

Matematik Projekt F21

GPS og geometri

Skrevet af GPS39:

Magnus Meyer	s204509
Josephine Bøgh	s204430
Noah Bro-Jørgensen	s204500
Niels Raunkjær Holm	s204503
Victor Henning Lintz Østergaard	s204456
Emilie Zunda Primdahl Thaulow	s204471

16. April, 2021

DTU



Indhold

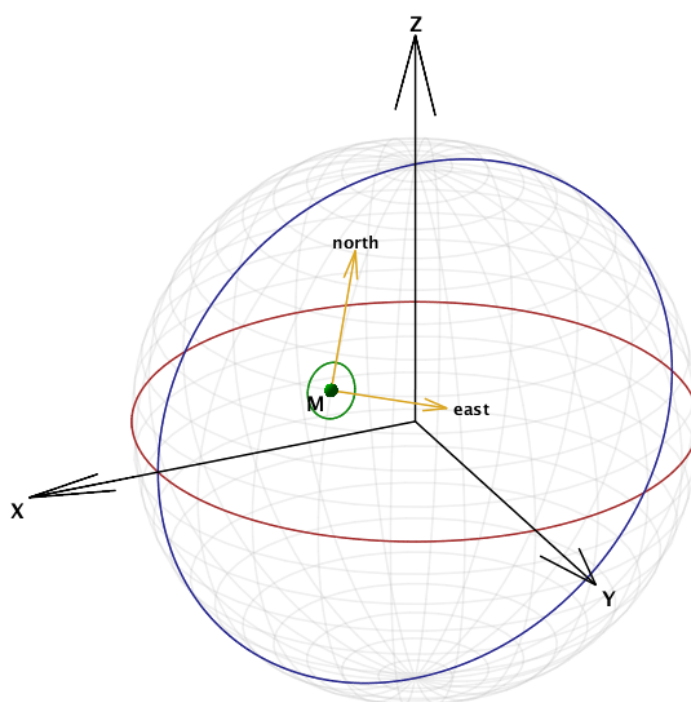
1	Introduktion	2
2	Problemformulering	3
3	Problemstillingen i 2D	4
3.1	Linearisering ved Substitution	6
3.2	Linearisering ved Subtraktion	6
3.2.1	Skæringlinjer i Ligningssystemer Uden Løsning	9
3.3	Mindste Kvadraters Metode i 2D	10
4	Den Egentlige Problemstilling	14
4.1	Variabelliste og Data	14
4.2	Skæringsplaner	15
4.3	Linearisering af Ligninger	18
4.3.1	Koefficienterne for matrix A	20
4.3.2	Koefficienterne for matrix b	22
4.3.3	Konkret eksempel af linearisering	23
4.4	Iterativ Taylor-Udvikling	25
4.5	Kombinatorisk løsning	27
4.6	Mindste Kvadraters Metode i 3D	28
4.6.1	Reducering af summen af de kvadrerede residualer	29
4.6.2	Anvendt i praksis	31
5	Diskussion	32
6	Konklusion	33
7	Bilag	34

1 Introduktion

Dette projekt omhandler GPS (Global Positioning System), et system hvorved en række satellitter i kredsløb om Jorden anvendes til positionering og navigation. Systemet fungerer ved at en GPS-modtager tager imod signaler fra GPS-satellitter. Ved hjælp af en særlig kode i signalet, kan modtageren afgøre den tid det har taget for et givet signal at rejse fra satellitten til modtageren, og derved afstanden mellem satellit og modtager.

GPS-satellitterne fungerer da som punkter med kendte koordinater, og GPS-modtageren som et punkt med ukendte koordinater, men kendte afstande til satellitterne.

Positionen af satellitter og modtager udregnes i et særligt koordinatsystem kaldet WGS84. Dette koordinatsystem har Jordens centrum som origo. x-aksen løber gennem punktet, hvor nulmeridianen og ækvator krydser, z-aksen løber langs Jordens omdrejningsakse, og y-aksen ligger vinkelret på de to andre akser som afbildet på Figur 17.



Figur 1: En grafisk repræsentation af Jorden og WGS84-koordinatsystemet. Den røde cirkel markerer ækvator, og den blå cirkel markerer nulmeridianen. x-aksen skærer hvor de to cirkler mødes, og z-aksen følger Jordens omdrejningsakse. Punktet M markerer en vilkårlig position for en GPS-modtager, og de gule pile viser retningen af nord og øst relativt til M. Dette projekt omhandler primært at finde punkt M's koordinater som funktion af de omkringliggende satellitters position og afstand til M.

Af afgørende betydning er urfejlen i GPS-modtageren. Da GPS-signalet bevæger sig med lysets hastighed, kan fejl på bare få millisekunder give afvigelser i afstandsbestemmelsen på hundredevis af kilometer. Fejlen i modtagerens ur er dermed en lige så vigtig ubekendt som modtagerens koordinater.

2 Problemformulering

Formålet med dette projekt er at finde koordinaterne til en GPS-modtager ud fra positionelle data og pseudoafstande for en række satellitter.

I projektet foretages der udregninger på seks satellitters data, herunder deres kendte positioner i WGS84-koordinatsystemet og deres "pseudoafstand" til en given GPS-modtager. Der er tale om pseudoafstande, fordi afstanden til hver satellit er beregnet ud fra den målte tid mellem signalets afsendelse til den bliver modtaget. Modtagerens ur er ikke perfekt, så den målte afstand vil i praksis ikke være den sande afstand mellem modtager og satellit. Tabel 1 viser satellitternes data.

Udover data for satellitterne, er der til denne opgave givet en foreløbig position (Tabel 2), dvs. en omtrentlig position for modtageren, samt den egentlige position for modtageren (Tabel 3), som bruges til sammenligning af resultaterne fra beregning af de andre data.

Data

Satellit nummer	X_s meter	Y_s meter	Z_s meter	P_s meter
4	4396623.907	-15219512.421	21395963.449	22745185
14	12508988.544	10190642.686	21073725.487	20490898
16	15690450.142	-6024729.548	20505506.585	20673780
18	25180700.379	-7388281.420	4051959.985	23193809
24	-8441248.473	-20422306.240	15063403.391	26237952
25	-2062043.991	17167206.349	20373015.170	22979470

Tabel 1: De seks satellitters data, herunder deres position i WGS84-koordinatsystemet, og deres pseudoafstand til GPS-modtageren.

X_0 meter	Y_0 meter	Z_0 meter
3504300	780800	5252100

Tabel 2: Den omtrentlige position af GPS-modtageren.

X meter	Y meter	Z meter
3504320.6	780753.5	5252128.8

Tabel 3: Den egentlige position for GPS-modtageren. Disse data skal ikke bruges i beregninger, men bruges til sammenligning, så det kan vurderes hvor akkurat en løsning er.

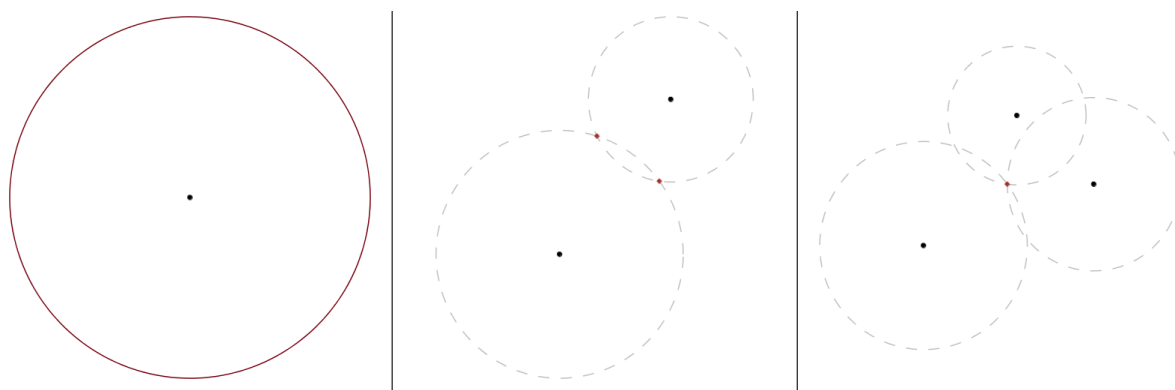
3 Problemstillingen i 2D

Inden der kigges på den fulde problemstilling, vil der i dette afsnit tages stilling til en simplere udgave af problemet. I denne version er der ingen urfejl på modtageren, og problemet betragtes kun i 2D-planen. Ved at se bort fra urfejlen er der ikke længere tale om en satellits pseudoafstand, men dens egentlige afstand til GPS-modtageren.

Den potentielle position af GPS-modtageren er dermed givet ved en cirkelligning, med satellittens position som centrum og den egentlige afstand som radius.

Figur 2 viser hvordan GPS-modtagerens position indsnævres som der indføres flere satellitter, eller hvordan løsningen til et system af cirkelligninger bliver mere entydig når der indføres flere ligninger. Ved et system af tre cirkelligninger er der altid én eller ingen løsninger, så længe de tre cirkler ikke ligger på en ret linje.

I planen skal der dermed bruges tre satellitter for at fastslå positionen af en GPS.

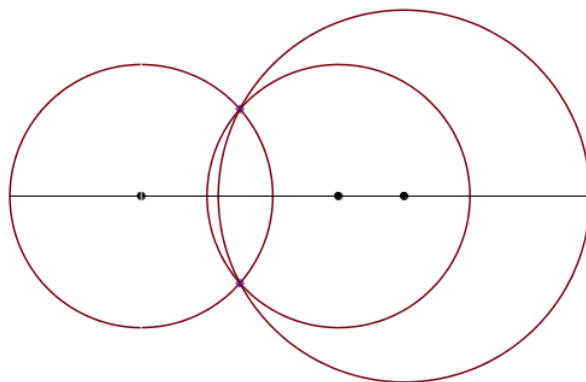


Figur 2: Systemer af cirkelligninger og deres løsninger (markeret med rødt) grafisk. For en enkelt ligning er der uendeligt mange løsninger, svarende til hele cirkelns periferi. For to ligninger er der 0, 1 eller 2 løsninger, svarende til krydsningspunktet for de to cirkler. For tre ligninger er der altid én eller ingen løsninger.

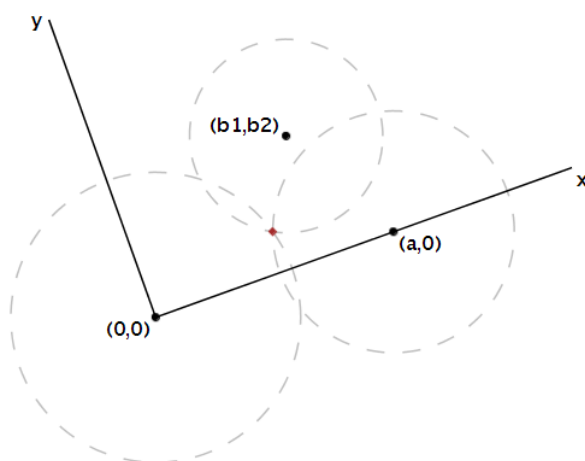
Der foretages her beregninger på vilkårlige cirkelligninger. Deres positioner er kun relevante i forhold til hinanden, så beregningerne kan foretages i et hvilket som helst koordinatsystem. I dette afsnit antages det, at koordinatsystemet altid placeres så cirkel 1 har centrum i origo $(0, 0)$, cirkel 2 har centrum i et vilkårligt punkt på x-aksen $(a, 0)$, og cirkel 3 har en vilkårlig position som ikke er på x-aksen (b_1, b_2) .

Det antages, at tredje cirkel ikke ligger på x-aksen, da systemet ikke har en entydig løsning hvis alle cirklers centrum ligger på samme linje. Dette er illustreret på Figur 3.

Et eksempel på hvordan et koordinatsystem kan lægges vises på Figur 4.



Figur 3: Et cirkelligningssystem hvor alle tre cirkler er placeret på samme linje. Der er to mulige løsninger, og den tredje cirkel er overflødig når det kommer til at indsnævre antallet af løsninger.



Figur 4: Et cirkelligningssystem og et koordinatsystem placeret således at én cirkel har centrum i origo og en anden cirkel har centrum på x-aksen.

Observationsligninger for GPS-satelliter uden urfejl opstilles som følgende ligningssystem:

$$\text{cirkel 1 : } (X - X_1)^2 + (Y - Y_1)^2 = P_1^2$$

$$\text{cirkel 2 : } (X - X_2)^2 + (Y - Y_2)^2 = P_2^2$$

$$\text{cirkel 3 : } (X - X_3)^2 + (Y - Y_3)^2 = P_3^2$$

Ved at indføre koordinatsystemet beskrevet ovenfor kan ligningssystemet simplificeres til:

$$\text{cirkel 1 : } X^2 + Y^2 = P_1^2$$

$$\text{cirkel 2 : } (X - a)^2 + Y^2 = P_2^2$$

$$\text{cirkel 3 : } (X - b_1)^2 + (Y - b_2)^2 = P_3^2$$

3.1 Linearisering ved Substitution

Ved at indføre variablen $s = X^2 + Y^2$, kan cirklerne ligninger lineariseres ved at substituere de kvadratiske ubekendte:

$$\text{Cirkel 1 : } -2XX_1 - 2YY_1 + X_1^2 + Y_1^2 + s = P_1^2$$

$$\text{Cirkel 2 : } -2XX_2 - 2YY_2 + X_2^2 + Y_2^2 + s = P_2^2$$

$$\text{Cirkel 3 : } -2XX_3 - 2YY_3 + X_3^2 + Y_3^2 + s = P_3^2$$

Ved at bruge det tilpassede koordinatsystem bliver systemet af ligninger til:

$$\text{Cirkel 1 : } s = P_1^2$$

$$\text{Cirkel 2 : } -2aX + a^2 + s = P_2^2$$

$$\text{Cirkel 3 : } -2b_1X - 2b_2Y + b_1^2 + b_2^2 + s = P_3^2$$

Dermed er der opstillet et system af 3 ligninger med 3 ubekendte.

Herfra opstilles følgende totalmatrix:

$$\underline{T} = \begin{bmatrix} 0 & 0 & 1 & P_1^2 \\ -2a & 0 & 1 & P_2^2 - a^2 \\ -2b_1 & -2b_2 & 1 & P_3^2 - b_1^2 - b_2^2 \end{bmatrix}$$

Hvis der laves rækkeoperationer på ovenstående totalmatrix fås en trappeform med løsningen for systemet af ligninger:

$$\underline{T} = \begin{bmatrix} 1 & 0 & 0 & \frac{P_1^2 - P_2^2 + a^2}{2ab_2} \\ 0 & 1 & 0 & \frac{P_1^2 a - P_1^2 b_1 + P_2^2 b_1 - P_3^2 a - a^2 b_1 + b_1^2 a + b_2^2 a}{P_1^2} \\ 0 & 0 & 1 & P_1^2 \end{bmatrix}$$

Løsningens koordinater kan her læses fra de første to rækker på den løste matrix ovenfor. Denne løsning stemmer overens med den fundet i forrige afsnit ved subtraktion af ligninger.

Det er værd at bemærke at $s = X^2 + Y^2 = P_1^2$, svarende til ligningen for cirkel 1. Denne løsning gælder altså kun hvis den ligger på cirkel 1's periferi, altså når der ikke er nogen urfejl.

3.2 Linearisering ved Subtraktion

Når der ikke er en urfejl på satellitterne, bliver skæringspunkterne for satellitternes cirkelligninger den entydige løsning for positionen af GPS-modtageren. Dermed er det af særlig interesse at undersøge cirklerne skæringspunkter.

Ved at trække ligningerne for to cirkler i, j fra hinanden opstilles følgende ligning:

$$(X - X_i)^2 + (Y - Y_i)^2 - ((X - X_j)^2 + (Y - Y_j)^2) = P_i^2 - P_j^2$$

Som kan omskrives til:

$$(X - X_i)^2 + (Y - Y_i)^2 - P_i^2 = (X - X_j)^2 + (Y - Y_j)^2 - P_j^2$$

Hvor X_i og Y_i er koordinater for centrum af cirkel i og P_i er radius til cirkel i.

Dette udtryk kan simplificeres ved at betragte længden s_i fra punktet $Q = (X, Y)$ til cirklerne centrum. Første led på hver side af ligningen kan dermed omskrives til kvadratet af denne længde.

$$s_i^2 - P_i^2 = s_j^2 - P_j^2 \quad (1)$$

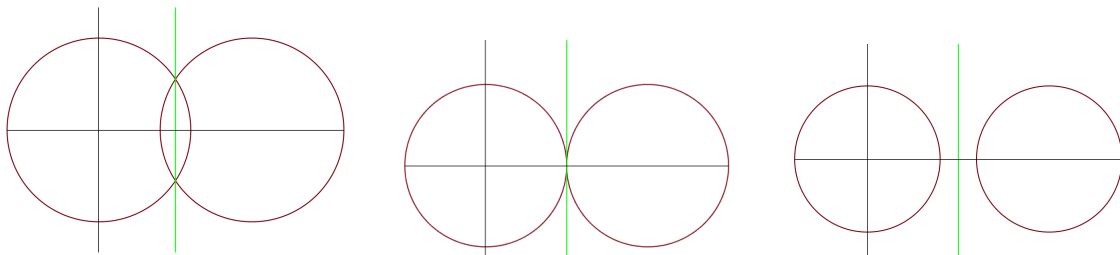
Ovenstående ligning beskriver en linje, der har den egenskab, at forskellen på den kvadrerede radius af cirklen og den kvadrerede afstand fra ethvert punkt på linjen til centrum af cirklen er den samme for begge cirkler.

Særligt gælder der at hvis $s_i^2 - P_i^2 = 0$ er der tale om et punkt på periferien af cirkel i, da punkt Q 's afstand er lig radius. I dette tilfælde gælder der pga. ligheden selvfølgelig også at $s_j^2 - P_j^2 = 0$, punktet

er også på periferien af cirkel j.

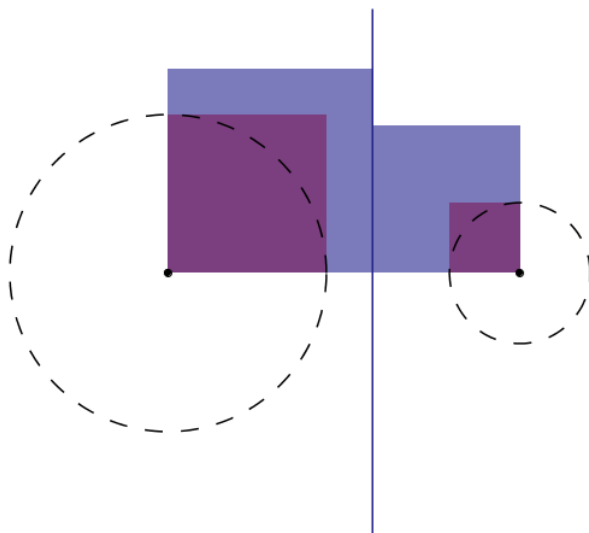
Ligningen der opstilles ved subtraktion af to cirkelligninger beskriver altså den linje, der løber gennem de to cirklers krydsningspunkter.

Figur 5 giver en grafisk repræsentation af denne skæringslinje for tre scenarier: Når cirklerne har to fælles løsninger, når de har en enkelt fælles løsning, og når de ikke har en fælles løsning.



Figur 5: Her kan man se de tre scenarier af cirklerne positioner og hvad linjen for et cirkelpar er.

Når cirklerne ikke krydser hinanden, beskriver den fundne skæringslinje naturligvis ikke nogen skæringspunkter. Den beskriver kun de punkter hvor forholdet beskrevet i Ligning (1) gælder. Dette forhold er grafisk repræsenteret i Figur 6.



Figur 6: En grafisk repræsentation af forholdet mellem cirklers radius og afstanden fra de cirklers centrum til et punkt på skæringlinjen. De lilla firkanters areal svarer til den kvadrerede radius P_i^2 , og de blå firkanters areal (inklusive den lilla firkant) repræsenterer den kvadrerede afstand mellem centrum af cirklen og samme punkt på skæringslinjen s_i^2 . Forskellen på de to arealer er den samme for begge cirkler, og det gælder uanset hvilket punkt på skæringslinjen der vælges.

For cirkel 1, 2 og 3 kan der ved parvis subtraktion opstilles følgende ligninger:

$$L_{1,2} : 2aX - a^2 = P_1^2 - P_2^2$$

$$L_{1,3} : 2b_1X + 2b_2Y - b_1^2 - b_2^2 = P_1^2 - P_3^2$$

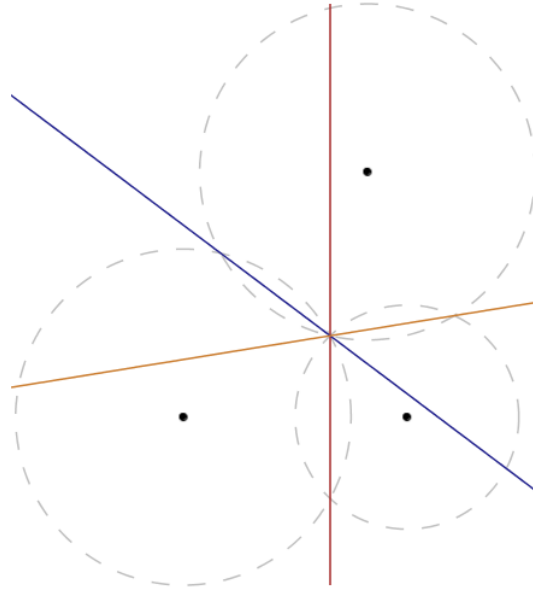
$$L_{2,3} : (-2a + 2b_1)X + 2b_2Y_2 + a^2 - b_1^2 - b_2^2 = P_2^2 - P_3^2$$

Disse ligninger er ikke just brugbare, men kan omskrives ved isolation af X eller Y :

$$\begin{aligned} L_{1,2}: \quad X &= \frac{a^2 + P_1^2 - P_2^2}{2a} \\ L_{1,3}: \quad Y &= -\frac{b_1}{b_2}X + \frac{b_1^2 + b_2^2 + P_1^2 - P_2^2}{2b_2} \\ L_{2,3}: \quad Y &= \frac{2a - 2b_1}{2b_2}X - \frac{a^2 - b_1^2 - b_2^2 - P_1^2 + P_2^2}{2b_2} \end{aligned}$$

Linjen $L_{1,2}$ er altså en konstant x -værdi, dvs. en linje, der løber langs y -aksen. $L_{1,3}$ og $L_{2,3}$, der løber mellem par med cirkel 3, er lineære ligninger, og beskriver dermed også linjer, som forventet. Figur 5 viser tre eksempler på hvordan linjen $L_{1,2}$ kan se ud.

Da hver linje enkeltvis beskriver de fælles løsninger for et cirkelpar, må løsningen for hele ligningssystemet findes på alle tre linjer, dvs. der må findes et krydsningspunkt for $L_{1,2}$, $L_{1,3}$ og $L_{2,3}$, som også er krydsningspunktet for cirkel 1, 2 og 3. Dette er illustreret ved Figur 7.



Figur 7: Et system af cirkelligninger med én løsning, samt skæringslinjerne for hvert cirkelpar. Skæringslinjerne krydser alle hinanden i netop ét punkt, som også er løsningen til systemet af ligninger.

Det viser sig at krydsningslinjerne altid krydser hinanden i ét punkt, selv hvis det bagvedliggende ligningssystem ikke har en løsning. Da der for $L_{1,2}$ og $L_{1,3}$ gælder:

$$s_1^2 - P_1^2 = s_2^2 - P_2^2 \quad \text{og} \quad s_1^2 - P_1^2 = s_3^2 - P_3^2$$

Må der også gælde at $s_2^2 - P_2^2 = s_3^2 - P_3^2$ i punktet hvor $L_{1,2}$ og $L_{1,3}$ mødes, og det må derfor også være et punkt som $L_{2,3}$ passerer igennem.

Dermed er det også vist at skæringspunktet for alle tre linjer kan findes ved blot at finde skæringspunktet mellem to af linjerne.

For linjerne $L_{1,2}$, $L_{1,3}$ og $L_{2,3}$ kan det fælles krydsningspunkt findes ved at løse totalmatricen for ligningssystemet. Som bemærket ovenfor er det kun nødvendigt at løse for to af ligningerne, da det er bevist at tredje linje krydser samme sted:

$$\underline{T} = \begin{bmatrix} 2a & 0 & P_1^2 - P_2^2 + a^2 \\ 2b_1 & 2b_2 & P_1^2 - P_3^2 + b_1^2 + b_2^2 \end{bmatrix}$$

På reduceret form er totalmatricen:

$$\underline{T} = \begin{bmatrix} 1 & 0 & \frac{P_1^2 - P_2^2 + a^2}{2a} \\ 0 & 1 & \frac{P_1^2 a - P_1^2 b_1 + P_2^2 b_1 - P_3^2 a - a^2 b_1 + b_1^2 a + b_2^2 a}{2ab_2} \end{bmatrix}$$

Som angiver hhv. x- og y-koordinaten for den fælles løsning af alle tre lineære ligninger. Hvis cirkelligningssystemet har en løsning, er dets løsning også givet ved disse koordinater.

3.2.1 Skæringlinjer i Ligningssystemer Uden Løsning

I den egentlige problemstilling arbejdes der med urfejl, hvilket vil sige at den målte radius af de cirkler der er arbejdet med i dette afsnit i princippet alle vil have en afvigelse der er konstant for alle cirkler. Det kan derfor ikke forventes, at der arbejdes med ligningssystemer der har en løsning. Det bør derfor undersøges hvad skæringlinjerne betyder for et ligningssystem, hvor der ingen løsninger er.

I første omgang betragtes et enkelt cirkelpar. Som beskrevet tidligere, gælder der følgende forhold for de to cirklers skæringlinje:

$$s_i^2 - P_i^2 = s_j^2 - P_j^2$$

Da s_i og s_j beskriver afstanden fra centrum af hver cirkel til en mulig fælles løsning og P_i og P_j beskriver radius af hver cirkel, forventes det at forskellen på de to størrelser er lig den samme urfejl w .

$$s_i - P_i = w$$

$$s_j - P_j = w$$

Der opstår dermed følgende ligning, der skal gælde for alle fælles løsninger:

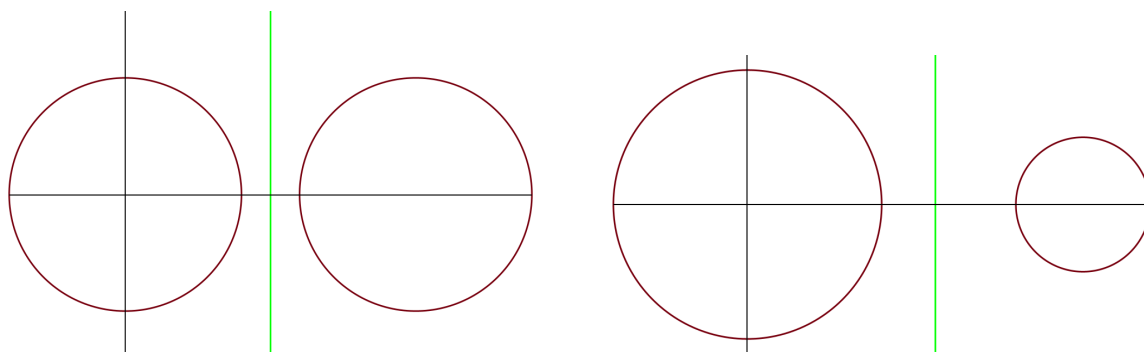
$$s_i - P_i = s_j - P_j \quad (2)$$

Ligning 2 stemmer kun overens med $L_{i,j}$ når $P_i = P_j$. Da udgår radierne, og begge ligheder kan dermed opfyldes. Når radierne er forskellige, beskriver Ligning 2 ikke længere samme linje som $L_{i,j}$, men en mere kompliceret kurve.

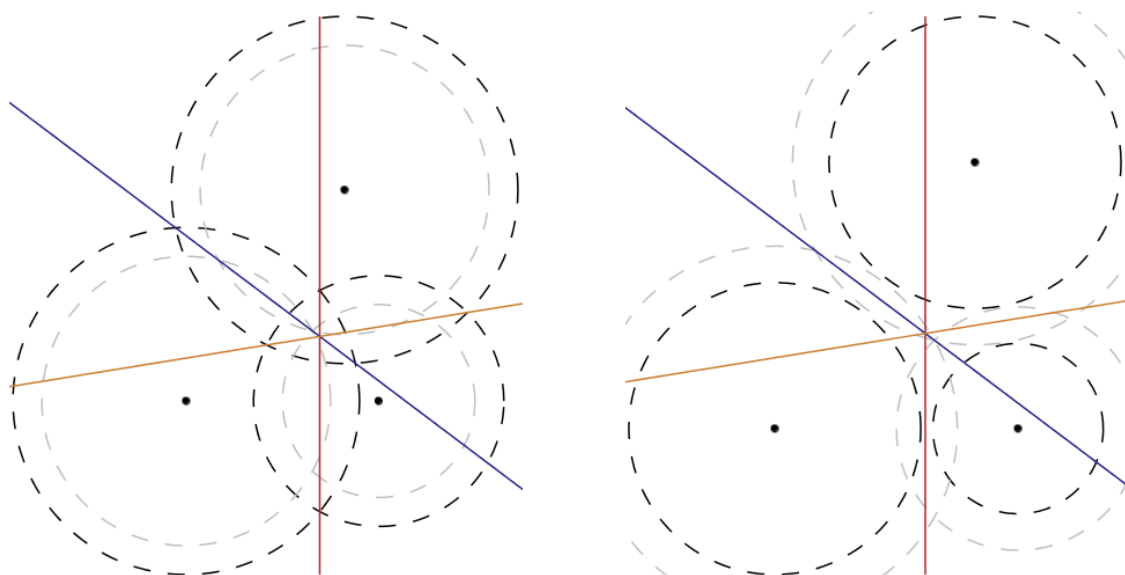
Det kan dermed konkluderes, at når der forskel på radierne, peger skæringlinjerne ikke på den sande løsning, da de indikerer de positioner hvor forskellen på kvadratet af afstandene er den samme. Afvigelsen af skæringlinjernes løsning afviger kvadratisk med unøjagtigheden af cirkelligningssystemet.

Jo mindre fejlen er, des mindre betyder denne kvadratiske afvigelse, så for systemer med en relativt lille fejl i forhold til de målte afstande, fungerer denne metode som en god approksimation for den sande løsning.

Figur 8 viser hvordan skæringlinjen opfører sig for cirkelpar med hhv. samme radius og forskellig radius. Figur 9 viser hvordan fællespunktet for skæringlinjerne ikke rammer samme punkt som findes ved at tage højde for urfejlen i ligningssystemet. Senere i dette afsnit vil det undersøges hvorvidt en mindste kvadraters metode kan kompensere for fejlen ved denne linearisering.



Figur 8: To cirkelpar uden fælles løsning og deres skæringlinje. I det ene tilfælde er begge radier ens, og skæringlinjen er lige langt fra hver cirkels periferi. I det andet tilfælde er der forskel på radierne, og skæringlinjen ligger nu tættere på den større cirkels periferi.



Figur 9: To cirkelligningssystemer hvor radius i hver cirkel er forskudt med en konstant urfejl. De grå cirkler beskriver den sande afstand, hvor der er en løsning til systemet, mens de sorte cirkler beskriver den målte pseudoafstand. Det er ud fra denne måling at skæringslinjerne er afsat, og det ses i begge tilfælde at skæringslinjernes krydsningspunkt rammer ved siden af den sande løsning.

3.3 Mindste Kvadraters Metode i 2D

Mindste kvadraters metode er en metode til at approksimere en løsning til systemer, hvor der er flere ligninger end ubekendte og hvor systemet - f.eks. grundet fejl el. lign. - er inkonsistent og derved ikke har en løsning. Intuitionen bag mindste kvadraters metode er at finde den approksimation, der minimerer den samlede fejlmængde, altså størrelsen af afvigelserne fra det faktiske punkt, mest muligt. I denne første del af opgaven er situationen med satellitterne blevet gennemgået i en simplere situation, nemlig i planen og uden urfejl. Dermed arbejdes der kun med 2 variable X og Y . Givet der er en ligning mere end der er variable, så er der herved tale om en situation, hvor dimensionen max er 3, så er det muligt grafisk at repræsentere metoden og fortolke den geometrisk. Når der senere arbejdes på den egentlige problemstilling vil metoden gennemgås i dybden via. matrixregning og calculus, så den herved vises at være gældende i n -dimensioner.

Det lineære ligningssystem bestående af linjerne $L_{1,2}$, $L_{1,3}$, $L_{2,3}$ undersøges nu igen, med henblik på at reducere fejlen på systemets løsning med mindste kvadraters metode:

$$\begin{aligned} L_{1,2} : \quad & 2aX - a^2 = P_1^2 - P_2^2 \\ L_{1,3} : \quad & 2b_1X + 2b_2Y - b_1^2 - b_2^2 = P_1^2 - P_3^2 \\ L_{2,3} : \quad & (-2a + 2b_1)X + 2b_2Y + a^2 - b_1^2 - b_2^2 = P_2^2 - P_3^2 \end{aligned}$$

Vi betragter det lineære ligningssystem på formen:

$$\underline{\underline{A}} \underline{x} = \underline{b}$$

hvor $\underline{\underline{A}}$ er en $n \times m$ - matrix og $\delta(A) = r$.

Generelt gælder det, at der findes en eksakt løsning til systemet, hvis $\underline{\underline{A}}$ er en kvadratisk og non-singulær matrix, så $n = m = r$ og $\det(\underline{\underline{A}}) \neq 0$. Matrixen har da fuld rang og kaldes også regulær. Herved er $\underline{\underline{A}}$, hvad man kalder invertibel, hvilket vil sige den har en invers matrix $\underline{\underline{A}}^{-1}$, således at:

$$\underline{\underline{A}} \underline{\underline{A}}^{-1} = \underline{\underline{A}}^{-1} \underline{\underline{A}} = \underline{\underline{E}}$$

hvor $\underline{\underline{E}}$ er enhedsmatrixen. Her vil løsningen \underline{x} altså opfylde følgende:

$$\underline{b} - \underline{\underline{A}} \underline{x} = 0$$

Da vi her har med $n = 2$ og $m=3$ at gøre, altså tre ligninger med to ubekendte. Af denne grund vil $\underline{\underline{A}}$ ikke være invertibel og en eksakt løsning findes da ikke. Geometrisk set betyder det, at vektoren \underline{b} ikke befinder sig i $\underline{\underline{A}}$'s billedemængde.

Det er nu her mindste kvadraters metode kan tages i brug. Da den eksakte løsning \underline{x} ikke eksisterer, så må det næstbedste være at finde den løsning x_0 , der kommer tættest på at opfylde alle ligningerne i ligningsystemet. Dvs. at metoden går ud på at finde det sæt variable (X, Y) , der minimerer den samlede totale fejlstørrelse. Denne findes ved at tage summen af de kvadrerede residualer. Et residual på en løsning er defineret som:

$$\underline{\underline{R}} = \underline{b} - \underline{\underline{A}} \underline{x}$$

Hvor der for det udvalgte ligningssystem gælder:

$$\underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a^2 + P_1^2 - P_2^2 \\ b_1^2 + b_2^2 + P_1^2 - P_3^2 \\ -a^2 + b_1^2 + b_2^2 + P_2^2 - P_3^2 \end{bmatrix}$$

$$\underline{\underline{A}} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{bmatrix} = \begin{bmatrix} 2a & 0 \\ 2b_1 & 2b_2 \\ -2a + 2b_1 & 2b_2 \end{bmatrix}$$

$$\underline{x} = \begin{bmatrix} X \\ Y \end{bmatrix}$$

Herefter sættes udtrykket for den samlede algoritme op. Formlen benytter kvadrede residualer bl.a. for at behandle positive og negative fejl ens og for at forsikre, at udtrykkene for residualerne altid er differentierbare, hvilket er brugbart senere:

$$\min_{\underline{x}} \sum_{i=1}^n R_i^2$$

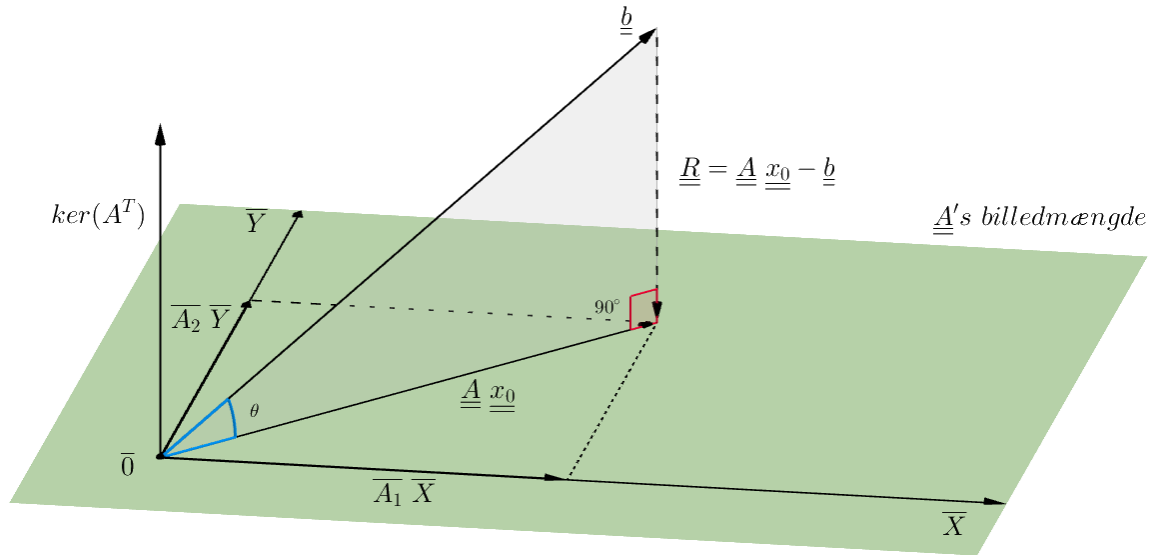
Da $\underline{b} \notin C(\underline{\underline{A}})$ og \underline{b} ikke kan skrives som et matrix-vektorprodukt af $\underline{\underline{A}} \underline{x}$, så må den vektor, der ligger tættest på \underline{b} vi kan komme frem til være projektionen af \underline{b} ned på det vektorrum som $\underline{\underline{A}}$'s søjler udspænder:

$$\underline{\underline{A}} \underline{x}_0 = proj_{C(\underline{\underline{A}})} \underline{b}$$

hvor \underline{x}_0 er mindste kvadraters løsning. Ved at trække \underline{b} fra dette udtryk findes den vektor, der går fra \underline{b} 's endepunkt til projektionen af \underline{b} 's endepunkt. Det er længden af denne der skal minimeres:

$$\underline{\underline{A}} \underline{x}_0 - \underline{b} = proj_{C(\underline{\underline{A}})} \underline{b} - \underline{b}$$

Denne længde svarer naturligvis til residualerne (se Figur 10).



Figur 10: Illustration af mindste kvadraters metode

Det vides at denne afstandsvektor/residualvektor må stå vinkelret på \underline{A} 's billedrum og herved være en del af matrixens ortogonalkompliment

$$\underline{A} \underline{x}_0 \in \underline{A}^\perp$$

Ortogonalkomplimentet udgør kernen for \underline{A}^T :

$$\underline{A}^\perp = \ker(\underline{A}^T) \Rightarrow \underline{A} \underline{x}_0 - \underline{b} \in \ker(\underline{A}^T)$$

Ved at gange denne transponerede matrix \underline{A}^T på udtrykket for residualen, så forsvinder hele fejlen/residualen:

$$\underline{A}^T \underline{R} = \underline{A}^T (\underline{b} - \underline{A} \underline{x}_0) = 0$$

Herved kan vi altså opskrive følgende udtryk:

$$\begin{aligned} \underline{A}^T \underline{A} \underline{x}_0 - \underline{A}^T \underline{b} &= 0 \Rightarrow \\ \underline{A}^T \underline{A} \underline{x}_0 &= \underline{A}^T \underline{b} \end{aligned}$$

Dette udtryk vil altid have en løsning, da projektionen af \underline{b} naturligvis altid må ligge i det vektorrum som \underline{A} 's søjlevektorer udspænder. Altså ligger projektionen i \underline{A} 's billedrum. Løsningen \underline{x}_0 kan da opskrives:

$$\underline{x}_0 = (\underline{A}^T \underline{A})^{-1} \underline{A}^T \underline{b}$$

Det er af interesse hvordan den mindste kvadraters metode kan bruges i samspil med de lineariseringer, der opstår ved subtraktion af ligninger. Da der altid er givet en eksakt løsning ved linjerne mellem 3 cirkler, er det derfor nødvendigt at undersøge et eksempel på et ligningssystem med flere cirkler, en urfejl og en kendt løsning.

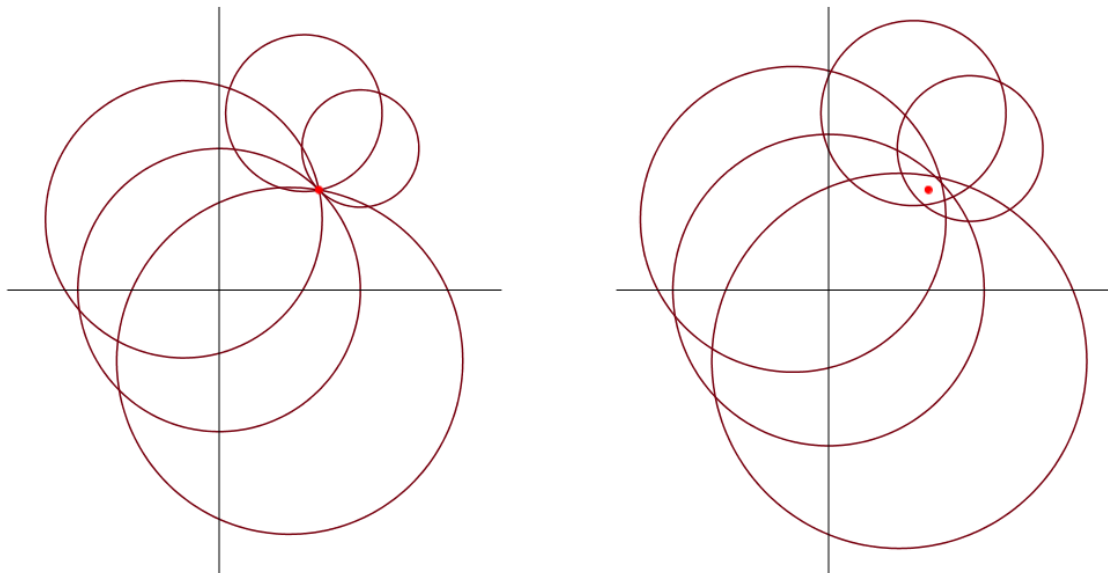
Figur 11 viser et eksempel et system af 5 cirkelligninger. Første plot viser de sande afstande, hvortil der findes én løsning til systemet, og andet plot viser samme system, men hvor raderne til alle cirkler er forskudt med samme konstant. Systemet har altså en urfejl. I begge tilfælde vises positionen for den sande position af GPS modtageren.

Ved at trække ligningerne fra hinanden for hvert cirkelpar i systemet, kan der dannes et system af i alt 10 lineære ligninger. Dette ligningssystem kan fortolkes på formen:

$$\underline{A} \underline{x} = \underline{b}$$

Løsningen til den mindste kvadraters metode er tidligere i afsnittet vist at være:

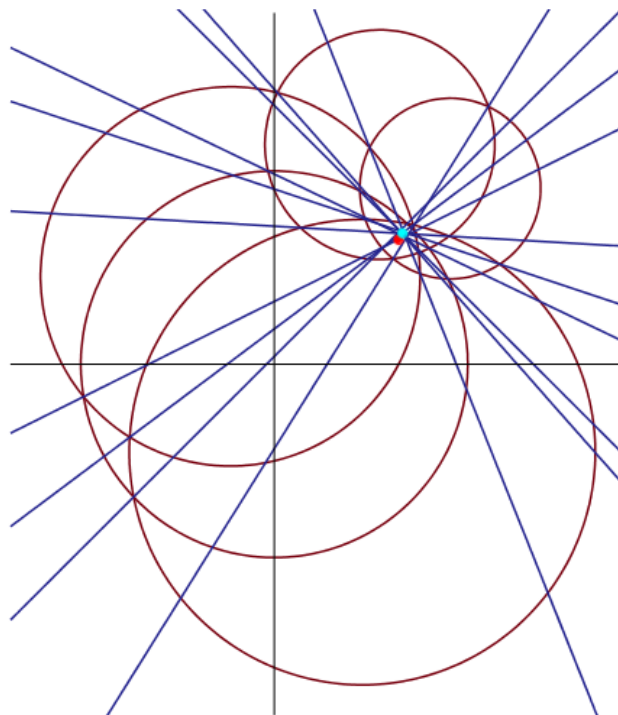
$$\underline{x} = \begin{bmatrix} X \\ Y \end{bmatrix} = (\underline{A}^T \underline{A})^{-1} \underline{A}^T \underline{b}$$



Figur 11: En illustration af et system af fem cirkelligninger, samt deres løsning. Venstre plot simulerer den sande afstand fra satellit til modtager, og højre plot viser samme system med en konstant urfejl for alle ligninger.

Figur 12 viser ligningssystemet med skæringslinjerne, samt den løsning der opnås ved at benytte mindste kvadraters metoden på de lineære ligninger.

Om denne metode er akkurat nok til bestemmelse af en GPS-modtager vides ikke på dette tidspunkt, og en sådan vurdering falder udenfor undersøgelsen af dette arbitrære ligningssystem. I stedet vil der i de kommende afsnit tages stilling til den originale problemstilling, og her vil flere løsningsmetoder tages i brug og sammenlignes.



Figur 12: Illustrationen af ligningssystemet fra Figur 11 med urfejl, nu tilføjet skæringslinjerne for hvert cirkelpar i systemet. Løsningen til mindste kvadraters metoden er angivet som den lyseblå prik.

4 Den Egentlige Problemstilling

Nu hvor den forsimplede 2D-problemstilling er blevet undersøgt, tages der igen stilling til den originale problemstilling.

Der arbejdes nu i det 3-dimensionale rum. Satellitters positioner er givet ved en (X, Y, Z) -koordinat, og deres data kan nu beskrives ved ligningen for en kugleflade, med satellittens koordinat som centrum og dens målte pseudoafstand som radius.

Der er flere tilgangsmetoder til at finde en optimal løsning til et sådant ligningssystem. I de kommende afsnit vil der undersøges hvordan systemet bedst kan fortolkes og simplificeres, og der vil blive taget flere løsningsmetoder i brug for at finde den bedst mulige metode til at udregne positionen af GPS-modtageren.

Den egentlige position er givet af projektopgaven, som også giver en foreløbig position af modtageren, som kan bruges til f.eks. Taylor-udviklinger.

De data der er beskrevet i projektbeskrivelsen er angivet på Tabel 1, 2 og 3. Tabellerne findes under problemformuleringen og bliver reitereret under næste afsnit.

4.1 Variabelliste og Data

Symbol	Forklaring	Værdi
X_s	Koordinater for en satellit.	-
Y_s		
Z_s		
P_s	Pseudoafstanden fra en satellit til GPS modtageren.	-
M	GPS-modtagerens kendte position.	(3504320.6, 780753.5, 5252128.8)
M_0	GPS-modtagerens foreløbige position.	(3504300, 780800, 5252100)
X_0	Koordinater for GPS-modtagerens foreløbige position.	3504300
Y_0		780800
Z_0		5252100

Tabel 4: Listen over variable der benyttes i beregninger, samt deres forkortelser.

Satellit nummer	X_s meter	Y_s meter	Z_s meter	P_s meter
4	4396623.907	-15219512.421	21395963.449	22745185
14	12508988.544	10190642.686	21073725.487	20490898
16	15690450.142	-6024729.548	20505506.585	20673780
18	25180700.379	-7388281.420	4051959.985	23193809
24	-8441248.473	-20422306.240	15063403.391	26237952
25	-2062043.991	17167206.349	20373015.170	22979470

Tabel 1: De seks satellitters data, herunder deres position i WGS84-koordinatsystemet, og deres pseudoafstand til GPS-modtageren.

X_0 meter	Y_0 meter	Z_0 meter
3504300	780800	5252100

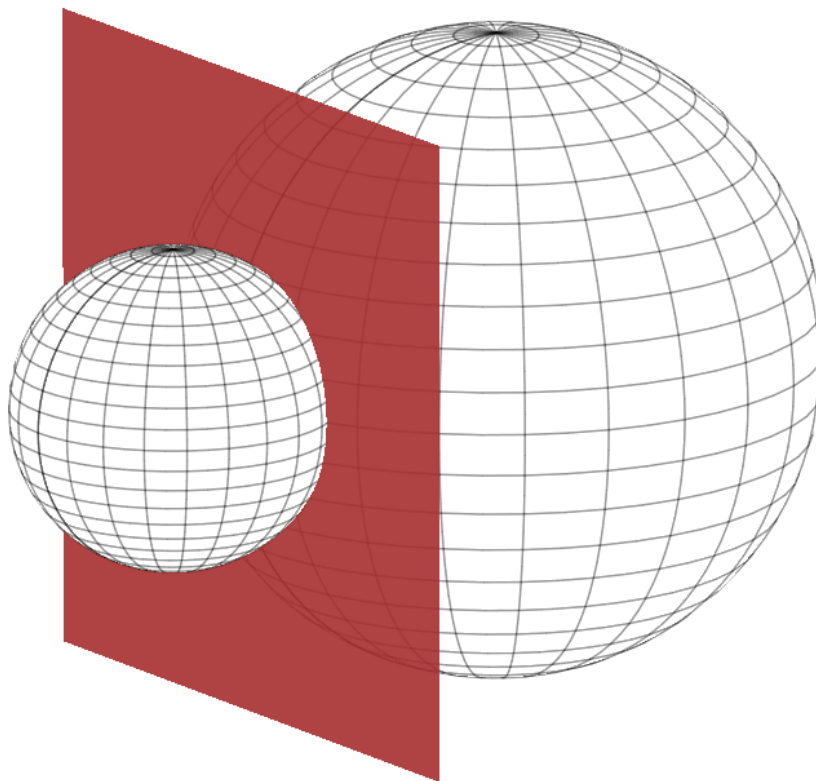
Tabel 2: Den omtrentlige position af GPS-modtageren. Denne er givet af opgavebeskrivelsen, men en tilsvarende værdi kunne findes ved at finde et fælles skæringspunkt for 4 af de kuglefladeligninger der kan opstilles for satellitterne i Tabel 1.

X	Y	Z
meter	meter	meter
3504320.6	780753.5	5252128.8

Tabel 3: Den egentlige position for GPS-modtageren. Disse data skal ikke bruges i beregninger, men bruges til sammenligning, så det kan vurderes hvor akkurat en løsning er.

4.2 Skæringsplaner

I 2D-problemstillingen var det muligt at finde skæringspunkterne mellem to cirkler ved at trække cirkelligningerne fra hinanden. Derved opstod ligningen for den linje, der krydsede cirklernes skæringspunkt. Samme metode kan bruges i det 3-dimensionale rum. Ved at trække to kugligninger fra hinanden fås ligningen for den plan, der indeholder alle skæringspunkter (se Figur 13).

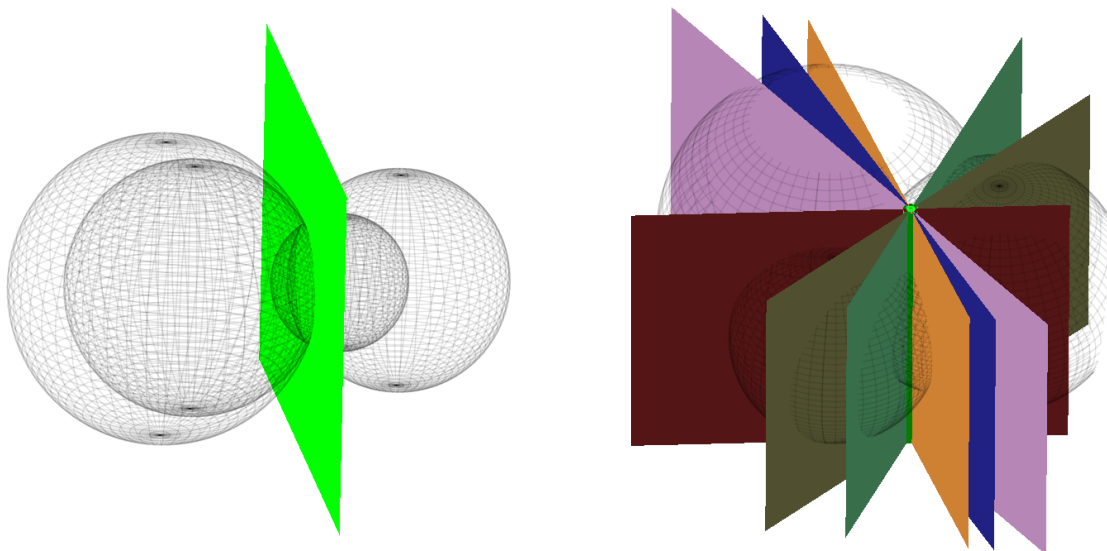


Figur 13: To vilkårlige kugleflader og deres skæringsflade.

Hvis der var en løsning til et system af 3 cirkelligninger i 2D, var det muligt at finde løsningen som det fælles krydsningspunkt for de tre linjer, der fremkom ved subtraktion af hvert ligningspar.

I det 3-dimensionale rum kan samme princip bruges for et system af 4 kuglefladeligninger. Det fælles krydsningspunkt for de planer, der fremkommer ved subtraktion af hvert ligningspar, er løsningen til ligningssystemet.

En entydig løsning fremkommer kun hvis der i ligningssystemet ikke er mere end 2 kuglecentre på en linje og ikke mere end 3 kuglecentre på et plan. Figur 14 viser hvordan løsningen ser ud når kuglecentrene er arrangeret på hhv. en fælles linje og plan.



Figur 14: Et system af kuglefladeligninger arrangeret først på samme linje, og så arrangeret på samme plan. I første tilfælde er der uendeligt mange krydsningspunkter arrangeret i en fælles skæringscirkel, og i andet tilfælde er der to krydsningspunkter der ligger på en ret linje.

I dette projekts problemstilling er der imidlertid ingen løsning til systemet af kuglefladeligninger der kan opstilles fra satellitternes data.

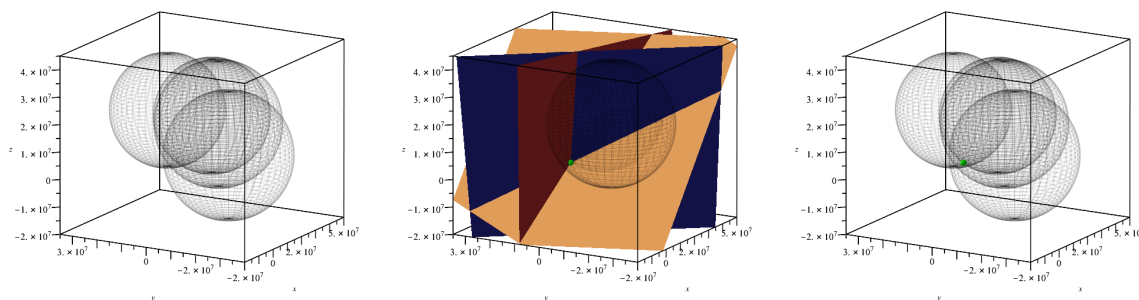
Udover at beskrive evt. fællespunkter, har planen mellem to kugler samme egenskab som skæringslinjen mellem to cirkler. Forskellen på kvadratet af afstanden s fra kuglens centrum til et givet punkt Q på planen og kvadratet af en kugles radius er den samme for begge kugler.

$$s_1^2 - r_1^2 = s_2^2 - r_2^2$$

Denne egenskab hjælper dog ikke dette projekts problemstilling. Urfejlen i GPS-modtageren er en konstant afvigelse mellem den sande afstand og den målte pseudoafstand. Den vokser ikke med forskellen til den målte pseudoafstand og afstanden til den egentlige position.

Det kan derfor ikke forventes, at GPS-modtagerens sande position kan bestemmes ved krydsningspunktet for planerne til hvert kuglepar.

Men hvis kuglefladerne er tæt på hinanden, kan denne metode hurtigt bestemme en værdi der approksimerer den sande værdi.



Figur 15: De overlappende kuglefladeligninger opstillet fra data for satellit 4, 14, 16 og 18. Venstre plot viser kun kuglefladerne. Midterste plot viser planerne der opstilles ud fra subtraktion af satellit 4's ligning fra de øvrige tre ligninger. Midterste og højre plot viser punktet, hvor alle tre planer krydser.

Figur 15 viser grafisk hvordan den approksimerede position af GPS-modtageren findes ud fra planernes krydsning. Punktet findes matematisk ved at løse ligningssystemet for de tre planer.

For satellitterne 4, 14, 16 og 18 kan der opstilles følgende lineære ligningssystem for alle parrene til satellit 4. Ligesom der i 2D kun behøves to linjer for at finde skæringslinjernes løsning, skal der i 3D kun

benyttes tre planer for at finde skæringsplanernes løsning.

$$l_{4,14} : 1.622472927 \cdot 10^7 X + 5.082031022 \cdot 10^7 Y - 644475.92 Z = 9.31413492 \cdot 10^{13}$$

$$l_{4,16} : 2.258765247 \cdot 10^7 X + 1.838956574 \cdot 10^7 Y - 1.78091374 \cdot 10^6 Z = 8.41505636 \cdot 10^{13}$$

$$l_{4,18} : 4.156815295 \cdot 10^7 X + 1.566246200 \cdot 10^7 Y - 3.468800693 \cdot 10^7 Z = -2.428767094 \cdot 10^{13}$$

Ved at løse ligningssystemet ovenfor fås koordinaterne (3503986.1 , 780685.2 , 5251653.0).

Ved sammenligning med den kendte position M , er afstanden mellem disse punkter 585.6 m.

Som forventet er resultatet af denne metode alt for langt fra M til at metoden kan bruges som løsning til problemstillingen. Resultatet af denne metode kan dog evt. bruges som et udviklingspunkt for de Taylor-udviklinger der foretages i de kommende afsnit. Hvis der i projektbeskrivelsen ikke allerede var angivet en foreløbig position, kunne denne metode bruges til hurtigt at fastlægge et omtrentligt udviklingspunkt.

4.3 Linearisering af Ligninger

Den målte afstand mellem GPS-modtageren og en given satellit, også kaldet pseudoafstanden, kan ved hjælp af pythagoras opstilles således:

$$P_s = \rho_s + c \cdot dT = \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + c \cdot dT \quad (3)$$

Hvor ρ_s er den geometriske afstand mellem modtager og satellit, (X, Y, Z) er modtagerens position (X_s, Y_s, Z_s) er satellittens position og cdT er urfejlen målt i sekunder ganget med lysets hastighed. Herved fås urfejlen i meter. Fremover bliver denne betegnet som w .

På baggrund af dette kan følgende ligningssystem opstilles. Idet der er 4 ubekendte (X, Y, Z, w) så skal der 4 satellitter og herved 4 ligninger til for at en løsning skal kunne findes:

$$\begin{aligned} \sqrt{(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2} + w &= P_1 \\ \sqrt{(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2} + w &= P_2 \\ \sqrt{(X - X_3)^2 + (Y - Y_3)^2 + (Z - Z_3)^2} + w &= P_3 \\ \sqrt{(X - X_4)^2 + (Y - Y_4)^2 + (Z - Z_4)^2} + w &= P_4 \end{aligned} \quad (4)$$

For at løse ligningssystemet (4) så foretages der en Taylor-udvikling af første orden. Herved fås der et lineært ligningssystem, hvor løsningen dertil naturligvis er approksimativ. Begyndelsesværdierne (Udviklingspunktet) sættes lig med $X = X_0$, $Y = Y_0$, $Z = Z_0$ og $w = 0$, for at bestemme det generelle taylor approksimation. Senere vil approksimationerne for de individuelle ligninger givet ved hver satellit blive beregnet via Maples indbyggede `mtaylor`-kommando.

Først opskrives ligningernes generelle udtryk:

$$\begin{aligned} \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + w &= P_s \Rightarrow \\ 0 &= \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + w - P_s \end{aligned} \quad (5)$$

Da udtrykket er afhængig af 4 variable, X , Y , Z og w , kan formen af det approksimerede 1. grads taylor-polynomie opskrives som:

$$\begin{aligned} Pol(x, y, z, w) &= f(x_0, y_0, z_0, w_0) + f'_x(x_0, y_0, z_0, w_0) \cdot (x - x_0) + f'_y(x_0, y_0, z_0, w_0) \cdot (y - y_0) \\ &\quad + f'_z(x_0, y_0, z_0, w_0) \cdot (z - z_0) + f'_w(x_0, y_0, z_0, w_0) \cdot (w - w_0) \end{aligned}$$

Først omskrives den generelle ligning til funktionsform idet udtrykket nu ikke sættes lig 0.

$$f(X, Y, Z, w) = \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + w - P_s \quad (6)$$

Nu bliver den første afledte af funktionen mht. X , Y , Z og w .

$$\begin{aligned} f'_X(X, Y, Z, w) &= \frac{2X - 2X_s}{2\sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}} \\ f'_Y(X, Y, Z, w) &= \frac{2Y - 2Y_s}{2\sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}} \\ f'_Z(X, Y, Z, w) &= \frac{2Z - 2Z_s}{2\sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}} \end{aligned}$$

$$f'_w(X, Y, Z, w) = 1$$

Nu hvor funktionen og alle dens partielle afledte er bestemt, kan Taylor approksimationen opstilles:

$$\begin{aligned} Pol(X, Y, Z, w) &= \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + 0 - P_s + \frac{(2X_0 - 2X_s)(X - X_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + \\ &\quad \frac{(2Y_0 - 2Y_s)(Y - Y_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + \frac{(2Z_0 - 2Z_s)(Z - Z_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + (w - 0) \\ &\quad \Downarrow \end{aligned}$$

$$\begin{aligned} Pol(X, Y, Z, w) &= \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + w - P_s + \\ &\quad \frac{(2X_0 - 2X_s)(X - X_0) + (2Y_0 - 2Y_s)(Y - Y_0) + (2Z_0 - 2Z_s)(Z - Z_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \end{aligned} \quad (7)$$

Udtrykket ser umiddelbart uoverskueligt ud. Ved at opskrive koefficienterne for X, Y, Z, w og konstantleddet hver for sig så fås:

$$X_{koef} = \frac{2X_0 - 2X_s}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} = \frac{X_0 - X_s}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$Y_{koef} = \frac{2Y_0 - 2Y_s}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} = \frac{Y_0 - Y_s}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$Z_{koef} = \frac{2Z_0 - 2Z_s}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} = \frac{Z_0 - Z_s}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$w_{koef} = 1$$

Konstantleddet opskrives:

$$K = \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} - P_s - \frac{((X_0 - X_s)X_0 + (Y_0 - Y_s)Y_0 + (Z_0 - Z_s)Z_0)}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Herved får man altså følgende ligning:

$$\begin{aligned} 0 &= X_{koef} \cdot X + Y_{koef} \cdot Y + Z_{koef} \cdot Z + w + K \Rightarrow \\ -K &= X_{koef} \cdot X + Y_{koef} \cdot Y + Z_{koef} \cdot Z + w \end{aligned}$$

Her ses det at efter Taylorapproksimation af (2), så er der tale om et lineært ligningssystem på formen:

$$\underline{\underline{A}} \underline{\underline{x}} = \underline{\underline{b}}$$

Hvor koefficienterne for 4 vilkårlige satellitter kan opskrives i koefficientmatrixen $\underline{\underline{A}}$, og deres tilsvarende konstantled kan opskrives som højresiden $\underline{\underline{b}}$ Herved fås:

$$\underline{\underline{A}} = \begin{bmatrix} X_{1,koef} & Y_{1,koef} & Z_{1,koef} & 1 \\ X_{2,koef} & Y_{2,koef} & Z_{2,koef} & 1 \\ X_{3,koef} & Y_{3,koef} & Z_{3,koef} & 1 \\ X_{4,koef} & Y_{4,koef} & Z_{4,koef} & 1 \end{bmatrix}$$

$$\underline{\underline{b}} = \begin{bmatrix} -K_1 \\ -K_2 \\ -K_3 \\ -K_4 \end{bmatrix}$$

Hvor løsningen til systemet er givet ved en vektor bestående af positionen af GPS modtageren og urfejlen w :

$$\underline{\underline{x}} = \begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix}$$

4.3.1 Koefficienterne for matrix **A**

Ved at observere udregningerne ovenfor så ses det indledningvis, at koefficienterne i $\underline{\underline{A}}$ svarer til koefficienterne for de partielt afledede af hver observation, med hensyn til hver variabel. Denne koefficientmatrix indeholder dog yderligere iformationer:

Fra ligningerne for X, Y og Z koefficienterne fremkommer det, at alle tre koordinater er divideret med:

$$\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}$$

Vi kan samle udtrykket for alle koefficienter i en vektor, ved at opstille:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 - X_s \\ Y_0 - Y_s \\ Z_0 - Z_s \end{bmatrix} \cdot \frac{1}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Hvis vi anser vores originale problemstilling, kan vi nemt opstille vektoren (\underline{v}) , som går fra en given satellit $(\underline{\underline{v}}_s)$ til GPS-modtagerens foreløbige position $(\underline{\underline{v}}_0)$:

$$\underline{\underline{v}}_s = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} \wedge \underline{\underline{v}}_0 = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

$$\underline{\underline{v}} = \underline{\underline{v}}_0 - \underline{\underline{v}}_s = \begin{bmatrix} X_0 - X_s \\ Y_0 - Y_s \\ Z_0 - Z_s \end{bmatrix}$$

Og længden af denne er:

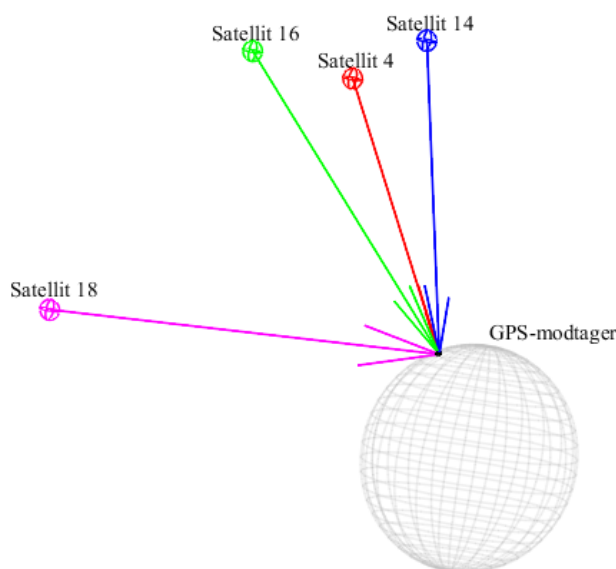
$$|\underline{v}| = \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}$$

Her ser vi, at udtrykket for koefficienterne af X, Y og Z kan omskrives til:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{\underline{v}}{|\underline{v}|}$$

Denne vektor \underline{v} er da blevet normeret og er nu en enhedsvektor. Koefficienterne af X, Y og Z svarer til koordinaterne for enhedsvektoren, der peger fra en given satellit mod GPS-modtagerens foreløbige position. Fra koefficientmatrixen, \underline{A} , ser vi da, at værdierne i de første tre kolonner i hver række svarer til koordinaterne for enhedsvektoren, som går fra den givne satellit mod GPS-modtagerens foreløbige position. Da GPS-modtagerens foreløbige position er ens for alle lineariseringerne, må de tre første kolonner i hver række pege mod det samme punkt blot fra forskellige positioner. Af denne grund, kan matricen betegnes som en design matrix.

Nedfor ses et eksempel mere 4-satellitter og de respektive retningsvektorer fundet i den tilhørende designmatrix \underline{A} . Her er de skaleret op så det kan ses at de røre hinanden i et punkt og herved er de naturligvis ikke enhedsvektorer længere.



Figur 16: situationen illustreret for de fire første satellitter

Udover dette, er det tydeligt, at den fjerde søjle i matricen består udelukkende af 1-taller. Den fjerde søjle betegner koefficienten for urfejlen w . Denne koefficient fremkommer fra lineariseringen og fortæller, at koefficienten for urfejlen er konstant og lig med 1 for enhver linearisering. Urfejlen er urfejlen for GPS-modtageren, og da GPS modtageren sandsynligvis opfanger satellitternes positioner i samme øjeblik, giver det god mening at urfejlen er den samme for alle ligninger i ligningssystemet.

Gennem opstillingen af det generelle udtryk af taylor approksimationen er begyndelsesværdien af urfejlen w sat lig med 0. Hvis vi indsætter begyndelsesværdien af urfejlen lig med et vilkårligt tal w_0 , for at undersøge om begyndelsesværdien af urfejlen har indflydelse på koefficienten for urfejlen, bliver taylor approksimation lig med:

$$Pol(X, Y, Z, w) = \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + w_0 - P_s + \frac{(2X_0 - 2X_s)(X - X_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + \frac{(2Y_0 - 2Y_s)(Y - Y_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + \frac{(2Z_0 - 2Z_s)(Z - Z_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + (w - w_0)$$

Her ser vi, at w_0 udligner sig selv og herefter er udtrykket det samme, som da begyndelsesværdien for urefejlen blev sat lig med 0. Da w_0 var et vilkårligt tal, må det betyde, at begyndelsesværdien for urefejlen ikke har indflydelse på ligningssystemet og hermed koefficienten for urefejlen i designmatricen.

4.3.2 Koefficienterne for matrix \mathbf{b}

Ovenfor er det bestemt at matricen \mathbf{b} består af konstantledende med negativt fortegn, hvor konstantledet er givet ved:

$$K = \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} - P_s - \frac{(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Og med negativt fortegn:

$$(-K) = -\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + P_s + \frac{(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Vi ser, at udtrykket er opdelt i 3 forskellige led. Det første led har vi set fra forrige delopgave, og svarer til den negative værdi af længden for vektoren, som går fra en given satellit til GPS-modtagerens foreløbige position (Se udtryk for v ovenfor).

Herefter ses værdien P_s , hvilket er pseudoafstanden mellem satellitten og GPS-modtagerens egentlige position. De første to led kan derfor anses som forskellen mellem afstanden til den foreløbige og (pseudo-) egentlige position.

For at undersøge betydningen af det sidste led, kigger vi på den tidligere defineret vektor \underline{v} , som betegner vektoren fra koordinatsystemets origo (Jordens centrum) til GPS-modtagerens foreløbige position.

Udover det, inddrager vi formlen for længden af en projektion mellem to vektorer:

$$|\vec{a}_{\vec{b}}| = \frac{|\vec{a} \cdot \vec{b}|}{|\vec{b}|}$$

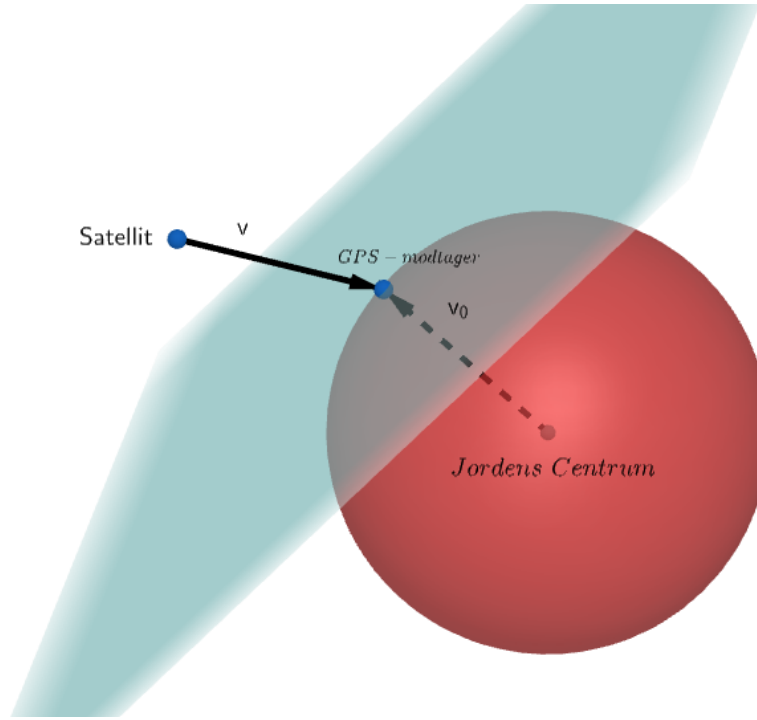
Hvis man bestemmer projektionen af vektor v_0 på vektor v bliver ligningen:

$$|\vec{v}_0_{\vec{v}}| = \frac{|\vec{v}_0 \cdot \vec{v}|}{|\vec{v}|} = \frac{|(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0|}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Har kan vi se, at det sidste led i udtrykket for $(-K)$ ligner meget længden af projektionen fra vektor \underline{v} på vektor \underline{v}_0 . Det eneste er, at den absolutte værdi er blevet fjernet fra udtrykket i $(-K)$. Hvis man ser på udtrykket for længden af projektionen, er fortegnet kun bestemt af fortegnet for skalarproduktet mellem v_0 og v , da længden af v altid er positiv. Vi inddrager nu udtrykket for skalarproduktet mellem to vektorer:

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\theta)$$

Hvor θ er vinklen mellem de to vektorer. Her ser vi yderligere, at fortegnet for skalarproduktet, og dermed længden af projektionen, kun er bestemt af vinklen mellem de to vektorer. Hvis vinklen mellem de to vektorer er spids, er fortegnet positiv og negativ hvis vinklen er stump. I praksis vil satellitterne altid være over horisonten for GPS-modtageren, for ellers ville jorden være i vejen og signalet mellem GPS-modtageren og satellitten ville være blokeret. Med andre ord så vil vinklen mellem vektoren som peger fra satellitten til GPS-modtagerens foreløbige punkt og vektoren fra Jordens centrum til det foreløbige punkt altid have en stump vinkel. Det ses i illustrationen under:



Figur 17: Illustrationen af vinkel mellem vektor fra satellit og jordens centrum til GPS-modtagerens foreløbige position. Planen repræsenterer horisonten. Hvis satellitten bevæger sig under horisonten bliver vinklen spids, og hermed bliver længden af projektionen positiv. Dette vil ikke forekomme i praksis.

I praksis vil længden af projektionen derfor have negativ fortegn inden den absolutte værdi bliver bestemt. Med dette i betragtning, kan værdien af $(-K)$, og hermed værdierne i matrice \underline{b} , anses som at være værdien af forskellen mellem pseudoafstanden og afstanden fra satellitten til den foreløbige position af GPS-modtageren og længden af projektionen af samme vektor med vektoren, som går fra Jordens centrum til GPS-modtagerens foreløbige position.

4.3.3 Konkret eksempel af linearisering

Der bliver nu gennemgået en linearisering af satellitterne 4, 14, 16 og 18, for at opstille matricerne \underline{A} og \underline{b} . Vi tager udgangspunkt i udtrykket for polynomiet beskrevet ovenfor. Lad os bestemme linearisering af udtrykket for afstanden mellem modtageren og satellit nr. 4. Vi i afsnit 4.1 (Variableliste og data) at satellit 4 ligger i punktet:

$$Sat4 = (4396623.907, -15219512.421, 21395963.449)$$

med en pseudoafstand til modtageren:

$$P_4 = 22745185m$$

Vi vælger udviklingspunktet til at være lig med den omtrentlige position af GPS-modtageren, som er givet til at være:

$$M_0 = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \begin{bmatrix} 3504300 \\ 780800 \\ 5252100 \end{bmatrix}$$

Dette er det samme udviklingspunkt, som bliver brugt for alle lineariseringerne. Ved at indsætte værdierne i udtrykket for det lineariserede polynomie fremstår:

$$Pol_4 = \sqrt{(3504300 - 4396623.907)^2 + (780800 - (-15219512.421))^2 + (5252100 - 21395963.449)^2} + w - 22745185 + \frac{(3504300 - 4396623.907)(X - 3504300) + (780800 - (-15219512.421))(Y - 780800) + (5252100 - 21395963.449)(Z - 5252100)}{\sqrt{(3504300 - 4396623.907)^2 + (780800 - (-15219512.421))^2 + (5252100 - 21395963.449)^2}}$$

⇓

$$Pol_4 \approx 3.317639443 \cdot 10^6 - 0.03922802906 \cdot X + 0.7034000947 \cdot Y - 0.7097108347 \cdot Z + w$$

Ved at gentage denne beregning for satellit 14, 16 og 18 fremkommer disse resultater:

$$Pol_{14} \approx 5.95514290 \cdot 10^6 - 0.4394081938 \cdot X - 0.4591787890 \cdot Y - 0.7720591161 \cdot Z + w$$

$$Pol_{16} \approx 5.685033122 \cdot 10^6 - 0.5893952657 \cdot X + 0.3291562018 \cdot Y - 0.7377461728 \cdot Z + w$$

$$Pol_{18} \approx 2.72993814 \cdot 10^6 - 0.9345007536 \cdot X + 0.3521808331 \cdot Y + 0.05173975975 \cdot Z + w$$

Herfra kan vi opstille matricerne $\underline{\underline{A}}$ og $\underline{\underline{b}}$:

$$\underline{\underline{A}} = \begin{bmatrix} -0.03922802906 & 0.7034000947 & -0.7097108347 & 1 \\ -0.4394081938 & -0.4591787890 & -0.7720591161 & 1 \\ -0.5893952657 & 0.3291562018 & -0.7377461728 & 1 \\ -0.9345007536 & 0.3521808331 & 0.05173975975 & 1 \end{bmatrix}$$

$$\underline{\underline{b}} = \begin{bmatrix} -3.317639443 \cdot 10^6 \\ -5.955142906 \cdot 10^6 \\ -5.685033122 \cdot 10^6 \\ -2.729938144 \cdot 10^6 \end{bmatrix}$$

Da vi nu har opstillet de to matricer kan ligningssystemet $\underline{\underline{A}} \underline{\underline{x}} = \underline{\underline{b}}$ løses ved brug af Gauss-Jordan elimination. Det bliver gjort i Maple ved brug af **LinearSolve**-funktionen. Herfra fremkommer resultatet:

$$\begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix} = \begin{bmatrix} 3504309.92 \\ 780753.45 \\ 5252120.18 \\ -1867.72 \end{bmatrix}$$

Hermed kan vi konkludere, at GPS modtagerens position, ifølge denne udregning, befinder sig i punktet:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 3504309.92 \\ 780753.45 \\ 5252120.18 \end{bmatrix}$$

med en urfejl på:

$$w = -1867.72m$$

Det svarer til et punkt, som er ca. 51.716 meter fra GPS-modtagerens egentlige punkt. Det er en relativt stor afvigelse. I de næste afsnit vil vi gennemgå, hvordan GPS-modtagerens position kan blive estimeret mere præcist.

4.4 Iterativ Taylor-Udvikling

Når ligningssystemet (4) lineariseres, bruges en Taylor-udvikling af 1. grad for at konvertere hver ligning til en lineær ligning. Denne lineære ligning er en approksimation af afstandsligningerne, og de løsninger der beregnes ud fra de ligninger er dermed også kun approksimationer.

Taylor-udviklingen foretages i punktet $M_0 = (3504300, 780800, 5252100)$, som er en approksimeret værdi af GPS-modtagerens egentlige position. Et Taylor-polynomium modellerer mest præcist den egentlige ligning i og omkring udviklingspunktet, så jo længere væk fra udviklingspunktet den egentlige position er, jo mindre præcist kan den bestemmes.

En mulig løsning er at bruge Gauss-Newton metoden, hvor der foretages flere Taylor-udviklinger i serie. For hver ny Taylor-udvikling, bruges forrige beregningens resultat som det nye udviklingspunkt.

Iteration	X	Y	Z	Afstand fra modtager
0	3504300.00	780800.00	5252100.00	58.4469845244388976937102741334
1	3504309.94	780753.44	5252120.20	13.6966188369066208056006284434
2	3504309.94	780753.44	5252120.20	13.6966134989807622225678795113
3	3504309.94	780753.44	5252120.20	13.6966134989807621661422189366
4	3504309.94	780753.44	5252120.20	13.6966134989807621661424077962
5	3504309.94	780753.44	5252120.20	13.6966134989807621661424436234
6	3504309.94	780753.44	5252120.20	13.6966134989807621661424576847
7	3504309.94	780753.44	5252120.20	13.6966134989807621661424856850
8	3504309.94	780753.44	5252120.20	13.6966134989807621661421205751

Tabel 5: En tabel over foreløbige positioner bestemt ved iterativ Taylor-udvikling for satellit 4, 14, 16 og 18. Kolonnen "Afstand fra modtager" beskriver afstanden fra den foreløbige position til den kendte position af GPS-modtageren.

Tabel 5 viser udviklingen i den beregnede position af GPS-modtageren ud fra data fra satellit 4, 14, 16 og 18. Række 0 viser positionen M_0 . Hver række derefter benytter forrige rækkes koordinater som et udviklingspunkt for en ny Taylor-udvikling.

Ud fra denne tabel er det tydeligt at allerede efter første iteration er den beregnede position langt mere nøjagtig end M_0 . Yderligere iterationer er mindre betydningsfulde: De efterfølgende iterationer konvergerer hurtigt mod en værdi, der er omtrent lige så nøjagtig som første iterations værdi.

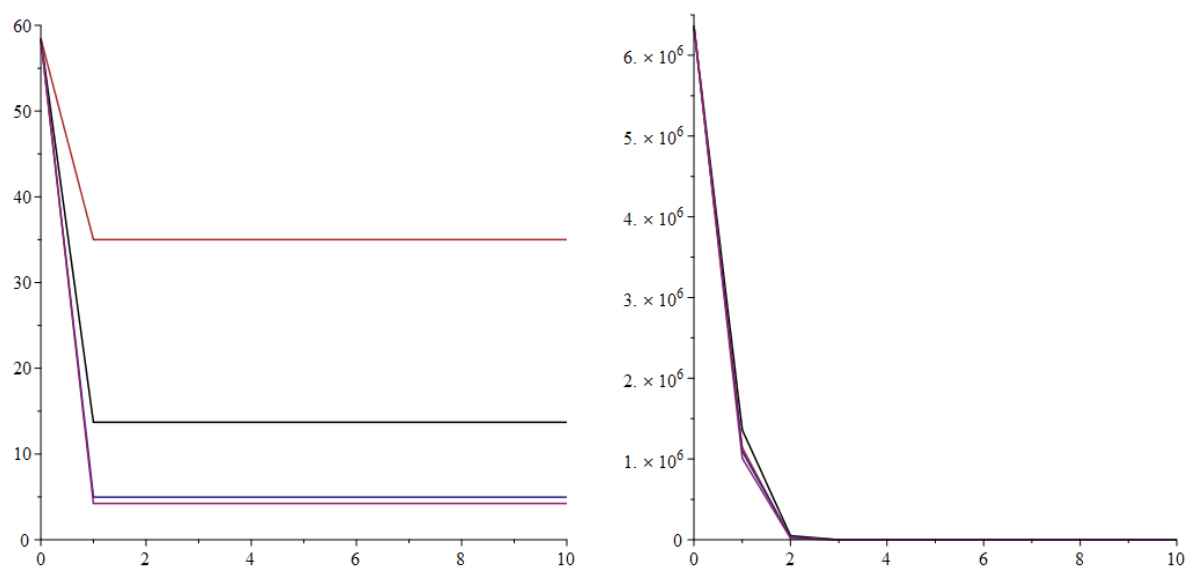
Afstandene i tabellen er vist med 30 decimalers nøjagtighed. Da beregningerne bl.a. er baseret på satellitters pseudoafstande, der er målt i meter, er det ikke realistisk at foretage afstandsbestemmelser med større præcision. Afstandene i tabellen er kun vist med 30 decimaler for at illustrere udviklingen og konvergensens af hver iterations beregnede afstand.

Figur 18 på næste side viser konvergensens i afstandsberregning for den iterative Taylor-udvikling foretaget på fire forskellige grupper af satellitter. Venstre plot er med M_0 som udviklingspunkt, og højre plot er med Jordens centrum som udviklingspunkt. For M_0 -udviklingen er det for alle grupper fælles, at kun en enkelt iteration er nødvendig, og at alle yderligere iterationer ikke har nogen betydning for den beregnede afstand. Ligeledes gælder det for 0-udviklingen at værdierne konvergerer efter 4 iterationer (se

Iteration	Afstand fra modtager			
	4, 14, 16, 18	4, 14, 24, 25	16, 18, 24, 25	14, 16, 18, 24
0	6361972.6	6361972.6	6361972.6	6361972.6
1	1364079.8	1141426.1	1095709.9	1010780.9
2	49376.0	19838.7	29447.3	19210.4
3	51.9	22.3	19.0	4.9
4	13.7	35.0	5.0	4.2
5	13.7	35.0	5.0	4.2
6	13.7	35.0	5.0	4.2
7	13.7	35.0	5.0	4.2
8	13.7	35.0	5.0	4.2

Tabel 6: Afstandsberregningerne pr. iteration for 4 forskellige grupper af satellitter, med Jordens centrum som udviklingspunkt. Fælles for alle grupper er, at afstandene konvergerer efter 4 iterationer.

også Tabel 6). Uanset udviklingspunkt konvergerer hver gruppe mod samme værdi, men ingen gruppe kan fuldstændigt lukke afstanden fra udviklingspunktet til den kendte position M .



Figur 18: Et udvalg af afstandsberregninger plottet efter antallet af Taylor-iterationer for forskellige satellitters data. For venstre plot er iterationerne begyndt med M_0 som udviklingspunkt, og for højre plot er Jordens centrum $(0, 0, 0)$ brugt. For begge plots konvergerer hver enkelt gruppe af satellitter mod samme værdi:

Satellit 4, 14, 16 og 18 (**Sort**) konvergerer mod 13.7.

Satellit 4, 14, 24 og 25 (**Rød**) konvergerer mod 35.0.

Satellit 16, 18, 24 og 25 (**Blå**) konvergerer mod 5.0.

Satellit 14, 16, 18 og 24 (**Lilla**) konvergerer mod 4.2.

Det er vigtigt at påpege, at disse grafer ikke kan bruges til at bestemme den mest præcise kombination af satellitter i praksis, da man her ikke kender den korrekte position og det vil variere fra måling til måling hvilke af satellitterne, der resulterer i den mest nøjagtige position.

Her kan man se, at afstanden til den egentlige position varierer en del mellem hver kombination af 4 satellitter. Dette er naturligvis til at forvente, da der i praksis er fejlmålinger for satellitterne. For at mindske effekten af fejlmålingerne for de enkelte kombinationer af satellitter, kan man bestemme flere kombinationer og bestemme gennemsnittet.

4.5 Kombinatorisk løsning

I den øvre gennemgang er der taget udgangspunkt i fire satellitter per Gauss-Newton gennemløb, som er vist i Figur 18. Da der skal 4 satellitter til at bestemme en position, og der er opgivet 6 satellitter, kan man beregne den samlede mængde af kombinationer af satellitter ved brug af binomialkoefficienten:

$$\binom{n}{k} = \binom{6}{4} = \frac{6!}{4! \cdot (6-4)!} = 15$$

Derfor er der 15 forskellige kombinationer af satellitter. Vi vil nu undersøge, hvilken position fremkommer, hvis man bestemmer positionen ud fra alle kombinationerne af satellitterne og tager gennemsnittet af positionerne.

Under bilaget, er der inkluderet et Maple-Script, som udfører disse beregninger. Den bestemmer først alle kombinationerne af 4 satellitter og herefter gennemsnittet af alle x-, y-, z-positionerne og urfejlene. Resultatet er den følgende position og urfejl:

X position	Y position	Z position	Urfejl
3504317.7147	780749.56549	5252139.43327	-1848.95656

Det giver en position som er ca. 11.69922 m. fra den egentlige position. Det er ønsket af undersøge, om der er en forbedring i resultatets præcision, hvis der bliver udført Gauss-Newton metoden for alle de 15 forskellige kombinationer af satellitter. Dette nye script er inkluderet bilaget. Herfra fremkommer disse x-, y-, z-positioner og urfejl:

X position	Y position	Z position	Urfejl
3504317.71472	780749.56558	5252139.43316	-1848.9567

Det giver en position som er ca. 11.69908 m. fra den egentlige position. Dvs. at metoden som inkorporerer Gauss-Newton metoden er ca. 0.00014 m. tættere på den egentlige position. Det er en minimal forbedring, men en forbedring som nok ikke har nogen praktisk effekt, da det svarer til en forbedring på kun 0.14 millimeter. Dette er som forventet, givet den meget lille effekt på præcisionen Gauss-Newton metoden var vist til at have i forrige afsnit.

4.6 Mindste Kvadraters Metode i 3D

Ligningssystemet (4) har vi nu løst ved hjælp af Gauss-Newtons metode og en kombinatorisk metode, som inddrog alle kombinationer af satellitterne. Der er derved fundet en approksimation af modtagerens position samt en værdi af ur-fejlen. Approksimationen vil med vores data ramme ca. 10 meter forbi den faktiske position, hvilket i mange anvendelser vil være acceptabelt. Men vi ønsker at undersøge hvor vidt det er muligt at reducere fejlen i positionsberegningen yderligere. En oplagt vej at gå, er igen at inddrage data fra alle satellitter til bestemmelse af positionen og ur-fejlen:

$$\underline{x} = \begin{bmatrix} X \\ Y \\ Z \\ \overline{w} \end{bmatrix}$$

Vi har allerede forsøgt en kombinatorisk løsning til at inddrage alle satellitterne, men denne metode er dyr computeringsmæssigt og giver stadig en unøjagtighed på nogle meter på den givne data. Retfærdiggørelsen af den kombinatoriske metode var at det at inddrage flere satellitter end vi har frie variable i næsten alle tilfælde et inkonsistent system, da der (virtuelt) altid vil være fejl i positions- og/eller tidsmålingerne for satellitterne/modtager.

I mindste kvadraters metode benytter vi alle 6 satellitter til at konstruere et ligningssystem med flere ligninger end ubekendte, som vi herefter kan approksimere den bedste løsning til med kun én enkelt matrix-ligning.

Vi betragter igen det lineære ligningssystemer på formen:

$$\underline{\underline{A}} \underline{x} = \underline{\underline{b}}$$

hvor $\underline{\underline{A}}$ er en $n \times m$ - matrix og $\delta(A) = r$.

Her vil løsningen igen \underline{x} opfylde følgende:

$$\underline{\underline{b}} - \underline{\underline{A}} \underline{x} = 0$$

For vores ligningssystem med alle 6 satellitter, vil dette naturligvis ikke være tilfældet da vi har flere ligninger end ubekendte og herved $n > m$. Da løber vi igen ind i samme problem som i afsnit 3.3.

$$\underline{\underline{A}} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & . & . & . \\ . & . & . & . \\ a_{n,1} & . & . & a_{n,4} \end{bmatrix}$$

Her antages det, at matrixen har rang 4 $\delta(\underline{\underline{A}}) = 4$, hvilket betyder at søjlerne i $\underline{\underline{A}}$ er lineært uafhængige og herved at billedrummet for den lineære afbildning med $\underline{\underline{A}}$ som afbildningsmatrix er 4-dimensionel: $n > m = r = 4$. I praksis betyder dette at GPS'erne der bruges til positionsberegningerne ikke ligger på en ret linje, da dette vil resultere i lineære systemer, der har lavere rang end 4. I disse scenarier vil en rang på hhv. 1 eller 2 i praksis betyde at man ikke kan præcisere modtagerens position til mere end hhv. én eller to dimensioner. Dette er det samme scenarie som i 2D, hvor satellitterne ligger på linje og derfor ikke kan bruges i samme system til positionsbestemmelse. Dette kommer dog ikke til udtryk i praksis når der er medtaget urfejl, idet denne næsten altid vil forårsage systemer med fuld rang.

Koefficientmatrixen $\underline{\underline{A}}$ er da ikke kvadratisk og derfor ikke invertibel. En eksakt løsning til dette ligningssystem findes lige som før ikke:

$$\underline{\underline{b}} - \underline{\underline{A}} \underline{x} \neq 0$$

Det eneste tilfælde, hvor der ville eksistere en eksakt løsning er, hvis $\underline{\underline{b}}$ er en linearkombination af koefficientmatrixens søjler, $\underline{\underline{b}} \in C(A)$. Herved forstås det, at $\underline{\underline{b}}$ skulle være en del af $\underline{\underline{A}}$'s billedmængde. Som nævnt er dette dog næsten umuligt i vores tilfælde givet de små fejl der indgår i datapunkterne og dermed de koefficienter vi får i matrixen.

Det er nu her mindste kvadraters metode ligeledes for situationen tidligere behandlet i 2D kan tages i brug. Et residual er igen defineret som:

$$\underline{\underline{R}} = \underline{\underline{b}} - \underline{\underline{A}} \underline{x}$$

Hvor det er følgende vi er interesseret i:

$$\min_{\underline{x}} \sum_{i=1}^n R_i^2$$

Funktionen vi ønsker at minimere kalder vi nu for $\phi(x)$ og den kan skrives op således:

$$\begin{aligned}\phi(\underline{x}) &= \sum_{i=1}^n R_i^2 = \underline{R}^T \underline{R} = (\underline{b} - \underline{A}\underline{x})^T (\underline{b} - \underline{A}\underline{x}) \Rightarrow \\ \phi(\underline{x}) &= \underline{b}^T \underline{b} - \underline{x}^T \underline{A}^T \underline{b} - \underline{b}^T \underline{A} \underline{x} + \underline{A}^T \underline{x}^T \underline{A} \underline{x}\end{aligned}$$

Ved minimering af dette udtryk fås et system af lineære ligninger, hvortil der findes en eksakt løsning. Belægget for dette vil blive gennemgået senere. Så lad os første tage kig på selve minimerings processen og hvilke informationer vi kan få ud af dette.

4.6.1 Reducering af summen af de kvadrerede residualer

Det vides at et udtryk antager den mindst eller størst mulige værdi det sted, hvor dens gradient er lig 0. Herved skal mindste kvadraters løsning x_0 opfylde følgende:

$$\nabla \phi(x_0) = 0$$

Idet der er tale om et samlet udtryk bestående af summen af flere led, så kan det med fordel opdeles i disse led ved udregning af gradienten:

$$\begin{aligned}\nabla(\underline{A}^T \underline{x}^T \underline{A} \underline{x}) &= \frac{\partial}{\partial \underline{x}} (\underline{A}^T \underline{x}^T \underline{A} \underline{x}) = \underline{A}^T \underline{A} \frac{\partial}{\partial \underline{x}} (\underline{x}^T \underline{x}) = 2 \underline{A}^T \underline{A} \underline{x} \\ \nabla(\underline{x}^T \underline{A}^T \underline{b}) &= \frac{\partial}{\partial \underline{x}} (\underline{x}^T \underline{A}^T \underline{b}) = \left(\frac{\partial}{\partial \underline{x}} (\underline{b}^T \underline{A} \underline{x}) \right)^T = (\underline{b}^T \underline{A})^T = \underline{A}^T \underline{b} \\ \nabla(\underline{b}^T \underline{A} \underline{x}) &= \frac{\partial}{\partial \underline{x}} (\underline{b}^T \underline{A} \underline{x}) = \left(\frac{\partial}{\partial \underline{x}} (\underline{x}^T \underline{A}^T \underline{b}) \right)^T = (\underline{A}^T \underline{b})^T = \underline{b}^T \underline{A} \\ \nabla(\underline{b}^T \underline{b}) &= 0\end{aligned}$$

Herved fås gradienten til:

$$\nabla \phi(\underline{x}) = 2 \underline{A}^T \underline{A} \underline{x} - 2 \underline{A}^T \underline{b} \quad (8)$$

Og i x_0 :

$$\begin{aligned}\nabla \phi(\underline{x}_0) &= 2 \underline{A}^T \underline{A} \underline{x}_0 - 2 \underline{A}^T \underline{b} \Rightarrow \\ \nabla \phi(\underline{x}_0) &= 2 [\underline{x}_0^T \underline{A}^T \underline{A} - \underline{b}^T \underline{A}]\end{aligned}$$

Dette udtryk (8) sættes nu lig 0, da der her må være tale om et ekstremum:

$$\begin{aligned}\nabla \phi(\underline{x}_0) &= 2 \underline{A}^T \underline{A} \underline{x}_0 - 2 \underline{A}^T \underline{b} = 0 \Rightarrow \\ 2 \underline{A}^T \underline{A} \underline{x}_0 &= 2 \underline{A}^T \underline{b} \Rightarrow \\ \underline{A}^T \underline{A} \underline{x}_0 &= \underline{A}^T \underline{b} \Rightarrow\end{aligned} \quad (9)$$

$$\underline{x}_0 = (\underline{A}^T \underline{A})^{-1} \underline{A}^T \underline{b} \quad (10)$$

Som sagt skulle summen af de kvadrerede residualer minimeres og lige nu ved vi ikke om løsningen til (12) er et maksimum, lokalt minimum eller et saddepunkt mm. Dette kan dog bedømmes ved at finde den anden afledede af funktionen

$$\nabla(\phi(\underline{x}))$$

altså Hessematrixen, $\underline{\underline{H}}$. Idet den vil beskrive et globalt minimum, hvis den er positivt afgrænset.

$$H(\phi(\underline{x})) = \frac{\partial}{\partial \underline{x}}(\nabla(\phi(\underline{x}))) = 2\underline{\underline{A}}^T \underline{\underline{A}} \quad (11)$$

Lad os nu undersøge matrixen $\underline{\underline{A}}^T \underline{\underline{A}}$. Det ses at der er tale om en kvadratisk 4×4 og symmetrisk matrix da følgende to regler gælder for transponering (jævnfør undervisningen):

$$(A^T)^T = A$$

$$(AB)^T = B^T A^T$$

Herved kan følgende udledes:

$$(\underline{\underline{A}}^T \underline{\underline{A}})^T = \underline{\underline{A}}^T (\underline{\underline{A}}^T)^T = \underline{\underline{A}}^T \underline{\underline{A}}$$

At en matrix er lig med sin egen transponerede gør sig kun gældende for symmetriske matricer. Fra før vides det at rangen af $\underline{\underline{A}}$ er 4 og herved må $\underline{\underline{A}}$ have 4 lineært uafhængige søjler/vektorer. Herved findes der kun en løsning, nemlig nul-løsningen på:

$$\underline{\underline{A}} \underline{x} = 0 \Rightarrow$$

$$a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,4}x_4$$

$$a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,4}x_4$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,4}x_4 \Leftrightarrow$$

$$\underline{x} = \underline{0}$$

Udtrykket er da uegentligt og $\ker(\underline{\underline{A}}) = \{0\}$. Derudover ses det at $\underline{\underline{A}}$ og $\underline{\underline{A}}^T \underline{\underline{A}}$ har den samme kerne da:

$$\underline{\underline{A}}^T \underline{\underline{A}} \underline{x} = 0 \Rightarrow$$

$$\underline{x}^T \underline{\underline{A}}^T \underline{\underline{A}} \underline{x} = 0 \Rightarrow$$

$$(\underline{\underline{A}} \underline{x})^T \underline{\underline{A}} \underline{x} = 0 \Rightarrow$$

$$||\underline{\underline{A}} \underline{x}||^2 = 0 \Rightarrow$$

$$\underline{\underline{A}} \underline{x} = 0$$

Herved har matrixen $\underline{\underline{A}}^T \underline{\underline{A}}$ den samme kerne som $\underline{\underline{A}}$, hvilket vil sige at $\underline{\underline{A}}^T \underline{\underline{A}}$ består af lineært uafhængige vektorer. Samtidigt ved vi, at 0 ikke er en egenværdi for matrixen idet kernen kun består af nul-vektoren. Ydermere kan det konkluderes at $\underline{\underline{A}}^T \underline{\underline{A}}$ må have positive egenværdier. Vi kan opstille følgende udtryk, hvor x er en egenvektor:

$$\underline{\underline{A}}^T \underline{\underline{A}} \underline{x} = \lambda \underline{x} \Rightarrow$$

$$\underline{\underline{A}}^T \underline{\underline{A}} \underline{x} \cdot \underline{x} = \lambda \underline{x} \cdot \underline{x} \Rightarrow$$

$$||\underline{\underline{A}} \underline{x}||^2 = \lambda |\underline{x}|^2 \geq 0$$

Altså er $\underline{\underline{A}}^T \underline{\underline{A}}$ en 4×4 , symmetrisk, non-singulær matrix med positive egenværdier. Herved er den hvad man kalder positivt afgrænset og da vil løsningen på (2) være et globalt minimum. I forlængelse af dette så ses det at ligningsystemet rent faktisk også har en eksakt løsning da $\delta(\underline{\underline{A}}^T \underline{\underline{A}}) = 4$, hvilket betyder den har fuld rang. For ligningsystemet, findes der derfor en eksakt løsning x_0 , som svarer til det globale minimum for gradienten og altså den løsning, der reducerer summen af de kvadrerede residualer, fejlstørrelsen, mest muligt.

4.6.2 Anvendt i praksis

Vi ønsker nu at anvende metoden på de 6 satellitter, som er givet i opgavebeskrivelsen. Vi benytter udtrykket, der udledes når gradienten sættes til 0, hvorved de kvadrerede residualer er mindst mulige:

$$\underline{\underline{x}}_0 = \begin{bmatrix} \underline{\underline{X}} \\ \underline{\underline{Y}} \\ \underline{\underline{Z}} \\ \underline{\underline{w}} \end{bmatrix} = (\underline{\underline{A}}^T \underline{\underline{A}})^{-1} \underline{\underline{A}}^T \underline{\underline{b}}$$

I denne ligning er $\underline{\underline{A}}$ vores designmatrix og $\underline{\underline{b}}$ vores løsningsvektor:

$$\underline{\underline{A}} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} \end{bmatrix} \quad \underline{\underline{b}} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}$$

Vores matrixligning ser altså ud således:

$$\left(\begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & a_{4,1} & a_{5,1} & a_{6,1} \\ a_{1,2} & a_{2,2} & a_{3,2} & a_{4,2} & a_{5,2} & a_{6,2} \\ a_{1,3} & a_{2,3} & a_{3,3} & a_{4,3} & a_{5,3} & a_{6,3} \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} & a_{5,4} & a_{6,4} \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} \end{bmatrix} \right)^{-1} \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & a_{4,1} & a_{5,1} & a_{6,1} \\ a_{1,2} & a_{2,2} & a_{3,2} & a_{4,2} & a_{5,2} & a_{6,2} \\ a_{1,3} & a_{2,3} & a_{3,3} & a_{4,3} & a_{5,3} & a_{6,3} \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} & a_{5,4} & a_{6,4} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}$$

Satelliternes værdier indsættes nu i udtrykket. For illustrationens skyld er der kun brugt 6 decimaler, men i de faktiske udregninger, som kan ses i bilaget, så er der brugt 30 decimaler.

$$\underline{\underline{A}} = \begin{bmatrix} -0.0392276 & 0.703397 & -0.709710 & 1 \\ -0.439407 & -0.459176 & -0.772058 & 1 \\ -0.589398 & 0.329157 & -0.737746 & 1 \\ -0.934504 & 0.352183 & 0.0517399 & 1 \\ 0.455249 & 0.808052 & -0.373908 & 1 \\ 0.242214 & -0.713029 & -0.657966 & 1 \end{bmatrix} \quad \underline{\underline{b}} = \begin{bmatrix} -3.31762 \cdot 10^6 \\ -5.95507 \cdot 10^6 \\ -5.68504 \cdot 10^6 \\ -2.72996 \cdot 10^6 \\ 260560 \\ -3.16544 \cdot 10^6 \end{bmatrix}$$

Udregningen bliver:

$$\left(\begin{bmatrix} -0.0392276 & -0.439407 & -0.589398 & -0.934504 & 0.455249 & 0.242214 \\ 0.703397 & -0.459176 & 0.329157 & 0.352183 & 0.808052 & -0.713029 \\ -0.709710 & -0.772058 & -0.737746 & 0.0517399 & -0.373908 & -0.657966 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.0392276 & 0.703397 & -0.709710 & 1 \\ -0.439407 & -0.459176 & -0.772058 & 1 \\ -0.589398 & 0.329157 & -0.737746 & 1 \\ -0.934504 & 0.352183 & 0.0517399 & 1 \\ 0.455249 & 0.808052 & -0.373908 & 1 \\ 0.242214 & -0.713029 & -0.657966 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} -0.0392276 & -0.439407 & -0.589398 & -0.934504 & 0.455249 & 0.242214 \\ 0.703397 & -0.459176 & 0.329157 & 0.352183 & 0.808052 & -0.713029 \\ -0.709710 & -0.772058 & -0.737746 & 0.0517399 & -0.373908 & -0.657966 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -3.31762 \cdot 10^6 \\ -5.95507 \cdot 10^6 \\ -5.68504 \cdot 10^6 \\ -2.72996 \cdot 10^6 \\ 260560 \\ -3.16544 \cdot 10^6 \end{bmatrix}$$

Figur 19: Satelliternes data indsat i mindste kvadraters metode. Bemærk antallet af decimaler er væsentligt reduceret for at eksemplet kunne være på dokumentet. Dette giver en signifikant større fejl. De rigtige udregninger er udført med 30 decimaler.

Udregningen resulterer i følgende vektor:

$$x = \begin{bmatrix} 3.50432055238355648342113412962 \cdot 10^6 \\ 780753.483912360695724385872885 \\ 5.25212877070907724682898052696 \cdot 10^6 \\ -1857.40902306054961770479514 \end{bmatrix}$$

Dette svarer til en difference på 0.0582 meter fra den virkelige position, regnet ud fra den euklidiske norm på den vektor, der går fra den reelle position til den ovenstående position der er fundet med mindste kvadraters metode.

5 Diskussion

I løbet af rapporten er der blevet gennemgået flere forskellige måder, at bestemme en GPS-modtagers position baseret på satellitters positioner og afstandene mellem dem og GPS-modtageren. De mest centrale estimater for GPS-modtagerens position og metoden brugt til at bestemme den, er vist i tabellen under:

Metode	Kombinatorisk	Kombinatorisk + Gauss-Newton	Mindske Kvadraters Metode
Afvigelse	10.617298 m.	10.617167 m.	0.0582 m.

Her er den kombinatoriske metode, den metode hvori enhver kombination af 4 satellitter og deres positioner blev bestemt. Den estimerede position er hermed et gennemsnit af alle disse positioner. I den kombinatoriske metode med Gauss-Newton metoden er det samme princip brugt, men hvor hver position er blevet yderligere estimeret ved brug af Gauss-Newton metoden. Til sidst er afstanden bestemt med mindste kvadraters metode. Det er tydeligt, at se den store forbedring mindste kvadraters metode giver. For de kombinatoriske metoder, er det vigtigt at påpege, at metoden kun gavner, hvis fejlmålingerne er normalfordelt. Hvis der er systematiske fejl i systemet, vil disse fejl forblive uanset hvor mange satellitter der bliver inddraget. En fejl, som er vigtig at inddrage, er effekten af den generelle relativitetsteori. Da satellitterne er i bevægelse omkring jorden, vil deres tid gå langsommere ift. GPS-modtageren på jorden¹. Det ville have en effekt på pseudoafstanden. Denne effekt er der dog taget højde for i de atom-ure, som er ombord på satellitterne i kredsløb. Derfor kan denne effekt ses bort fra.

Alle pseudoafstandene er givet i hele meter. Det antages, at pseudoafstandene er blevet afrundet til nærmeste hele tal. Med den antagelse, er det rimeligt at sige, at pseudoafstandene har en afvigelse på ± 0.5 m. For at undersøge effekten af en variation på ± 0.5 meter, blev den estimerede position for GPS-modtageren på baggrund af satellit 4, 14, 16 og 18 bestemt, hvor pseudoafstandene bliver afviget med ± 0.5 meter. Resultatet af dette vises i tabellen:

	X	Y	Z	Urfejl (w)
Pseudoafstand -0.5 m	3504309.9376	780753.4421	5252120.2031	-1868.2027
Pseudoafstand +0.5 m	3504309.9376	780753.4421	5252120.2031	-1867.2027

Herfra ser vi, at ændringen kun har indflydelse på urfejlens værdi. Koordinaterne er meget ens. Forskellen på hvert koordinat er mindre end 1 millimeter. Derfor ser det ikke ud til, at denne afrunding har nogen egentlig effekt på estimatet af GPS-modtagerens position. Det er dog relevant at pointere, at urfejlens værdi ændrer sig med præcist 1 meter, idet pseudoafstanden ændres med 1 meter, som den gør mellem de to rækker.

¹<http://www.astronomy.ohio-state.edu/~pogge/Ast162/Unit5/gps.html>

6 Konklusion

For at sammenligne de forskellige præsenterede metoder for positionsapproksimering i rummet ud fra satellitterne begyndte vi med en geometrisk metode hvor vi bestemte den bedste løsning ud fra skæringsplaner mellem kugleligningssystemer bestående af kugler med satellitterne i centrum. Med denne metode opnåede vi en afvigelse fra det faktiske punkt på 585.6 meter, altså en relativt stor fejl.

Herefter anvendte vi forskellige iterative metoder, herunder Gauss-Newton metoden, som opstiller et lineært system ud fra en førsteordens taylor-approksimation af en funktion for afstanden til 4 udvalgte satellitter. I funktionen er der indlagt en urfejl som fjerde variabel, for at kunne tage højde for de afvigelser i målingerne, der måtte være. Der opstilles et ligningssystem ud fra satellitternes position og der løses for den ubekendte vektor. Denne vektor er en approksimation af modtagerens position. Det viser sig, at hvis man tager og repeterer denne proces et antal gange, hver gang med approksimationen fra den forrige iteration som nyt udviklingspunkt for taylor-approksimationen, så vil approksimationen af modtagerens position konvergere mod en værdi - ikke den korrekte værdi, men på deterministisk vis, uafhængigt af udviklingspunktet. Det tager dog længere tid, dvs. flere gennemløb eller iterationer, at nå derhen jo fjernere det første udviklingspunkt er fra den værdi, det konvergerer imod. Med denne metode vil der være variation i præcisionen afhængigt af hvilke 4 satellitter, der vælges som datapunkter. Fejlstørrelsen vil dog formentlig være fordelt omkring den korrekte værdi. I de konkrete stikprøver vi tog i nogle forskellige kombinationer af satellitter lå præcisionen mellem 4.2 og omkring 200 meter (vores tabel viser dog kun op til 35 m) med denne metode.

I et forsøg på at fjerne den usikkerhed der ligger i at man ikke ved hvilken kombination af 4 satellitter, der ville give den tætteste approksimation forsøgte vi os også med en kombinatorisk metode, hvori vi udregner afstanden iterativt for samtlige kombinationer af 4 satellitter valgt ud fra de i alt 6 og derefter tog et gennemsnit af deres resultater. Med denne metode opnåede vi en præcision på ± 10.617 meter.

Sidst benyttede vi mindste kvadraters metode. Vi opstiller et udtryk for den samlede fejlmængde ud fra summen af de kvadrerede residualer, hvorefter vi sætter et udtryk op for den løsning der resulterer i det globale minimum for fejlmængden og dermed er den bedste løsning på de datapunkter vi inkluderer. Med denne metode opnår vi den største grad af præcision og kommer altså helt ned på en afvigelse på kun 0.0582 meter fra den faktiske position ud fra de 6 satellitter.

Hvis der antages, at beregningen til alle fremtidige målinger har omtrent samme præcision, vurderes det, at denne løsning i praksis fungerer fint til mange af de formål GPS opfylder. Hvis der eksempelvis bruges GPS til at navigere i bil, er en afvigelse på ± 6 cm tilstrækkelig lille.

Dermed vurderer vi, at problemstillingen er opfyldt.

7 Bilag

Gauss-Newton method

(Re-)Initialize environment with packages and env. vars.

```
> restart:
  with(LinearAlgebra):
  with(plots):
  with(plottools):
  Digits:=30:
```

Provided satellite data:

```
> sat4:= [4396623.907, -15219512.421, 21395963.449, 22745185]:
  sat14:= [12508988.544, 10190642.686, 21073725.487, 20490898] :
  sat16:= [ 15690450.142, -6024729.548, 20505506.585, 20673780] :
  sat18:= [ 25180700.379, -7388281.420, 4051959.985, 23193809] :
  sat24:= [-8441248.473, -20422306.240, 15063403.391, 26237952] :
  sat25:= [-2062043.991, 17167206.349, 20373015.170, 22979470] :
```

List of satellites that we will initially look at:

```
> satList := [sat4,sat14,sat16,sat18]:
```

Approximate and real position:

```
> app_pos := <3504300, 780800, 5252100>:
  real_pos := <3504320.6, 780753.5, 5252128.8>:
```

```
> initPoint := [op(app_pos),0]:
```

Procedure for inserting satellite data into the provided equation. Interface is the satellite as a list [x, y, z, pseudopos]

```
> linearize := proc (sat):
  sqrt((x - sat[1])^2 + (y - sat[2])^2 + (z - sat[3])^2) + w -
  sat[4]
end proc:
```

Procedure for doing Taylor approximation on a satellite, from a given beginning point. Uses the linearize procedure declared above.

```

> approximate := proc (sat, beg):
    local tmp_exp := (x,y,z,w) -> linearize(sat) :
    mtaylor(tmp_exp(x,y,z,w), [x=beg[1],y=beg[2],z=beg[3], w], 2)
end proc:

```

Approximation of each of the 6 satellites:

```

> approximate(sat4, app_pos)
3.31763944140038917932031174838 106 - 0.0392280290653032854814181312247 x
+ 0.703400094720224927045962794910 y - 0.709710834413023683095244625521 z + w (1)
> approximate(sat14, app_pos)
5.95514290568067707894525788721 106 - 0.439408193889949892143158551645 x
- 0.459178788832056563964350410091 y - 0.772059116278732770203084371485 z + w (2)
> approximate(sat16, app_pos)
5.68503313078117413398460862045 106 - 0.589395265768463767757931528680 x
+ 0.329156201909414575309440884598 y - 0.737746172768311057007551471999 z + w (3)
> approximate(sat18, app_pos)
2.72993814435100502972097177373 106 - 0.934500753415284901685898899515 x
+ 0.352180832989992329499442273428 y + 0.0517397597669337330677927791449 z
+ w (4)
> approximate(sat24, app_pos)
-260539.89526255618620541816392 + 0.455244610982836133001594661277 x
+ 0.808049950462625806344051800033 y - 0.373908574023696567232193651370 z + w (5)
> approximate(sat25, app_pos)
3.16549386410151839237366732686 106 + 0.242211787617919719572311238474 x
- 0.713031889664420000983015144517 y - 0.657965785022685000554631895337 z + w (6)

```

Procedure for generating a linear system from a given list of satellites and a beginning point. Note that it can deal with more satellites than there are variables (no. of rows > variables), but this will result in an inconsistent system (unless there is no measurement error, of course).

```

> generateLinearSystem := proc (satList, beg) :
    local eqns := []:

    for local i from 1 to nops(satList) by 1 do:
        eqns := [op(eqns), approximate(satList[i], beg)]
    end do:

    GenerateMatrix(eqns, [x,y,z,w])

end proc:

```

Apply and test method on the 4 specified satellites, with the approximate position as starting point:

```
> generateLinearSystem(satList, app_pos)
[[ -0.0392280290653032854814181312247, 0.703400094720224927045962794910,
  -0.709710834413023683095244625521, 1 ],
 [ -0.439408193889949892143158551645, -0.459178788832056563964350410091,
  -0.772059116278732770203084371485, 1 ],
 [ -0.589395265768463767757931528680, 0.329156201909414575309440884598,
  -0.737746172768311057007551471999, 1 ],
 [ -0.934500753415284901685898899515, 0.352180832989992329499442273428,
  0.0517397597669337330677927791449, 1 ]],
 [ -3.31763944140038917932031174838 106
  -5.95514290568067707894525788721 106
  -5.68503313078117413398460862045 106
  -2.72993814435100502972097177373 106 ]
```

(7)

Solve the matrix equation for the unknown vector:

```
> approx_sol := LinearSolve((7))
approx_sol := [ 3.50430993757308037733173792735 106
  780753.442103340214994455010953
  5.25212020310127290097297234086 106
  -1867.70266356858885618879110492 ]
```

(8)

Procedure for calculating distance between points.

Can use both vectors and lists:

```
> distanceBetweenPoints := proc (_end, start):
  VectorCalculus[Norm] (<_end[1], _end[2], _end[3]> - <start[1],
  start[2], start[3]>);
end proc;
```

How good is the calculated solution? Distance from real position:

```
> distFromReal = distanceBetweenPoints(real_pos, <Row(approx_sol,1)
  [1], Row(approx_sol,2) [1], Row(approx_sol,3) [1]>)
distFromReal = 13.6966188369066208056006284434
```

(9)

Procedure for tracking convergence of the distance between calculated solution using Gauss-Newton

and the real position of the receiver, given a starting point of approximation.

As parameters, it uses a list of satellite coordinates (satList), a point to do the Taylor approximations from (pos) and an optional parameter choosing starting x-value for the plot (startPlot).

The procedure is hardcoded to do 10 iterations, as our testing revealed that this was more than sufficient to converge towards a value for the distance with a precision of around 20 decimal places.

```
> distanceIteration := proc ( satList, pos, startPlot := 0 ):
    Digits:=30:
    local posVec := pos:
    local dist :=[]:
    if startPlot = 0 then:
        dist:= [op(dist), <0, distanceBetweenPoints(real_pos, posVec)
    >]:
    end if:

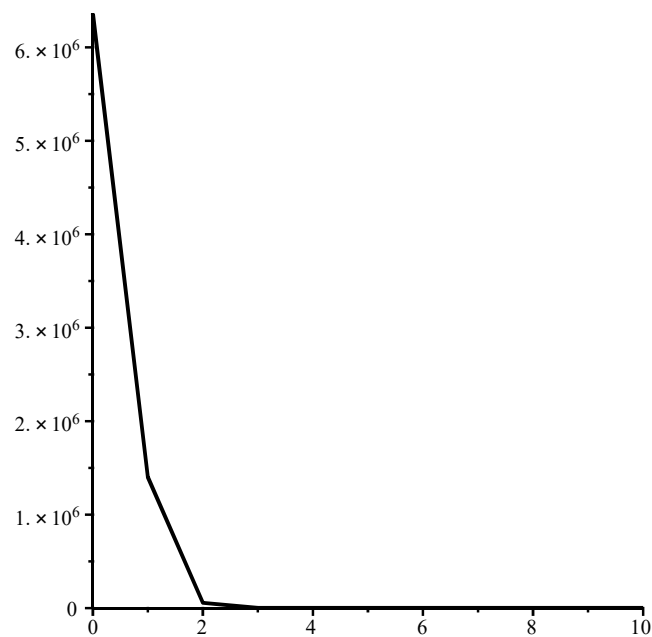
    for local i from 1 to 10 by 1 do:
        posVec := LinearSolve(generateLinearSystem(satList, posVec));
        if i > startPlot then:
            dist := [op(dist), <i, distanceBetweenPoints(real_pos,
posVec)>]:
        end if:
    end do:

    dist;
end proc:
```

Plot the result from distance iteration:

```
> plotConvergence := proc ( satList, pos, startPlot := 0 ):
    pointplot(distanceIteration(satList, pos, startPlot), connect=
true);
end proc:
```

```
> plotConvergence([sat4,sat14,sat18,sat25], <0,0,0>, 0)
```



Take only the distance computed in the last iteration of the Gauss Newton procedure:

```
> distanceIteration([sat4,sat14,sat18,sat25],<0,0,0>,0) [10] [2]
5.02695453995789418170432295035
```

(10)

Recursively find all combinations of a given size in a list of satellites.

sats = list of satellites.

combs = size of each combination.

i = pointer to start index in list.

n = length of sats.

k = iteration variable.

```
> FindAllCombs := proc (sats, combs, i, n, k):
    local allCombs2 := [];
    if (k = 0) then
        combs
    else
        for local j from i to (n) by 1 do:
            if (k - 1 = 0) then
                allCombs2 := [op(allCombs2), FindAllCombs(sats, ([op
(combs), sats[j]]), (j + 1), n, (k - 1))]:
            else
                allCombs2 := [op(allCombs2), op(FindAllCombs(sats, ([op
(combs), sats[j]]), (j + 1), n, (k - 1)))]:
            end if;
        end do;
        allCombs2
    end if;
end proc;
```


>

Same as procedure distanceIteration, but instead of outputting a "time series" of the convergence, it just saves the intermediate position that results from and is used in solving the linear system from iteration to iteration.

```
> distanceIteration2 := proc ( satList, pos ) :
    Digits:=30:
    local posVec := pos:

    for local i from 1 to 10 by 1 do:
        posVec := LinearSolve(generateLinearSystem(satList, posVec));
    end do:

    posVec;
end proc:
```

Calculate the mean position of all the combinations of the satellites.

```
> CalculateAverage := proc(sats):
    local combinations := FindAllCombs(sats, [], 1, nops(sats), 4):
    local total := nops(combinations):
    local sumX := 0:
    local sumY := 0:
    local sumZ := 0:
    local sumW := 0:
    for local i from 1 to total by 1 do:
        local solved_ := LinearSolve(generateLinearSystem
(combinations[i], app_pos)):
        (sumX, sumY, sumZ, sumW) := (sumX + solved_(1), sumY +
solved_(2), sumZ + solved_(3), sumW + solved_(4)):
    end do;
    [sumX/total, sumY/total, sumZ/total, sumW/total];
end proc:
```

Calculate the mean position found via Gauss Newton method on all combinations of the satellites.

```
> CalculateAverageGauss := proc(sats):
    local combinations := FindAllCombs(sats, [], 1, nops(sats), 4):
    local total := nops(combinations):
    local sumX := 0:
    local sumY := 0:
    local sumZ := 0:
    local sumW := 0:
    for local i from 1 to total by 1 do:
        local solved_ := distanceIteration2(combinations[i], app_pos)
:
        (sumX, sumY, sumZ, sumW) := (sumX + solved_(1), sumY +
solved_(2), sumZ + solved_(3), sumW + solved_(4)):
    end do;
    [sumX/total, sumY/total, sumZ/total, sumW/total]
end proc:
```

Comparison between the calculations of mean using the Gauss Newton method, and just the plain approximations:

```
> # Gauss newton
> CalculateAverageGauss([sat4, sat14, sat16, sat18, sat24, sat25])
[3504317.714718891862657870596340000000, 780749.565581598788622057338193000000, (11)
 5252139.433158533181309221162770000000,
 -1848.956707415260152409744826510000 ]
```

```
> # Gauss newton
> distanceBetweenPoints(real_pos, (11))
11.699083537633682346099562815600 (12)
```

```
> # average
> CalculateAverage([sat4, sat14, sat16, sat18, sat24, sat25])
[3504317.714704525951382381566110000000, 780749.565489996468144628271960000000, (13)
 5252139.433271020114299776042360000000,
 -1848.956558672996976641628405590000 ]
```

```
> # average
> distanceBetweenPoints(real_pos, (13))
11.699220124763029649033294609600 (14)
```

Compare the methods' precisions:

```
> (14)-(12)
0.00014 (15)
```

A bit disappointing? :(

Least squares method

Generate a linear system from all the satellites:

```
> generateLinearSystem([sat4, sat14, sat16, sat18, sat24, sat25],
  app_pos)
```

$$\begin{bmatrix} -0.0392276 & 0.703397 & -0.709710 & 1 \\ -0.439407 & -0.459176 & -0.772058 & 1 \\ -0.589398 & 0.329157 & -0.737746 & 1 \\ -0.934504 & 0.352183 & 0.0517399 & 1 \\ 0.455249 & 0.808052 & -0.373908 & 1 \\ 0.242214 & -0.713029 & -0.657966 & 1 \end{bmatrix}, \begin{bmatrix} -3.31762 \cdot 10^6 \\ -5.95507 \cdot 10^6 \\ -5.68504 \cdot 10^6 \\ -2.72996 \cdot 10^6 \\ 260560. \\ -3.16544 \cdot 10^6 \end{bmatrix} \quad (16)$$

Declare design matrix A and solution vector b:

```
> A := (16)[1]:
> b := (16)[2]:
```

Calculate the position using the least squares method:

```
> leastSquares := proc (designMatrix, solutionVector):
    (Transpose(designMatrix).designMatrix)^(-1).Transpose
    (designMatrix).solutionVector:
end proc:
```

```
> ans:=leastSquares(A,b)
```

$$ans := \begin{bmatrix} 3.50432055238355648342113412962 \cdot 10^6 \\ 780753.483912360695724385872885 \\ 5.25212877070907724682898052696 \cdot 10^6 \\ -1857.40902306054961770479514 \end{bmatrix} \quad (17)$$

```
> real_pos
```

$$\begin{bmatrix} 3.5043206 \cdot 10^6 \\ 780753.5 \\ 5.2521288 \cdot 10^6 \end{bmatrix} \quad (18)$$

```
> distanceBetweenPoints(real_pos, <ans[1],ans[2],ans[3]>)
0.0581729833108484372786215973941 \quad (19)
```

Show computation of least squares:

```
> A; Transpose(A); b;
```

$$\begin{bmatrix} -0.03923 & 0.70340 & -0.70971 & 1.00000 \\ -0.43941 & -0.45918 & -0.77206 & 1.00000 \\ -0.58940 & 0.32916 & -0.73775 & 1.00000 \\ -0.93450 & 0.35218 & 0.05174 & 1.00000 \\ 0.45525 & 0.80805 & -0.37391 & 1.00000 \\ 0.24221 & -0.71303 & -0.65797 & 1.00000 \end{bmatrix}$$

$$\begin{bmatrix} -0.03923 & -0.43941 & -0.58940 & -0.93450 & 0.45525 & 0.24221 \\ 0.70340 & -0.45918 & 0.32916 & 0.35218 & 0.80805 & -0.71303 \\ -0.70971 & -0.77206 & -0.73775 & 0.05174 & -0.37391 & -0.65797 \\ 1.00000 & 1.00000 & 1.00000 & 1.00000 & 1.00000 & 1.00000 \end{bmatrix}$$

$$\begin{bmatrix} -3317620.00000 \\ -5955070.00000 \\ -5685040.00000 \\ -2729960.00000 \\ 260560.00000 \\ -3165440.00000 \end{bmatrix}$$

(20)

$$\left(\begin{bmatrix} -0.0392276 & -0.439407 & -0.589398 & -0.934504 & 0.455249 & 0.242214 \\ 0.703397 & -0.459176 & 0.329157 & 0.352183 & 0.808052 & -0.713029 \\ -0.709710 & -0.772058 & -0.737746 & 0.0517399 & -0.373908 & -0.657966 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.0392276, \\ 0.703397, -0.709710, 1], \\ -0.439407, -0.459176, -0.772058, 1], \\ -0.589398, 0.329157, -0.737746, 1], \\ -0.934504, 0.352183, 0.0517399, 1], \\ 0.455249, 0.808052, -0.373908, 1], \\ 0.242214, -0.713029, -0.657966, 1] \end{bmatrix} \right)$$

0.703397, -0.709710, 1],

[-0.439407, -0.459176, -0.772058, 1],
 [-0.589398, 0.329157, -0.737746, 1],
 [-0.934504, 0.352183, 0.0517399, 1],
 [0.455249, 0.808052, -0.373908, 1],

[0.242214, -0.713029, -0.657966, 1]])

-1

$$\begin{bmatrix} -0.0392276 & -0.439407 & -0.589398 & -0.934504 & 0.455249 & 0.242214 \\ 0.703397 & -0.459176 & 0.329157 & 0.352183 & 0.808052 & -0.713029 \\ -0.709710 & -0.772058 & -0.737746 & 0.0517399 & -0.373908 & -0.657966 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -3.31762 \cdot 10^6 \\ -5.95507 \cdot 10^6 \\ -5.68504 \cdot 10^6 \\ -2.72996 \cdot 10^6 \\ 2.6056 \cdot 10^5 \\ -3.16544 \cdot 10^6 \end{bmatrix}$$

Investigate systems where the satellites are in the same plane and line respectively.

```
> s1 := <1,0,0,1>:
s2 := <1,1,0,sqrt(2)>:
s3 := <2,2,0,2*sqrt(2)>:
s4 := <3,3,0,3*sqrt(2)>:
> generateLinearSystem([s1,s2,s3,s4],<0,0,0>)
```

$$\begin{bmatrix} -1.00000 & 0.00000 & 0.00000 & 1.00000 \\ -0.50000 \sqrt{2.00000} & -0.50000 \sqrt{2.00000} & 0.00000 & 1.00000 \\ -0.50000 \sqrt{2.00000} & -0.50000 \sqrt{2.00000} & 0.00000 & 1.00000 \\ -0.50000 \sqrt{2.00000} & -0.50000 \sqrt{2.00000} & 0.00000 & 1.00000 \end{bmatrix} \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \end{bmatrix} \quad (21)$$

```
> ReducedRowEchelonForm(<(21)[1] |(21)[2]>)
```

$$\begin{bmatrix} 1.00000 & 0.00000 & 0.00000 & -1.00000 & 0.00000 \\ 0.00000 & 1.00000 & 0.00000 & 1 - \sqrt{2.00000} & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \end{bmatrix} \quad (22)$$

```
> s1 := <1,0,0,1>:
s2 := <2,0,0,2>:
s3 := <3,0,0,3>:
s4 := <4,0,0,4>:
> generateLinearSystem([s1,s2,s3,s4],<0,0,0>)
```

(23)

$$\begin{bmatrix} -1.00000 & 0.00000 & 0.00000 & 1.00000 \\ -1.00000 & 0.00000 & 0.00000 & 1.00000 \\ -1.00000 & 0.00000 & 0.00000 & 1.00000 \\ -1.00000 & 0.00000 & 0.00000 & 1.00000 \end{bmatrix}, \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \end{bmatrix} \quad (23)$$

> ReducedRowEchelonForm(<(23)[1] |(23)[2]>)

$$\begin{bmatrix} 1.00000 & 0.00000 & 0.00000 & -1.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \end{bmatrix} \quad (24)$$

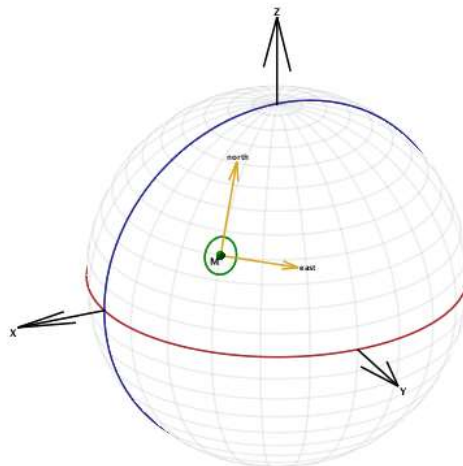
Intro plot

```
> restart:with(plots):
> ring := u -> <cos(u), sin(u)>:
> sphere := (u,v) -> <cos(u) * sin(v), sin(u) * sin(v), cos(v)>:
> equator := spacecurve([ring(u)[1], ring(u)[2], 0], u=0..2*Pi,
  color=brown):
> pmeridian := spacecurve([ring(u)[1], 0, ring(u)[2]], u=0..2*Pi,
  color=navy):
> earth := plot3d([sphere(u,v)[1], sphere(u,v)[2], sphere(u,v)[3]],
  u=0..2*Pi, v=0..Pi, transparency=0.75, color=gray, style=
  wireframeopaque):
> x := arrow(<1.5,0,0>, shape=arrow, plane=<0,0,1>, color=black):
> y := arrow(<0,1.5,0>, shape=arrow, plane=<0,0,1>, color=black):
> z := arrow(<0,0,1.5>, shape=arrow, plane=<1,1,0>, color=black):
> lambda := Pi/4:
> theta := Pi/3:
> ppoint := pointplot3d(sphere(lambda,theta), symbol=solid sphere,
  color="Green"):
> spherering := (lambda, theta, r, u) -> sphere(lambda + r * ring
  (u)[1], theta + r * ring(u)[2]):
> pring := spacecurve([spherering(lambda, theta, 0.1, u)[1],
  spherering(lambda, theta, 0.1, u)[2], spherering(lambda, theta,
  0.1, u)[3]], u=0..2*Pi, color="Green"):
> ngrad := unapply(diff(sphere(u,v),v), [u,v]):
> egrad := unapply(diff(sphere(u,v),u), [u,v]):
> nvector := arrow(sphere(lambda,theta), -0.5 * ngrad(lambda,
  theta), shape=arrow, plane=sphere(lambda,theta), color=
  "Goldenrod"):
> evector := arrow(sphere(lambda,theta), 0.5 * egrad(lambda,
  theta), shape=arrow, plane=sphere(lambda,theta), color=
  "Goldenrod"):
> xlabel := textplot3d([1.5, 0, 0, "X"], align=[BELOW,LEFT], font=
  ["Helvetica", bold, 12]):
```

```

> ylabel := textplot3d([0, 1.5, 0, "Y"], align=[BELOW,RIGHT], font=
["Helvetica", bold, 12]):
> xlabel := textplot3d([0, 0, 1.5, "Z"], align=[ABOVE], font=
["Helvetica", bold, 12]):
> ulabel := textplot3d([sphere(lambda,theta)[1], sphere(lambda,
theta)[2], sphere(lambda,theta)[3], "M"], align=[BELOW,LEFT],
font=["Helvetica", bold, 12]):
> nlabel := textplot3d([sphere(lambda,theta)[1] - 0.5 * ngrad
(lambda, theta)[1], sphere(lambda,theta)[2] - 0.5 * ngrad(lambda,
theta)[2], sphere(lambda,theta)[3] - 0.5 * ngrad(lambda, theta)
[3], "north", align=[ABOVE], font=["Helvetica", bold, 10]):
> elabel := textplot3d([sphere(lambda,theta)[1] + 0.5 * egrad
(lambda, theta)[1], sphere(lambda,theta)[2] + 0.5 * egrad(lambda,
theta)[2], sphere(lambda,theta)[3] + 0.5 * egrad(lambda, theta)
[3], "east", align=[RIGHT], font=["Helvetica", bold, 10]):
> display([equator, pmeridian, earth, x, y, z, ppoint, pring,
nvector, evector, xlabel, ylabel, xlabel, ulabel, nlabel,
elabel], axes=none, orientation=[65,65,0])

```



2D

```

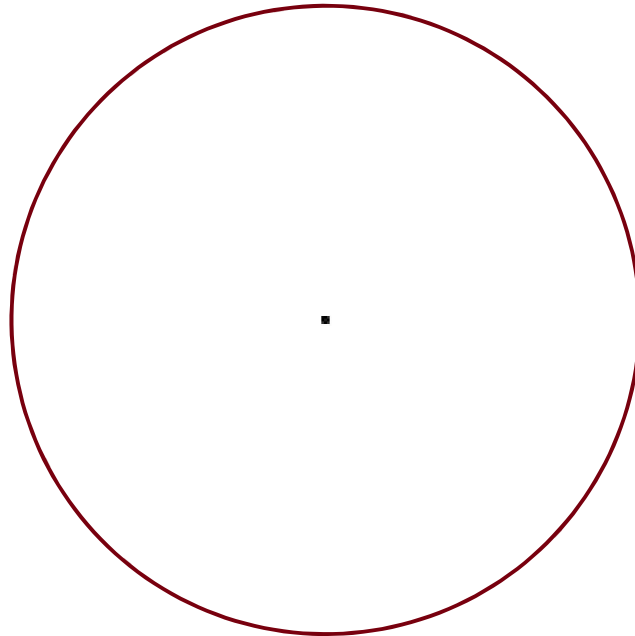
> restart:with(plots):
> eq := (xc, yc, r) -> (x-xc)^2 + (y-yc)^2 = r^2:

```

```

> circle := (x, y, r, theta) -> op([x + r * cos(theta), y + r * sin
  (theta)]):
=
> x1 := 0: y1 := 0: r1 := 1:
=
> display([pointplot([x1,y1], color=black, symbol=solidcircle),
  plot([circle(x1,y1,r1, theta), theta=0..2*Pi]], axes=None,
  scaling=constrained)

```



```

> x2 := 9/10: y2 := 5/4: r2 := 2/3:
> ys := unapply(solve(eq(x1,y1,r1) - eq(x2,y2,r2), y), x)

```

$$y_s := x \mapsto -\frac{18 \cdot x}{25} + \frac{10541}{9000} \quad (25)$$

```

> xs := solve(sqrt((x-x1)^2 + (ys(x)-y1)^2) = r1^2, x)

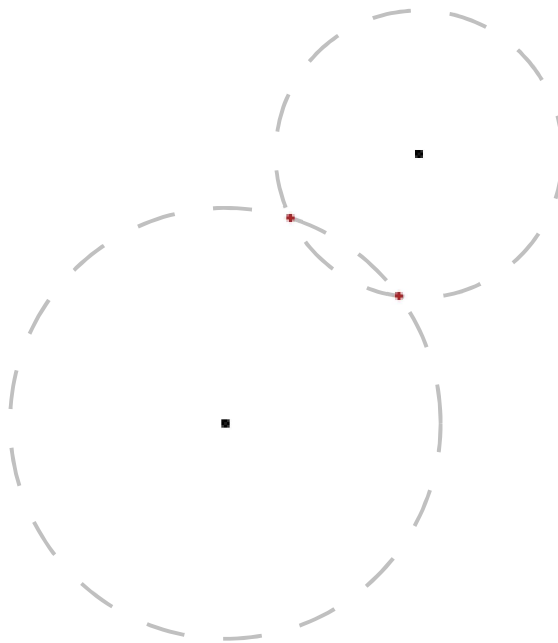
```

$$x_s := \frac{10541}{18980} + \frac{5\sqrt{11877719}}{68328}, \frac{10541}{18980} - \frac{5\sqrt{11877719}}{68328} \quad (26)$$

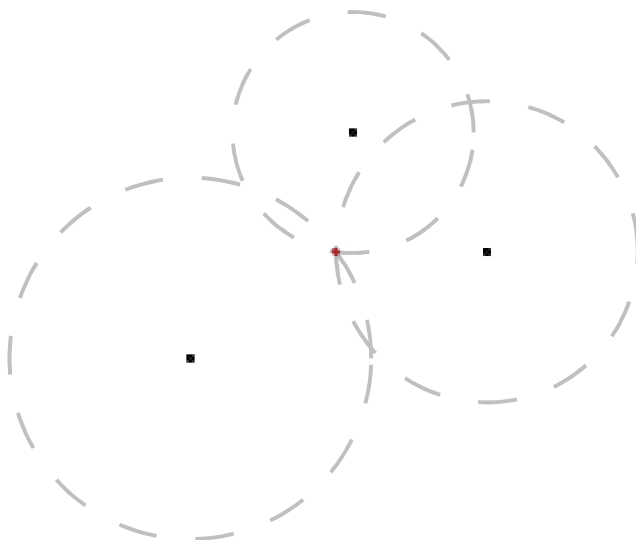
```

> display([pointplot([[x1,y1],[x2,y2]], color=black, symbol=
  solidcircle), plot([circle(x1,y1,r1, theta), theta=0..2*Pi],
  linestyle=spacedash, color=gray), plot([circle(x2,y2,r2, theta),
  theta=0..2*Pi], linestyle=spacedash, color=gray), pointplot([[xs
  [1], ys(xs[1])],[xs[2], ys(xs[2])]], color=brown, symbol=
  soliddiamond)], axes=None, scaling=constrained)

```

```
> x3 := xs[1] + r3: y3 := ys(xs[1]): r3 := 5/6:
> display([pointplot([[x1,y1],[x2,y2],[x3,y3]], color=black,
symbol=solidcircle), plot([circle(x1,y1,r1, theta), theta=0..2*
Pi], linestyle=spacedash, color=gray), plot([circle(x2,y2,r2,
theta), theta=0..2*Pi], linestyle=spacedash, color=gray), plot(
[circle(x3,y3,r3, theta), theta=0..2*Pi], linestyle=spacedash,
color=gray), pointplot([xs[1], ys(xs[1])], color=brown, symbol=
soliddiamond)], axes=None, scaling=constrained)
```

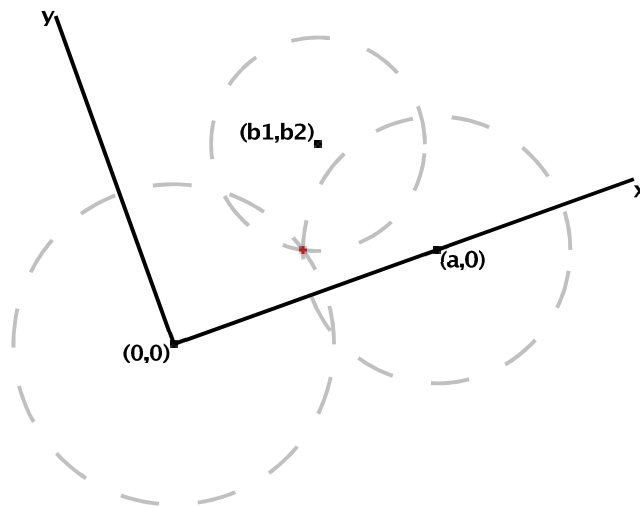


```
> display([pointplot([[x1,y1],[x2,y2],[x3,y3]], color=black,
symbol=solidcircle),
plot([circle(x1,y1,r1, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([circle(x2,y2,r2, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
```

```

plot([circle(x3,y3,r3, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
pointplot([xs[1], ys[xs[1]]], color=brown, symbol=soliddiamond),
pointplot([<x1,y1>, 7/4*(<x3,y3> - <x1,y1>)], connect=true),
pointplot([<x1,y1>, 5/4*(<-y3,x3> - <x1,y1>)], connect=true),
textplot([7/4*(x3 - x1), 7/4*(y3 - y1), "x"], align=[BELOW,
RIGHT], font=["Helvetica", bold, 12]),
textplot([5/4*(-y3 - x1), 5/4*(x3 - y1), "y"], align=[LEFT],
font=["Helvetica", bold, 12]),
textplot([x1, y1, "(0,0)"], align=[BELOW,LEFT], font=
["Helvetica", bold, 12]),
textplot([x3, y3, "(a,0)"], align=[BELOW,RIGHT], font=
["Helvetica", bold, 12]),
textplot([x2, y2, "(b1,b2)"], align=[ABOVE,LEFT], font=
["Helvetica", bold, 12])], axes=none, scaling=constrained)

```



```

> line_13 := unapply(solve((x - x1)^2 + (y - y1)^2 - r1^2 = (x -
x3)^2 + (y - y3)^2 - r3^2, y), x)

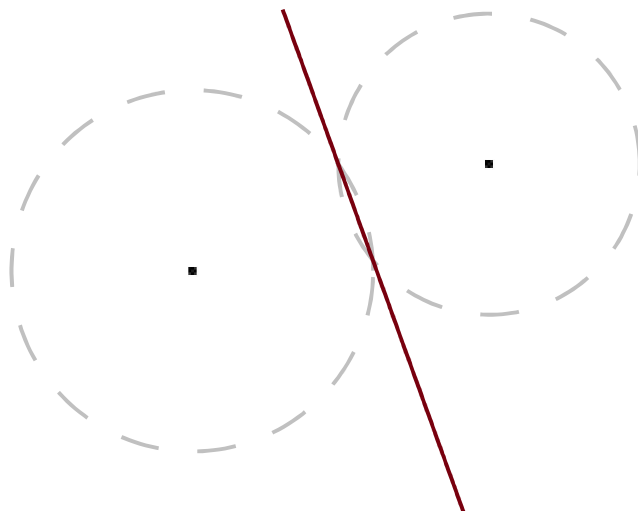
```

$$line_13 := x \mapsto \frac{-2998530 + 150 \cdot x \cdot \sqrt{11877719} - 125 \cdot \sqrt{11877719} + 2846628 \cdot x}{6 \cdot (18 \cdot \sqrt{11877719} - 263525)} \quad (27)$$

```

> display([pointplot([x1,y1],[x3,y3]), color=black, symbol=
solidcircle),
plot([circle(x1,y1,r1, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([circle(x3,y3,r3, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot(line_13(x), x=0.5..1.5)], axes=none, scaling=constrained)

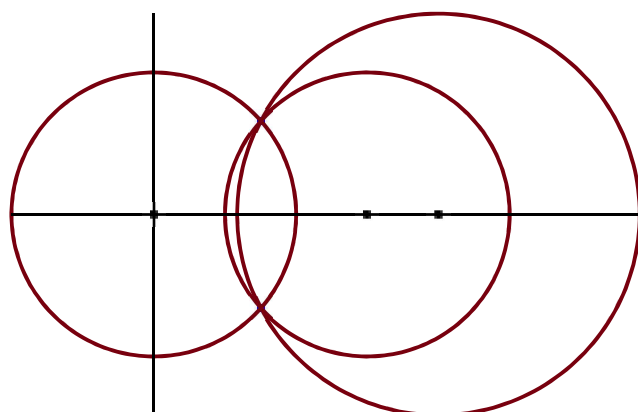
```



```

> x1 := 1.5/2: y1 := sin(solve(cos(theta) = 1.5/2, theta)):
> len := sqrt((x1 - 2)^2 + (y1)^2)
len := 1.414213562
(28)
> display([pointplot([[0,0],[1.5,0],[2,0]], color=black, symbol=
solidcircle),
pointplot([[x1,y1],[x1,-y1]], color=blue, symbol=diamond),
plot([circle(0,0,1, theta), theta=0..2*Pi]),
plot([circle(1.5,0,1, theta), theta=0..2*Pi]),
plot([circle(2,0,len, theta), theta=0..2*Pi])], tickmarks=[0,0],
scaling=constrained)

```



```

> eq1 := (x - 0)^2 + (y - 0)^2 = p1^2
eq1 := x^2 + y^2 = p1^2
(29)
> eq2 := (x - a)^2 + (y - 0)^2 = p2^2

```

$$eq2 := (x - a)^2 + y^2 = p2^2 \quad (30)$$

```
> eq3 := (x - b1)^2 + (y - b2)^2 = p3^2
```

$$eq3 := (x - b1)^2 + (y - b2)^2 = p3^2 \quad (31)$$

```
> simplify(eq2 - eq3)
```

$$(-2a + 2b1)x + a^2 + 2yb2 - b1^2 - b2^2 = p2^2 - p3^2 \quad (32)$$

```
> solve(eq1 - eq2, x)
```

$$\frac{a^2 + p1^2 - p2^2}{2a} \quad (33)$$

```
> collect( (33), x );
```

$$\frac{a^2 + p1^2 - p2^2}{2a} \quad (34)$$

```
> x1 := 0: y1 := 0: r1 := 1:
```

```
> x2 := 4/3: y2 := 0: r2 := 2/3:
```

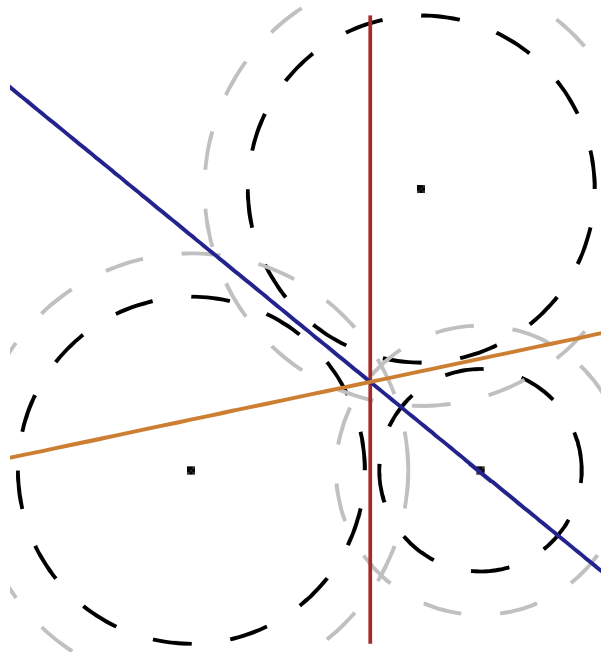
```
> xs := solve(eq(x1,y1,r1) - eq(x2,y2,r2), x):
```

```
> ys := solve(sqrt((xs-x1)^2 + (y-y1)^2) = r1^2, y)[1]:
```

```
> x3 := xs + r3 * cos(3*Pi/7): y3 := ys + r3 * sin(3*Pi/7): r3 := 1:
```

```
> w := -2/10:
```

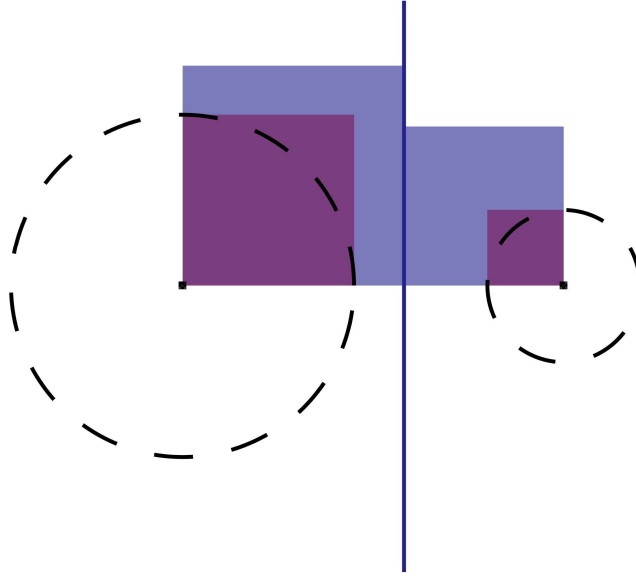
```
> display([pointplot([[x1,y1],[x2,y2],[x3,y3]], color=black,
symbol=solidcircle),
plot([circle(x1,y1,r1+w, theta), theta=0..2*Pi], linestyle=
spacedash, color=black),
plot([circle(x2,y2,r2+w, theta), theta=0..2*Pi], linestyle=
spacedash, color=black),
plot([circle(x3,y3,r3+w, theta), theta=0..2*Pi], linestyle=
spacedash, color=black),
plot([circle(x1,y1,r1, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([circle(x2,y2,r2, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([circle(x3,y3,r3, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([solve(eq(x1,y1,r1+w) - eq(x2,y2,r2+w), x), y, y=y1-(r1+w)..
y3+(r3+w)], color=brown),
plot([solve(eq(x1,y1,r1+w) - eq(x3,y3,r3+w), x), y, y=y1-(r1+w)..
y3+(r3+w)], color=navy),
plot([solve(eq(x2,y2,r2+w) - eq(x3,y3,r3+w), x), y, y=y1-(r1+w)..
y3+(r3+w)], color=gold)], axes=none, scaling=constrained, view=
[x1-(r1+w)..x3+(r3+w), y1-(r1+w)..y3+(r3+w)])
```



```

> w := -2/5:
> xs := solve(eq(x1,y1,r1+w) - eq(x2,y2,r2+w), x):
> display([shadebetween(y2, y2 + (x2 - xs), x=xs..x2, showboundary=
false, color=navy),
shadebetween(y2, y2 + (r2 + w), x=x2-(r2+w)..x2, showboundary=
false, color=red),
shadebetween(y1, y1 + (xs - x1), x=x1..xs, showboundary=false,
color=navy),
shadebetween(y1, y1 + (r1 + w), x=x1..x1+(r1+w), showboundary=
false, color=red),
pointplot([[x1,y1],[x2,y2]], color=black, symbol=solidcircle),
plot([circle(x1,y1,r1+w, theta), theta=0..2*Pi], linestyle=
spacedash, color=black),
plot([circle(x2,y2,r2+w, theta), theta=0..2*Pi], linestyle=
spacedash, color=black),
plot([xs, y, y=-1..1], color=navy)], axes=None, scaling=
constrained)

```



```
> with(LinearAlgebra):
> T := <2*a, 0, P1^2 - P2^2 + a^2; 2*b1, 2*b2, P1^2 - P3^2 + b1^2 + b2^2>
```

$$T := \begin{bmatrix} 2a & 0 & P1^2 - P2^2 + a^2 \\ 2b1 & 2b2 & P1^2 - P3^2 + b1^2 + b2^2 \end{bmatrix} \quad (35)$$

```
> ReducedRowEchelonForm(T)
```

$$\begin{bmatrix} 1 & 0 & \frac{P1^2 - P2^2 + a^2}{2a} \\ 0 & 1 & \frac{P1^2 a - P1^2 b1 + P2^2 b1 - P3^2 a - a^2 b1 + b1^2 a + b2^2 a}{2ab2} \end{bmatrix} \quad (36)$$

```
> xpos := unapply((36)[1][3], [a, P1, P2])
```

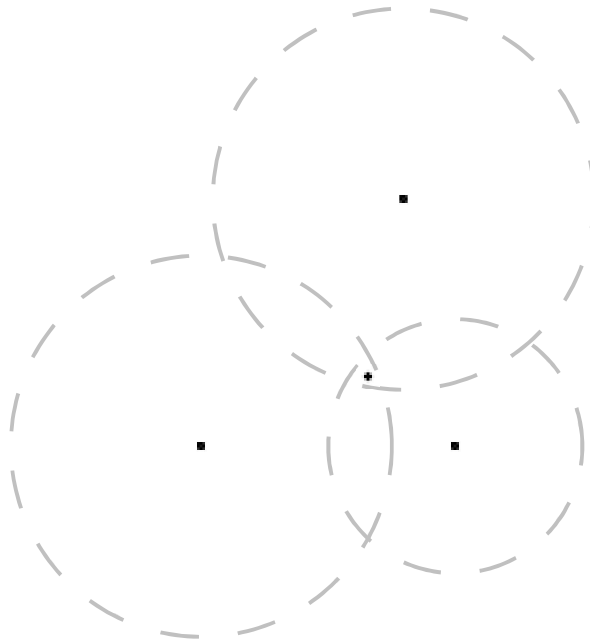
$$xpos := (a, P1, P2) \mapsto \frac{P1^2 - P2^2 + a^2}{2 \cdot a} \quad (37)$$

```
> ypos := unapply((36)[2][3], [a, b1, b2, P1, P2, P3])
```

```
ypos := (a, b1, b2, P1, P2, P3) \quad (38)
```

$$\mapsto \frac{P1^2 \cdot a - P1^2 \cdot b1 + P2^2 \cdot b1 - P3^2 \cdot a - a^2 \cdot b1 + a \cdot b1^2 + a \cdot b2^2}{2 \cdot a \cdot b2}$$

```
> display([pointplot([[x1,y1],[x2,y2],[x3,y3]], color=black,
symbol=solidcircle),
plot([circle(x1,y1,r1, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([circle(x2,y2,r2, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
plot([circle(x3,y3,r3, theta), theta=0..2*Pi], linestyle=
spacedash, color=gray),
pointplot([xpos(x2, r1, r2), ypos(x2, x3, y3, r1, r2, r3)]]],
axes=none, scaling=constrained)
```



```

> restart:with(plots):with(LinearAlgebra):
> eqCircle := (param) -> (x - param[1])^2 + (y - param[2])^2 =
  param[3]^2:
> seqCircle := (ci, cj) -> eqCircle(ci) - eqCircle(cj):
> plotLine := (ci, cj) -> plot(solve(eqCircle(ci) - eqCircle(cj),
  y), x=-3..4, color=navy):
> plotCircle := (param) -> plot([param[1] + param[3]*cos(theta),
  param[2] + param[3]*sin(theta), theta=0..2*Pi]):
> plotCorrectCircle := (param) -> plot([param[1] + (param[3]-w)*cos
  (theta), param[2] + (param[3]-w)*sin(theta), theta=0..2*Pi]):
> M := <sqrt(2), sqrt(2)>:
> w := 1/5

```

$$w := \frac{1}{5} \quad (39)$$

```

> Dist := coord -> sqrt((coord[1] - M[1])^2 + (coord[2] - M[2])^2):
> c := (x,y) -> [x, y, Dist(<x,y>) + w]:
> c1 := c(0,0):
  c2 := c(2,2):
  c3 := c(1.2,2.5):
  c4 := c(1,-1):
  c5 := c(-0.5,1):
> #A0,b0 := GenerateMatrix([eqCircle(c1),eqCircle(c2),eqCircle(c3),
  eqCircle(c4),eqCircle(c5)], [x, y])
> #A := <A0|<1,1,1,1,1>>;
> #b := b0 + <x^2+y^2,x^2+y^2,x^2+y^2,x^2+y^2,x^2+y^2>
> A,b := GenerateMatrix([seqCircle(c1,c2),seqCircle(c1,c3),
  seqCircle(c1,c4),seqCircle(c1,c5),seqCircle(c2,c3),seqCircle(c2,
  c4),seqCircle(c2,c5),seqCircle(c3,c4),seqCircle(c3,c5),seqCircle
  (c4,c5)], [x, y])

```

$$\begin{aligned}
 A, b := & \begin{bmatrix} 4 & 4 \\ 2.4 & 5.0 \\ 2 & -2 \\ -1.0 & 2. \\ -1.6 & 1.0 \\ -2 & -6 \\ -5.0 & -2. \\ -0.4 & -7.0 \\ -3.4 & -3.0 \\ -3.0 & 4. \end{bmatrix}, \left[\left[\frac{321}{25} - \left(\sqrt{2} (2 - \sqrt{2}) + \frac{1}{5} \right)^2 \right], \right. \\
 & \left[12.530000000 - \left(\sqrt{(1.2 - \sqrt{2})^2 + (2.5 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right], \\
 & \left[\frac{171}{25} - \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + \frac{1}{5} \right)^2 \right], \\
 & \left[6.090000000 - \left(\sqrt{(-0.5 - \sqrt{2})^2 + (1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right], \\
 & \left[-0.31 + 1. \left(\sqrt{2} (2. - 1. \sqrt{2}) + 0.2000000000 \right)^2 \right. \\
 & \left. - \left(\sqrt{(1.2 - \sqrt{2})^2 + (2.5 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right], \\
 & \left[-6 + \left(\sqrt{2} (2 - \sqrt{2}) + \frac{1}{5} \right)^2 - \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + \frac{1}{5} \right)^2 \right], \\
 & \left[-6.75 + 1. \left(\sqrt{2} (2. - 1. \sqrt{2}) + 0.2000000000 \right)^2 \right. \\
 & \left. - \left(\sqrt{(-0.5 - \sqrt{2})^2 + (1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right], \\
 & \left[-5.69 + 1. \left(\sqrt{(1.2 - \sqrt{2})^2 + (2.5 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right. \\
 & \left. - \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right], \\
 & \left[-6.44 + 1. \left(\sqrt{(1.2 - \sqrt{2})^2 + (2.5 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right. \\
 & \left. - \left(\sqrt{(-0.5 - \sqrt{2})^2 + (1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right], \\
 & \left[-0.75 + 1. \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right. \\
 & \left. - \left(\sqrt{(-0.5 - \sqrt{2})^2 + (1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right] \Big]
 \end{aligned} \tag{40}$$

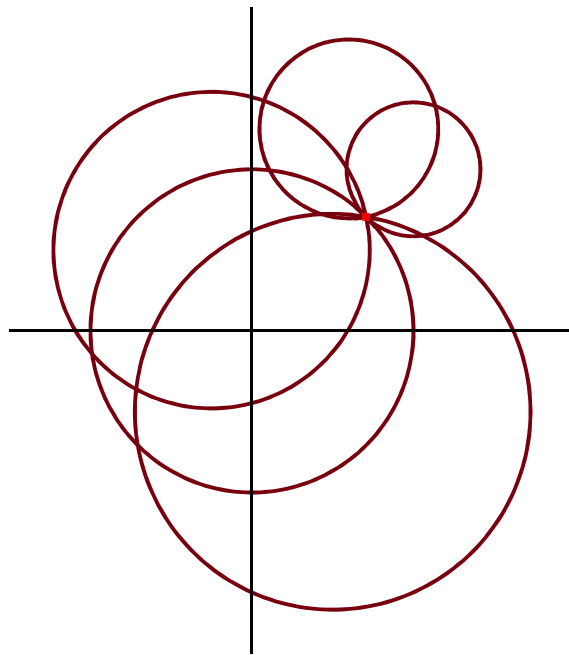
$$\begin{aligned} &> \text{sol} := (\text{Transpose}(\mathbf{A}) \cdot \mathbf{A})^{(-1)} \cdot \text{Transpose}(\mathbf{A}) \cdot \mathbf{b} \\ \text{sol} := &\left[\left[1.54007311129163 - 0.0496434696272227 \left(\sqrt{2} (2 - \sqrt{2}) + \frac{1}{5} \right)^2 \right. \right. \end{aligned} \quad (41)$$

$$\begin{aligned} &- 0.00541565123206065 \left(\sqrt{(1.2 - \sqrt{2})^2 + (2.5 - \sqrt{2})^2} + 0.2000000000 \right)^2 \\ &- 0.0313385684628577 \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + \frac{1}{5} \right)^2 \\ &+ 0.187471793483166 \left(\sqrt{(-0.5 - \sqrt{2})^2 + (1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \\ &- 0.0937810271685170 \left(\sqrt{2} (2. - 1. \sqrt{2}) + 0.2000000000 \right)^2 \\ &- 0.0815777597256070 \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \Big], \\ &\left[0.920797725426482 - 0.0478382525498691 \left(\sqrt{2} (2 - \sqrt{2}) + \frac{1}{5} \right)^2 \right. \\ &- 0.0961278093690766 \left(\sqrt{(1.2 - \sqrt{2})^2 + (2.5 - \sqrt{2})^2} + 0.2000000000 \right)^2 \\ &+ 0.0570737431176099 \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + \frac{1}{5} \right)^2 \\ &- 0.0557089990071306 \left(\sqrt{(-0.5 - \sqrt{2})^2 + (1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \\ &+ 0.0187201010921563 \left(\sqrt{2} (2. - 1. \sqrt{2}) + 0.2000000000 \right)^2 \\ &\left. \left. + 0.0886614315371423 \left(\sqrt{(1 - \sqrt{2})^2 + (-1 - \sqrt{2})^2} + 0.2000000000 \right)^2 \right] \right] \end{aligned}$$

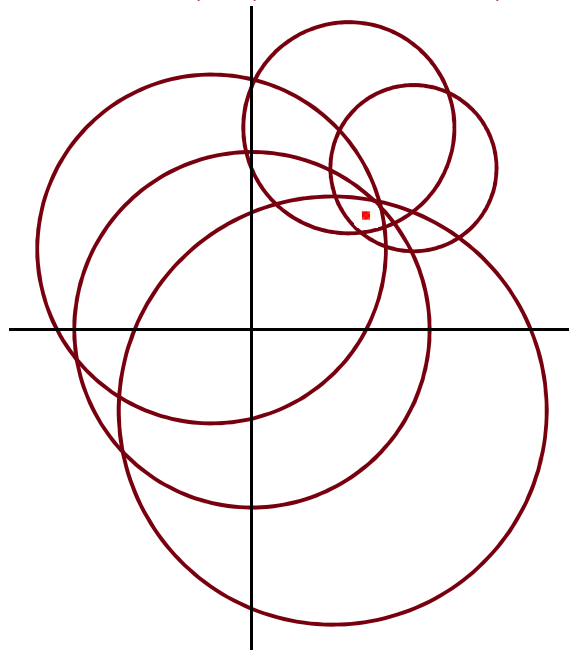
> evalf(sol)

$$\begin{bmatrix} 1.45994885325309 \\ 1.48933395242571 \end{bmatrix} \quad (42)$$

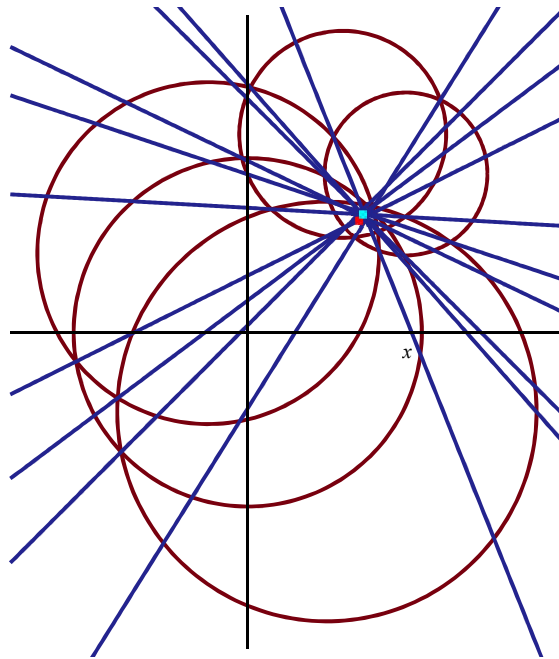
**> display([plotCorrectCircle(c1), plotCorrectCircle(c2),
plotCorrectCircle(c3), plotCorrectCircle(c4), plotCorrectCircle
(c5),
pointplot([M[1], M[2]], symbol=solidcircle, color=red)], scaling=
constrained, tickmarks=[0,0], view=[-3..4, -4..4])**



```
> display([plotCircle(c1),plotCircle(c2),plotCircle(c3),plotCircle(c4),plotCircle(c5),
pointplot([M[1],M[2]], symbol=solidcircle, color=red)], scaling=
constrained, tickmarks=[0,0], view=[-3..4, -4..4])
```



```
> display([plotCircle(c1),plotCircle(c2),plotCircle(c3),plotCircle(c4),plotCircle(c5),
plotLine(c1, c2),plotLine(c1, c3),plotLine(c1, c4),plotLine(c1, c5),plotLine(c2,c3),plotLine(c2,c4),plotLine(c2,c5),plotLine(c3, c4),plotLine(c3,c5),plotLine(c4,c5),
pointplot([[sol[1],sol[2]], [M[1],M[2]]], symbol=solidcircle,
color=[cyan,red]]), scaling=constrained, tickmarks=[0,0], view=
[-3..4, -4..4])
```



Opgaver

Indlæs Data

restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 25 :

Table of satellite data:

```
Data := Matrix([ [ 4396623.907 , -15219512.421 , 21395963.449 , 22745185 ],
                  [ 12508988.544 , 10190642.686 , 21073725.487 , 20490898 ],
                  [ 15690450.142 , -6024729.548 , 20505506.585 , 20673780 ],
                  [ 25180700.379 , -7388281.420 , 4051959.985 , 23193809 ],
                  [ -8441248.473 , -20422306.240 , 15063403.391 , 26237952 ],
                  [ -2062043.991 , 17167206.349 , 20373015.170 , 22979470 ]]) :
```

```
sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :
```

```
sat25 := Data[6] :
```

```
ppos := <3504300, 780800, 5252100> :
```

```
apos := <3504320.6, 780753.5, 5252128.8> :
```

Opgave 1

Vi skal opstille et lineart ligningssystem på formen:

$$\underline{A} \cdot \underline{x} = \underline{b}$$

hvor

$$\underline{x} = \begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \\ \bar{w} \end{bmatrix}$$

Hvor A er en koefficientmatrice for ligningssystemerne med variablerne A, Y, Z og w.

Af ligningssystemet som fremstår når dataen fra tabellen bliver indsat i Taylor Polynomiet af første orden af formelen for GPS'ens position.

Formlen er som følgende:

$$P_s := \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + w :$$

Hvor $w = c \cdot dT$

Nu bliver taylor polynomiet af formelen fundet af første orden for alle sateliterne med udviklingspunkt i den foreløbige position for GPS-modtageren. Vi har tænkt os at generalisere fremgangsmåden, men viser den første beregning her:

$$\begin{aligned} &mtaylor\left(\sqrt{(X - Data[1][1])^2 + (Y - Data[1][2])^2 + (Z - Data[1][3])^2} - Data[1][4] + w, [X \right. \\ &\quad \left. = ppos[1], Y = ppos[2], Z = ppos[3]], 2\right) \\ &3.317639441400389179320311 \cdot 10^6 + w - 0.03922802906530328548141814 X \\ &\quad + 0.7034000947202249270459627 Y - 0.7097108344130236830952448 Z \end{aligned} \quad (43)$$

GENNEM AF TAYLOR HER
TAGER UDGANGSPUNKT I:

$$P_s := \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + w :$$

OMSKRIVES SOM:

$$f(X, Y, Z) := \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} + w - P_s :$$

$$StartPunkt = \langle X_0, Y_0, Z_0 \rangle$$

$$\begin{aligned} P(X, Y, Z) = &f(X_0, Y_0, Z_0) + f'_X(X_0, Y_0, Z_0) \cdot (X - X_0) + f'_Y(X_0, Y_0, Z_0) \cdot (Y - Y_0) + f'_Z(X_0, Y_0, Z_0) \\ &\cdot (Z - Z_0) \\ &f(X_0, Y_0, Z_0) \end{aligned}$$

$$\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + w - P_s \quad (44)$$

$$\frac{d}{dX}(f(X, Y, Z)), \frac{d}{dY}(f(X, Y, Z)), \frac{d}{dZ}(f(X, Y, Z))$$

$$\begin{aligned} &\frac{2X - 2X_s}{2\sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}}, \frac{2Y - 2Y_s}{2\sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}}, \\ &\frac{2Z - 2Z_s}{2\sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}} \end{aligned} \quad (45)$$

$$df_X(X, Y, Z) := (45)[1] :$$

$$df_Y(X, Y, Z) := (45)[2] :$$

$$df_Z(X, Y, Z) := \textbf{(45)}[3] :$$

$$P_1 = f(X_0, Y_0, Z_0) + df_X(X_0, Y_0, Z_0) \cdot (X - X_0) + df_Y(X_0, Y_0, Z_0) \cdot (Y - Y_0) + df_Z(X_0, Y_0, Z_0) \cdot (Z - Z_0)$$

$$\begin{aligned} P_1 = & \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + w - P_s \\ & + \frac{(2X_0 - 2X_s)(X - X_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \\ & + \frac{(2Y_0 - 2Y_s)(Y - Y_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \\ & + \frac{(2Z_0 - 2Z_s)(Z - Z_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \end{aligned} \quad \textbf{(46)}$$

$$\begin{aligned} P(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0) := & \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + w - P_s \\ & + \frac{(2X_0 - 2X_s)(X - X_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} + \frac{(2Y_0 - 2Y_s)(Y - Y_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \\ & + \frac{(2Z_0 - 2Z_s)(Z - Z_0)}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \end{aligned}$$

$$\begin{aligned} P := (X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0) \mapsto & \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} + w - P_s \\ & + \frac{(2 \cdot X_0 - 2 \cdot X_s) \cdot (X - X_0)}{2 \cdot \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \\ & + \frac{(2 \cdot Y_0 - 2 \cdot Y_s) \cdot (Y - Y_0)}{2 \cdot \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \\ & + \frac{(2 \cdot Z_0 - 2 \cdot Z_s) \cdot (Z - Z_0)}{2 \cdot \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \end{aligned} \quad \textbf{(47)}$$

$$a := \text{coeff}(P(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0), X, 1); b := \text{coeff}(P(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0), Y, 1); c := \text{coeff}(P(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0), Z, 1); d := \text{coeff}(P(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0), w, 1);$$

$$\begin{aligned} a := & \frac{2X_0 - 2X_s}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \\ b := & \frac{2Y_0 - 2Y_s}{2\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} \end{aligned}$$

$$c := \frac{2 Z_0 - 2 Z_s}{2 \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$d := 1 \quad (48)$$

$$k := \text{coeff}(P(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0), X, 0)$$

$$K(X_s, Y_s, Z_s, P_s, X_0, Y_0, Z_0) := \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} - P_s$$

$$- \frac{(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$L(X_s, Y_s, Z_s, X_0, Y_0, Z_0) := \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} :$$

$$L(\text{Data}[1][1], \text{Data}[1][2], \text{Data}[1][3], \text{ppos}[1], \text{ppos}[2], \text{ppos}[3]) - \text{Data}[1][4]$$

$$K(\text{Data}[1][1], \text{Data}[1][2], \text{Data}[1][3], \text{Data}[1][4], \text{ppos}[1], \text{ppos}[2], \text{ppos}[3])$$

$$A := \langle a|b|c|d \rangle$$

$$B := -\langle k \rangle$$

Vi kan nu se, at de eneste variable er X, Y, Z og w, med den højeste grad af 1. Da variablene er af højeste grad 1, må ligningen være lineær og dens tilsvarende ligningssystem være et lineært ligningssystem.

$$P(\text{Data}[1][1], \text{Data}[1][2], \text{Data}[1][3], \text{Data}[1][4], \text{ppos}[1], \text{ppos}[2], \text{ppos}[3])$$

Vi skal vise det er samme form som $\underline{A} \cdot \underline{x} = \underline{b}$

Hvis vi tager udgangspunkt i ligningen over, kan vi opstille udtrykket:

$$3.317639443 \cdot 10^6 + w - 0.03922802906 X + 0.7034000947 Y - 0.7097108347 Z = 0$$

$$w - 0.03922802906 X + 0.7034000947 Y - 0.7097108347 Z = -3.317639443 \cdot 10^6$$

$$A = \begin{bmatrix} -0.03922802906 & 0.7034000947 & -0.7097108347 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} -3.317639443 \cdot 10^6 \end{bmatrix}$$

Indlæs Data

restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 25 :

Table of satellite data:

Data := Matrix([[4396623.907 , -15219512.421 , 21395963.449 , 22745185],
[12508988.544 , 10190642.686 , 21073725.487 , 20490898],
[15690450.142 , -6024729.548 , 20505506.585 , 20673780],
[25180700.379 , -7388281.420 , 4051959.985 , 23193809],
[-8441248.473 , -20422306.240 , 15063403.391 , 26237952],
[-2062043.991 , 17167206.349 , 20373015.170 , 22979470]]) :

sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :
sat25 := Data[6] :

ppos := <3504300, 780800, 5252100> :

apos := <3504320.6, 780753.5, 5252128.8> :

Opgave 2

Proceduren fra opgave 1 bliver generaliseret her:

```
Linearize := proc( sat, pos, W )
    mtaylor(sqrt((x - sat[1])^2 + (y - sat[2])^2 + (z - sat[3])^2) + w - sat[4], [x=pos[1], y
        =pos[2], z=pos[3], w=W], 2)
end proc;
```

Det bliver gjort for alle satellitterne

```
Linearize(sat4, ppos, 0)
Linearize(sat14, ppos, 0) Linearize(sat16, ppos, 0)
Linearize(sat18, ppos, 0)
```

Vi kan nu anse alle ligningerne som et samlet ligningssystem og opstille koefficientmatricen A, når vi tager udgangspunkt i satellitterne 4, 14, 16 og 18:

```
A := [[ -0.03922802906530328548141814, 0.7034000947202249270459627,
        -0.7097108344130236830952448, 1 ],
       [ -0.4394081938, -0.4591787890, -0.7720591161, 1 ],
       [ -0.5893948449, 0.3291559673, -0.7377466136, 1 ],
       [ -0.9345007536, 0.3521808331, 0.05173975975, 1 ]]:
```

Og

$$b := \begin{bmatrix} -3.317639441400389179320311 \cdot 10^6 \\ -5.955142906 \cdot 10^6 \\ -5.685048905 \cdot 10^6 \\ -2.729938144 \cdot 10^6 \end{bmatrix} :$$

Nu bliver matricerne sammensat og bliver løst med Gauss-Jordan

```
answer := ReducedRowEchelonForm(⟨A|b⟩)
```

```
Data := Matrix([ [ -20000000, 10000000, 10000000, 25690538 ],
                  [ 12508988.544, 10190642.686, 21073725.487, 20490898 ],
                  [ 15690450.142, -6024729.548, 20505526.585, 20673780 ],
                  [ 25180700.379, -7388281.420, 4051959.985, 23193809 ],
                  [ -8441248.473, -20422306.240, 15063403.391, 26237952 ],
                  [ -2062043.991, 17167206.349, 20373015.170, 22979470 ]]):
```

$$\begin{bmatrix} 1. & -0. & -0. & -0. & 3.505994688610145198956435 \cdot 10^6 \\ -0. & 1. & -0. & -0. & 7.810227438326264747126625 \cdot 10^5 \\ -0. & -0. & 1. & -0. & 5.252867484526145300672420 \cdot 10^6 \\ 0. & 0. & 0. & 1. & -426.8076172138542455936285 \end{bmatrix}$$

Vi kan nu bestemme den procentmæssige forskel mellem de beregnede værdier mod de kendte positioner:

$$afvigelse_X = answer[1][5] - apos[1], PA_X = \frac{(answer[1][5] - apos[1])}{apos[1]} \cdot 100$$

$$afvigelse_Y = answer[2][5] - apos[2], PA_Y = \frac{(answer[2][5] - apos[2])}{apos[2]} \cdot 100$$

$$afvigelse_Z = answer[3][5] - apos[3], PA_Z = \frac{(answer[3][5] - apos[3])}{apos[3]} \cdot 100$$

Indlæs Data

restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 25 :

Table of satellite data:

```
Data := Matrix([ [ 4396623.907 , -15219512.421 , 21395963.449 , 22745185 ],
  [ 12508988.544 , 10190642.686 , 21073725.487 , 20490898 ],
  [ 15690450.142 , -6024729.548 , 20505506.585 , 20673780 ],
  [ 25180700.379 , -7388281.420 , 4051959.985 , 23193809 ],
  [ -8441248.473 , -20422306.240 , 15063403.391 , 26237952 ],
  [ -2062043.991 , 17167206.349 , 20373015.170 , 22979470 ]]) :
sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :
sat25 := Data[6] :
ppos := <3504300, 780800, 5252100> :
apos := <3504320.6, 780753.5, 5252128.8> :
```

Opgave 3

Vi tager udgangspunkt i koefficientmatricen og "afstands"-matricen fra sidste opgave

$$A := \begin{bmatrix} -0.03922802906 & 0.7034000947 & -0.7097108347 & 1 \\ -0.4394081938 & -0.4591787890 & -0.7720591161 & 1 \\ -0.5893948449 & 0.3291559673 & -0.7377466136 & 1 \\ -0.9345007536 & 0.3521808331 & 0.05173975975 & 1 \end{bmatrix} :$$

ReducedRowEchelonForm(Transpose(A)) ;

Og

$$b := \begin{bmatrix} -3.317639443 \cdot 10^6 \\ -5.955142906 \cdot 10^6 \\ -5.685048905 \cdot 10^6 \\ -2.729938144 \cdot 10^6 \end{bmatrix} :$$

Og de bruges i formlen på formen:

$$\underline{A} \cdot \underline{x} = \underline{b}$$

hvor

$$\underline{x} = \begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \\ \bar{w} \end{bmatrix}$$

Hvis koefficienterne i A-matricen udgør retningsvektorer og b-matricen angiver pseudoafstanden fra

sateliten til GPS-modtageren, må retningsvektorerne angive retningen fra satelliten til GPS-modtagerens foreløbige position.

Fra opgave 1, ser vi, at koefficienten for X, Y og Z er givet ved:

$$X_k = \frac{2 X_0 - 2 X_s}{2 \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$Y_k = \frac{2 Y_0 - 2 Y_s}{2 \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

$$Z_k = \frac{2 Z_0 - 2 Z_s}{2 \sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Hvis vi anser koefficienterne som koordinater for en retningsvektor, kan denne retningsvektor opstilles på formen:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 - X_s \\ Y_0 - Y_s \\ Z_0 - Z_s \end{bmatrix} \cdot \frac{1}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Det svarer til vektoren som går fra en given satellit til GPS-modtageren. Herefter bliver denne vektorer normaliseret, ved at blive divideret med længden af sig selv.

Så ikke nok med at det er en retningsvektor for retningen fra satelliten til GPS-modtageren, men det er også en enhedsvektor.

Da ligningssystemet har 4 ubekendte, skal koefficientmatricen og samtidigt totalmatrixen have en rank på 4 for, at der skal findes en entydig løsning. Altså skal A matrixens rækker/kolonner være lineært uafhængige og herved udgøre en basis for det fire-dimensionelle rum.

Hvis nu to af satellitterne der bruges til at bestemme positionen befinder sig præcis på samme sted, så vil det ikke være muligt at finde en entydig løsning og rangen af koefficientmatricen vil være lig 3 (vektorrummet de udspænder er 3 dimensionelt). Her vil der være en fri variabel og da vil løsningsmængden være en ret linje. Tilsvarende en plan hvis 3 satellitter befinder sig samme sted osv.

Procedure for getting first order approximation of equation for a satellite and position:

```
> Linearize := proc( sat, pos )
    mtaylor(sqrt((x - sat[1])^2 + (y - sat[2])^2 + (z - sat[3])^2)
    + w - sat[4], [x=pos[1], y=pos[2], z=pos[3], w], 2)
end proc;
```

Procedure for getting coefficient table for a list of satellites and a position:

```
> LinearMatrix := proc( satList, pos )
    local eqns := [];
    for local i from 1 by 1 to nops(satList) do
```

```

    eqns := [op(eqns), Linearize(satList[i], pos)]:
  end do:
  local matrix := GenerateMatrix(eqns, [x,y,z,w]):
  <matrix[1] | matrix[2]>:
end proc:
> T := LinearMatrix([sat4, sat14, sat16, sat18], ppos):
TS:=LinearSolve[ReducedRowEchelonForm](T);

```

$$TS := \begin{bmatrix} 3.504309937573080377331740 \cdot 10^6 \\ 780753.4421033402149944548 \\ 5.252120203101272900972967 \cdot 10^6 \\ -1867.702663568588856190903 \end{bmatrix} \quad (49)$$

```

> v4 := plots[arrow](<sat4[1], sat4[2], sat4[3]>, sat4[4] * <T[1]
[1], T[1][2], T[1][3]>, shape=arrow, color = red):
> v14 := plots[arrow](<sat14[1], sat14[2], sat14[3]>, sat14[4] * <T
[2][1], T[2][2], T[2][3]>, shape=arrow,color = blue):
> v16 := plots[arrow](<sat16[1], sat16[2], sat16[3]>, sat16[4] * <T
[3][1], T[3][2], T[3][3]>, shape=arrow,color = green):
> v18 := plots[arrow](<sat18[1], sat18[2], sat18[3]>, sat18[4] * <T
[4][1], T[4][2], T[4][3]>, shape=arrow,color = magenta):
> SAT:=<<sat4[1], sat4[2], sat4[3]>|<sat14[1], sat14[2], sat14[3]
>|<sat16[1], sat16[2], sat16[3]>|<sat18[1], sat18[2], sat18[3]>>:
SATPlot:=plots[pointplot3d](SAT, symbolsize=20,color=[red,blue,
green,magenta],symbol=sphere):
text4:=plots[textplot3d]([sat4[1], sat4[2], sat4[3]*1.05,
"Satellit 4"]):
text14:=plots[textplot3d]([sat14[1], sat14[2], sat14[3]*1.05,
"Satellit 14"]):
text16:=plots[textplot3d]([sat16[1], sat16[2], sat16[3]*1.05,
"Satellit 16"]):
text18:=plots[textplot3d]([sat18[1], sat18[2], sat18[3]*1.3,
"Satellit 18"]):
textGPS:=plots[textplot3d]([apos[1]*(-1.01),apos[2],apos[3]*1.5,
"GPS-modtager"]):

```

$SAT :=$ (50)

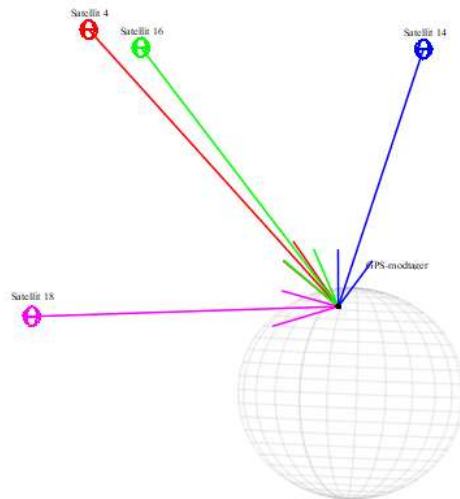
$$\begin{bmatrix} 4.396623907 \cdot 10^6 & 1.2508988544 \cdot 10^7 & 1.5690450142 \cdot 10^7 & 2.5180700379 \cdot 10^7 \\ -1.5219512421 \cdot 10^7 & 1.0190642686 \cdot 10^7 & -6.024729548 \cdot 10^6 & -7.388281420 \cdot 10^6 \\ 2.1395963449 \cdot 10^7 & 2.1073725487 \cdot 10^7 & 2.0505506585 \cdot 10^7 & 4.051959985 \cdot 10^6 \end{bmatrix}$$

```

> receiver := pointplot3d(apos, color=black,symbolsize = 2):
receiverCalculated := pointplot3d([TS[1],TS[2],TS[3]], color=
blue, symbolsize = 2):
> sphereparam := (r,u,v) -> <r * cos(u) * sin(v), r * sin(u) * sin
(v), r * cos(v)>:
> r := 6371000:
> earth := plot3d([sphereparam(r,u,v)[1], sphereparam(r,u,v)[2],
sphereparam(r,u,v)[3]], u=0..2*Pi, v=0..Pi, transparency=0.75,
color=gray, style=wireframeopaque):
> display([v4, v14, v16, v18, receiver, earth,SATPlot,text4,text14,

```

```
text16,text18,textGPS]);
```



Indlæs Data

```
restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 25 :
```

Table of satellite data:

```
Data := Matrix([ [ 4396623.907 , -15219512.421 , 21395963.449 , 22745185 ],
                  [ 12508988.544 , 10190642.686 , 21073725.487 , 20490898 ],
                  [ 15690450.142 , -6024729.548 , 20505506.585 , 20673780 ],
                  [ 25180700.379 , -7388281.420 , 4051959.985 , 23193809 ],
                  [ -8441248.473 , -20422306.240 , 15063403.391 , 26237952 ],
                  [ -2062043.991 , 17167206.349 , 20373015.170 , 22979470 ] ]):
```

```
sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :
sat25 := Data[6] :
```

```
ppos := <3504300, 780800, 5252100> :
```

```
apos := <3504320.6, 780753.5, 5252128.8> :
```

Opgave 4

Fra før fik vi designmatrixen til at være:

$$A := \begin{bmatrix} -0.03922802906 & 0.7034000947 & -0.7097108347 & 1 \\ -0.4394081938 & -0.4591787890 & -0.7720591161 & 1 \\ -0.5893948449 & 0.3291559673 & -0.7377466136 & 1 \\ -0.9345007536 & 0.3521808331 & 0.05173975975 & 1 \end{bmatrix} :$$

Hele den fjerde søjle udgøres af lutter 1-taller da koefficienten for urfejlen er lig 1 for alle satellitterne. Herved er det altså antaget, at alle satellitterne har samme urfejl:
Der undersøges nu, hvad der sker ved urfejlen i udviklingspunktet ændres for alle satellitterne:

Urfejlen sættes til k for alle satellitter

```
Linearize := proc( sat, pos, W )
  mtaylor( sqrt( (x - sat[1])^2 + (y - sat[2])^2 + (z - sat[3])^2 ) + w - sat[4], [x=pos[1], y
    =pos[2], z=pos[3], w=W], 2 )
end proc;
tay4 := Linearize(sat4, ppos, k);
tay14 := Linearize(sat14, ppos, k) tay16 := Linearize(sat16, ppos, k)
tay18 := Linearize(sat18, ppos, k)
Det ses allerede her at koefficienten for urfejlen w er den samme. Herved sker der ingen ændring, hvis
alle satellitternes urfejl ændres til den samme værdi. Tjekkes lige efter !!!

A := <coeff(tay4, x, 1), coeff(tay14, x, 1), coeff(tay16, x, 1), coeff(tay18, x, 1)|coeff(tay4, y, 1),
  coeff(tay14, y, 1), coeff(tay16, y, 1), coeff(tay18, y, 1)|coeff(tay4, z, 1), coeff(tay14, z, 1),
  coeff(tay16, z, 1), coeff(tay18, z, 1)|lcoeff(tay4), lcoeff(tay14), lcoeff(tay16), lcoeff(tay18)>;
Og
b := <tcoeff(tay4), tcoeff(tay14), tcoeff(tay16), tcoeff(tay18)>;
Nu bliver matricerne sammensat og bliver løst med Gauss-Jordan

answer := ReducedRowEchelonForm( <A|b> )
```

Urfejlen sættes til forskellige værdier for alle satellitter

```
tay4 := Linearize(sat4, ppos, k1) :
tay14 := Linearize(sat14, ppos, k2) :
tay16 := Linearize(sat16, ppos, k3) :
tay18 := Linearize(sat18, ppos, k4) :
A := <coeff(tay4, x, 1), coeff(tay14, x, 1), coeff(tay16, x, 1), coeff(tay18, x, 1)|coeff(tay4, y, 1),
  coeff(tay14, y, 1), coeff(tay16, y, 1), coeff(tay18, y, 1)|coeff(tay4, z, 1), coeff(tay14, z, 1),
  coeff(tay16, z, 1), coeff(tay18, z, 1)|lcoeff(tay4), lcoeff(tay14), lcoeff(tay16), lcoeff(tay18)>;
Og
b := <tcoeff(tay4), tcoeff(tay14), tcoeff(tay16), tcoeff(tay18)>;
Nu bliver matricerne sammensat og bliver løst med Gauss-Jordan

answer := ReducedRowEchelonForm( <A|b> )
```

Her ses det igen at det ikke gør en forskel

Urfejlen sættes til en funktion af w

Ikke den mest videnskabelige fremgangsmetode xD

```
tay4 := Linearize(sat4, ppos, w) :  
tay14 := Linearize(sat14, ppos, w) :  
tay16 := Linearize(sat16, ppos, w) :  
tay18 := Linearize(sat18, ppos, w) :  
A := <coeff(tay4, x, 1), coeff(tay14, x, 1), coeff(tay16, x, 1), coeff(tay18, x, 1)|coeff(tay4, y, 1),  
      coeff(tay14, y, 1), coeff(tay16, y, 1), coeff(tay18, y, 1)|coeff(tay4, z, 1), coeff(tay14, z, 1),  
      coeff(tay16, z, 1), coeff(tay18, z, 1)|lcoeff(tay4), lcoeff(tay14), lcoeff(tay16), lcoeff(tay18)>;  
Og  
b := <tcoeff(tay4), tcoeff(tay14), tcoeff(tay16), tcoeff(tay18)>;  
Nu bliver matricerne sammensat og bliver løst med Gauss-Jordan
```

```
answer := ReducedRowEchelonForm( <A|b> )
```

Det ses her at der altid vil findes en løsning til systemet uafhængigt af urfejlen. Det ses dog at urfejlen bliver negativ ???

Indlæs Data

```
restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 25 :
```

Table of satellite data:

```
Data := Matrix( [[ 4396623.907 , -15219512.421 , 21395963.449 , 22745185 ],  
                 [ 12508988.544 , 10190642.686 , 21073725.487 , 20490898 ],  
                 [ 15690450.142 , -6024729.548 , 20505506.585 , 20673780 ],  
                 [ 25180700.379 , -7388281.420 , 4051959.985 , 23193809 ],  
                 [ -8441248.473 , -20422306.240 , 15063403.391 , 26237952 ],  
                 [ -2062043.991 , 17167206.349 , 20373015.170 , 22979470 ] ] ) :  
sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :  
sat25 := Data[6] :  
ppos := <3504300, 780800, 5252100> :  
apos := <3504320.6, 780753.5, 5252128.8> :
```

Opgave 5

Jeg opstiller udtrykket for konstanten, som blev bestemt i Opgave 1.

$$K = \frac{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2} - P_s}{(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0} - \frac{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Vi kan se, at udtrykket er opdelt i 3 forskellige led.

Det første har vi set i de tidligere opgaver, og svarer til længden af vektoren, som går fra satelliten til den foreløbige GPS-lokation.

P_s svarer til Pseudoafstanden mellem satelliten og GPS-modtageren (Den målte afstand)

Denne afstand bliver trukket fra længden mellem satelliten og den foreløbige GPS-lokation.

Derfor kan de to første led samles til et udtryk for afvigelsen mellem afstanden til kendte punkt og det målte punkt.

For at beskrive den sidste del, vil vi nu opstille to vektorer:

$$v_1 := \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} : , \text{ som svarer til punktet for den foreløbige position af GPS-modtageren, og som vektor,}$$

vektor fra jordens tyngdecentrum til positionen.

$$v_2 := \begin{bmatrix} X_0 - X_s \\ Y_0 - Y_s \\ Z_0 - Z_s \end{bmatrix} : , \text{ som svarer til vektoren, som går fra satelliten til den foreløbige position af GPS-}$$

modtageren.

Hvis vi tager længden af projektion mellem disse to vektorer får vi:

$$\left| v_1 \cdot v_2 \right| = \frac{|v_1 \cdot v_2|}{|v_2|} = \frac{|(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0|}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}}$$

Her ser vi, at det sidste led beskriver længden af projektionen af vektoren v_1 ned på v_2 . (Nogenlunde--- diskuter absolut værdi og hvorfor den bliver negativ)

Det ses dog at for udtrykket i vores konstantled K, der tages absolutværdien ikke af prikproduktet.

Geometrisk set:

Vektorene v_1 og v_2 har samme slutpunkt nemlig (X_0, Y_0, Z_0) når man så tager projektionen, så finder man naturligvis prikproduktet.

Det vides at prikproduktet også kan skrives således:

$v_1 \cdot v_2 = |v_1| \cdot |v_2| \cdot \cos(\theta)$, hvor θ er vinklen mellem de to vektorer. Hvis denne vinkel ligger i intervallet

$\left[\frac{\pi}{2}, \frac{3\pi}{2} \right]$, altså hvis satellitten foran tangentplanen til punktet (tangentplanens retningsvektor er her v_1 normalvektor, så \hat{v}_1), så vil dette prikprodukt naturligvis være negativt og da vil længden af

projektionen få et negativt fortegn. Tilsvarende hvis satellitten befinder sig bag ved tangentplanen, så vil prikproduktet være positivt.

Derfor er værdierne i b-matricen værdien af forskellen mellem pseudoafstanden og afstanden til den foreløbige GPS-lokation lagt sammen med længden af projektionen af vektoren fra v_1 på v_2 .

$$L(X_s, Y_s, Z_s, X_0, Y_0, Z_0) := - \frac{(X_0 - X_s) X_0 + (Y_0 - Y_s) Y_0 + (Z_0 - Z_s) Z_0}{\sqrt{(X_0 - X_s)^2 + (Y_0 - Y_s)^2 + (Z_0 - Z_s)^2}} :$$

$L(Data[1][1], Data[1][2], Data[1][3], ppos[1], ppos[2], ppos[3])$
3.317639441400389179320311

Indlæs Data

restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 25 :

Table of satellite data:

```
Data := Matrix([ [ 4396623.907 , -15219512.421 , 21395963.449 , 22745185 ],
  [ 12508988.544 , 10190642.686 , 21073725.487 , 20490898 ],
  [ 15690450.142 , -6024729.548 , 20505506.585 , 20673780 ],
  [ 25180700.379 , -7388281.420 , 4051959.985 , 23193809 ],
  [ -8441248.473 , -20422306.240 , 15063403.391 , 26237952 ],
  [ -2062043.991 , 17167206.349 , 20373015.170 , 22979470 ]]) :
sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :
sat25 := Data[6] :
ppos := <3504300, 780800, 5252100> :
apos := <3504320.6, 780753.5, 5252128.8> :
```

Opgave 8

Generel metode

Vi har det lineariserede system, hvor A er en $n \times m$ matrix og $\delta(A) = r$

$$Ax = b$$

Det vides at hvis A er en kvadratisk, $m = n$ og invertibel ($n=m=r$, nonsingulær/ $\det(A) \neq 0$) så findes der en eksakt løsning til systemet. Dvs:

$$b - Ax = 0$$

For ligningsystemet (5) er dette dog ikke tilfældet, da vi har flere rækker end vi har variable, altså $n > m$, $\delta(A) = 4 = m$. Herved er A ikke kvadratisk og da singulær (ikke invertibel):

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & & \cdot \\ a_{n,1} & \cdot & \cdot & a_{n,4} \end{bmatrix}$$

Derfor findes der ikke nødvendigvis en løsning til ligningsystemet med mindre $b \in C(A)$ (b indgår i billedet af A/A 's billedmængde).

Hvis $b \in A^\perp$, altså hvis b står vinkelret på A så findes der fx ikke en løsning idet $b \in \ker(A^T)$:

$$A^T Ax = A^T b \Rightarrow A^T Ax = 0$$

(Billedet af b ligger i kernen for A^T hvilket vil sige vi umuligt kan finde ud af, hvilken en vektor b var i definitionsrummet/hvilke en vektor x , b er et billede af).

Geometrisk set vil en system uden en løsning betyde at den højreside/vektor b vi gerne vil have fat i ikke ligger i billedrummet for $A(C(A))$.

I stedet kan man dog finde en løsning

$$x_0 = \begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix}, \text{ hvor residualerne } R, \text{ også kaldet fejlen man begår ved}$$

valget af løsningen er mindst muligt:

$$R = b - Ax$$

Her minimeres der med hensyn til de kvadrede residualer bl.a. for at behandle positive og negative fejl ens og for at forsikre, at udtrykkene for residualerne altid er differentierbare.

Dette betyder i praksis at vi skal minimere følgende, for at finde den bedste mulige løsning til systemet, her kaldet x_0 :

$$\begin{aligned} \phi(x) &= \sum_{i=1}^n (R_i^2) = \|Ax - b\|^2 = (Ax - b)^T (Ax - b) \Rightarrow \\ \phi(x) &= b^T b - A^T x^T b - b^T A x + A^T x^T A x \end{aligned}$$

Ved minimering af dette udtryk fås et system af lineære ligninger, hvortil der findes en eksakt løsning.

Minimeringen:

Det vides at et udtryk antager den mindst eller størst mulige værdi det sted, hvor dens gradient er lig 0. Herved skal x_0 opfylde følgende:

$$\nabla \phi(x_0) = 0 \quad (1)$$

Idet der er tale om et udtryk bestående af summer, så kan det med fordel opdeles ved udregning af gradienten:

$$\nabla (x^T A^T A x) = \frac{\partial}{\partial x} (x^T A^T A x) = A^T A \frac{\partial}{\partial x} (x^T x) = A^T A 2x = 2 A^T A x$$

$$\nabla (A^T x^T b) = \frac{\partial}{\partial x} (A^T x^T b) = \left(\frac{\partial}{\partial x} (b^T x A) \right)^T = (b^T A)^T = A^T b$$

$$\nabla (b^T A x) = \frac{\partial}{\partial x} (b^T A x) = \left(\frac{\partial}{\partial x} (x^T A^T b) \right)^T = (A^T b)^T = A b^T = b A^T$$

$$\nabla (b^T b) = 0$$

Herved fås gradienten til:

$$\nabla \phi(x) = 2 A^T A x - 2 A^T b \quad (2)$$

Og i x_0 :

$$\nabla \phi(x_0) = 2 \cdot A^T A x_0 - 2 A^T b \Rightarrow$$

$$\nabla \phi(x_0) = 2 [x_0^T A^T A - b^T A]$$

(Det første er en $1 \times n$ matrix mens den anden er en $n \times 1$ matrix).

Dette udtryk sættes nu lig 0, da der her må være tale om et ekstremum

$$\nabla \phi(x) = 2 \cdot A^T A x - 2 A^T b = 0 \Rightarrow$$

$$2 \cdot A^T A x = 2 A^T b \Rightarrow$$

$$A^T A x = A^T b$$

Residualerne skal minimeres og lige nu ved vi ikke om løsningen er et maksimum, lokalt minimum eller et saddepunkt mm. Dette kan dog bedømmes ved at finde den anden afledede af funktionen $\nabla \phi(x)$, også kalde Hessematrixen. Idet den vil beskrive et globalt minimum, hvis den er positivt defineret. (referer til undervisnings note):

$$H(\phi(x)) = \frac{d}{dx} (\nabla \phi(x)) = 2 A^T A$$

Lad os nu undersøge matrixen $A^T A$.

Det ses at der er tale om en kvadratisk 4x4 og symmetrisk matrix idet det vides at (jævnfør undervisningen):

$$(A^T)^T = A \quad \wedge \quad (AB)^T = B^T A^T$$

Hvilket vil sige at der i vores tilfælde gælder:

$$(A^T A)^T = A^T (A^T)^T = A^T A$$

Hvilket kun gør sig gældende for symmetriske matricer (jævnfør undervisningen)

Fra før vides det at $\delta(A) = 4$ og herved må A have 4 lineært uafhængige søjler/vektorer, så $m = 4$ (står også øverst). Herved findes der kun en løsning, nemlig nul-løsningen på:

$$Ax = 0 \Rightarrow$$

$$a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = 0$$

$$a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = 0$$

.

.

.

$$a_{m,1}x_1 + a_{m,2}x_2 + \dots + a_{m,n}x_n = 0$$

\Leftrightarrow

$$x = 0$$

Udtrykket er da uegentlig og $\ker(A) = \{0\}$

Derudover ses det at A og $A^T A$ har den samme kerne da:

$$A^T A x = 0 \Rightarrow$$

$$x^T A^T A x = 0 \Rightarrow$$

$$(Ax)^T Ax = 0 \Rightarrow$$

$$\|Ax\|^2 = 0 \Rightarrow$$

$$Ax = 0$$

Herved har matrixen $A^T A$ således også kun en løsning på $A^T A x = 0$. Nemlig $x = 0$. Vi ved da at 0 ikke er en egenverdi for matrixen idet kernen kun består af nul-vektoren.

Det ses samtidigt, at $A^T A$ må have positive egenverdier:

$$A^T A x = \lambda x \Rightarrow$$

$$A^T A x \cdot x = \lambda x \cdot x \Rightarrow$$

$$|Ax|^2 = \lambda |x|^2 \geq 0$$

Så for at summere så er $A^T A$ en 4x4, symmetrisk, nonsingulær matrix med positive egenverdier. Herved er den hvad man kalder positivt defineret og da vil løsningen på (2) være et globalt minimum. I forlængelse af dette så ses det at ligningsystemet rent faktisk også har en eksakt løsning. Kort opsummeret her:

Da $A^T A$ er nonsingulær (invertibel) dvs:

$$A^T A B = B A^T A = I$$

Så har $A^T A$ 4 ledende et-taller på reduceret form, altså $\delta(A^T A) = 4$, hvilket betyder den har fuld rang. Dette betyder for vores ligningsystem, at der findes en eksakt løsning x_0 .

Geometrisk tolkning

Som nævnt vil vi gerne finde den løsning, der gør residualerne mindst mulig:
 $R = b - Ax$

Vi ved at der ikke findes en eksakt løsning, altså en løsning, hvor:
 $b - Ax = 0$

Derfor vil vi prøve at finde en "mindste kvadraters løsning", som gør denne foreskel mindst mulig. Den vektor der ligger tættest på b må være projektionen af b på $C(A)$ (det vektorrum A udspænder/columnspace of A):

$$Ax_0 = \text{proj}_{C(A)} b$$

Herved fås:

$$Ax_0 - b = \text{proj}_{C(A)} b - b$$

Vektoren der går fra b's endepunkt til projektionen af b's endepunkt kaldes nu for v: $v = \text{proj}_{C(A)} b - b$

Længden af denne vektor svare naturligvis til residualerne.

Det vides at denne vektor v må stå vinkelret på $C(A)$ og derfor er den en del af A's ortogonale komplement A^\perp :

$$Ax_0 - b \in A^\perp$$

Det vides desuden at A's ortogonalkomplement udgør kernen for A^T :

$$C(A)^\perp = N(A^T) \Rightarrow Ax_0 - b \in N(A^T)$$

Ved at gange A's transponerede på vores udtryk for R så fås:

$$A^T R = A^T (b - Ax_0) = 0$$

Altså findes der en eksakt løsning til dette udtryk og samme udtryk som vi fandt før fås:

$$A^T Ax_0 - A^T b = 0 \Rightarrow A^T Ax_0 = A^T b$$

Dette udtryk vil altid have en løsning da projektionen af b på $C(A)$ naturligvis altid må ligge i $\text{span}(A)$.

Matrixen $A^T A$ er symmetrisk idet $(A^T A) = A^{TT} A^T = A A^T$ (En symmetrisk matrix er altid lig dens transponerede).

Indlæs Data

restart : with(LinearAlgebra) : with(plots) : with(plottools) : Digits := 30 :

Table of satellite data:

```
Data := Matrix([ [ 4396623.907 , -15219512.421 , 21395963.449 , 22745185 ],
    [ 12508988.544 , 10190642.686 , 21073725.487 , 20490898 ],
    [ 15690450.142 , -6024729.548 , 20505506.585 , 20673780 ],
    [ 25180700.379 , -7388281.420 , 4051959.985 , 23193809 ],
    [ -8441248.473 , -20422306.240 , 15063403.391 , 26237952 ],
    [ -2062043.991 , 17167206.349 , 20373015.170 , 22979470 ]]) :
sat4 := Data[1] : sat14 := Data[2] : sat16 := Data[3] : sat18 := Data[4] : sat24 := Data[5] :
sat25 := Data[6] :
ppos := <3504300, 780800, 5252100> :
apos := <3504320.6, 780753.5, 5252128.8> :
```

Opgave 9

Vi opstiller A matricen med de 6-Observationer

Først bliver den koefficientmatricen A bestemt:

Proceduren fra opgave 1 bliver generaliseret her:

```
Linearize := proc( sat, pos, W )
    mtaylor( sqrt( (x - sat[1])^2 + (y - sat[2])^2 + (z - sat[3])^2 ) + w - sat[4], [x=pos[1], y
        = pos[2], z=pos[3], w=W], 2 )
end proc :
```

Det bliver gjort for alle satellitterne

Linearize(sat4, ppos, 0)

$$3.31763944140038917932031174838 \cdot 10^6 - 0.0392280290653032854814181312247 x + 0.703400094720224927045962794910 y - 0.709710834413023683095244625521 z + w \quad (51)$$

Linearize(sat14, ppos, 0)

$$5.95514290568067707894525788721 \cdot 10^6 - 0.439408193889949892143158551645 x - 0.459178788832056563964350410091 y - 0.772059116278732770203084371485 z + w \quad (52)$$

Linearize(sat16, ppos, 0)

$$5.68503313078117413398460862045 \cdot 10^6 - 0.589395265768463767757931528680 x + 0.329156201909414575309440884598 y - 0.737746172768311057007551471999 z + w \quad (53)$$

Linearize(sat18, ppos, 0)

$$2.72993814435100502972097177373 \cdot 10^6 - 0.934500753415284901685898899515 x + 0.352180832989992329499442273428 y + 0.0517397597669337330677927791449 z \quad (54)$$

$$\begin{aligned}
& + w \\
& \text{Linearize}(\text{sat24}, \text{ppos}, 0) \\
& -260539.89526255618620541816392 + 0.455244610982836133001594661277 x \\
& + 0.808049950462625806344051800033 y - 0.373908574023696567232193651370 z + w
\end{aligned} \tag{55}$$

$$\begin{aligned}
& \text{Linearize}(\text{sat25}, \text{ppos}, 0) \\
& 3.16549386410151839237366732686 \cdot 10^6 + 0.242211787617919719572311238474 x \\
& - 0.713031889664420000983015144517 y - 0.657965785022685000554631895337 z + w
\end{aligned} \tag{56}$$

$$\begin{aligned}
l_1 &:= \langle \text{coeff}((\mathbf{51}), x, 1) | \text{coeff}((\mathbf{51}), y, 1) | \text{coeff}((\mathbf{51}), z, 1) | \text{coeff}((\mathbf{51}), w, 1) \rangle : \\
l_2 &:= \langle \text{coeff}((\mathbf{52}), x, 1) | \text{coeff}((\mathbf{52}), y, 1) | \text{coeff}((\mathbf{52}), z, 1) | \text{coeff}((\mathbf{52}), w, 1) \rangle : \\
l_3 &:= \langle \text{coeff}((\mathbf{53}), x, 1) | \text{coeff}((\mathbf{53}), y, 1) | \text{coeff}((\mathbf{53}), z, 1) | \text{coeff}((\mathbf{53}), w, 1) \rangle : \\
l_4 &:= \langle \text{coeff}((\mathbf{54}), x, 1) | \text{coeff}((\mathbf{54}), y, 1) | \text{coeff}((\mathbf{54}), z, 1) | \text{coeff}((\mathbf{54}), w, 1) \rangle : \\
l_5 &:= \langle \text{coeff}((\mathbf{55}), x, 1) | \text{coeff}((\mathbf{55}), y, 1) | \text{coeff}((\mathbf{55}), z, 1) | \text{coeff}((\mathbf{55}), w, 1) \rangle : \\
l_6 &:= \langle \text{coeff}((\mathbf{56}), x, 1) | \text{coeff}((\mathbf{56}), y, 1) | \text{coeff}((\mathbf{56}), z, 1) | \text{coeff}((\mathbf{56}), w, 1) \rangle :
\end{aligned}$$

$$\begin{aligned}
A &:= \langle l_1; l_2; l_3; l_4; l_5; l_6 \rangle \\
A &:= [[-0.0392280290653032854814181312247, 0.703400094720224927045962794910, \\
& -0.709710834413023683095244625521, 1], \\
& [-0.439408193889949892143158551645, -0.459178788832056563964350410091, \\
& -0.772059116278732770203084371485, 1], \\
& [-0.589395265768463767757931528680, 0.329156201909414575309440884598, \\
& -0.737746172768311057007551471999, 1], \\
& [-0.934500753415284901685898899515, 0.352180832989992329499442273428, \\
& 0.0517397597669337330677927791449, 1], \\
& [0.455244610982836133001594661277, 0.808049950462625806344051800033, \\
& -0.373908574023696567232193651370, 1], \\
& [0.242211787617919719572311238474, -0.713031889664420000983015144517, \\
& -0.657965785022685000554631895337, 1]]
\end{aligned} \tag{57}$$

$$\begin{aligned}
b &:= \langle (-1) \cdot 3.31763944140038917932031174838 \cdot 10^6; (-1) \\
& \cdot 5.95514290568067707894525788721 \cdot 10^6; (-1) \cdot 5.68503313078117413398460862045 \cdot 10^6; (-1) \\
& \cdot 2.72993814435100502972097177373 \cdot 10^6; (-1) \cdot (-260539.89526255618620541816392); (-1) \\
& \cdot 3.16549386410151839237366732686 \cdot 10^6 \rangle :
\end{aligned}$$

$$x_1 := \begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix} :$$

$$(\text{Transpose}(A) \cdot A)^{-1} \cdot \text{Transpose}(A) \cdot b$$

$$\begin{bmatrix} 3504320.55238 \\ 780753.48391 \\ 5252128.77071 \\ -1857.40902 \end{bmatrix} \quad (58)$$

$$\text{Transpose}(A) \cdot A \cdot x_1 = \text{Transpose}(A) \cdot b$$

$$\langle \text{Transpose}(A) \cdot A | (\text{Transpose}(A) \cdot b) \rangle$$

$$\text{answer} := \text{ReducedRowEchelonForm}(\text{??})$$

Vi kan nu bestemme den procentmæssige forskel mellem de beregnede værdier mod de kendte positioner:

$$\text{afvigelse}_X = \text{answer}[1][5] - \text{apos}[1], PA_X = \frac{(\text{answer}[1][5] - \text{apos}[1])}{\text{apos}[1]} \cdot 100$$

$$\text{afvigelse}_Y = \text{answer}[2][5] - \text{apos}[2], PA_Y = \frac{(\text{answer}[2][5] - \text{apos}[2])}{\text{apos}[2]} \cdot 100$$

$$\text{afvigelse}_Z = \text{answer}[3][5] - \text{apos}[3], PA_Z = \frac{(\text{answer}[3][5] - \text{apos}[3])}{\text{apos}[3]} \cdot 100$$