Berlin Heidelberg 2014, ISBN 978-3-642-53886-5, 978-3-642-53887-2 (eBook).
- H. Stachel: *Why shall we also teach the theory behind Engineering Graphics*. Institut für Geometrie, TU Wien, Technical Report **35** (1996).
- R. Williams: *Geometry of Navigation*. Horwood Publ., 1998, Sign. 9916.

