

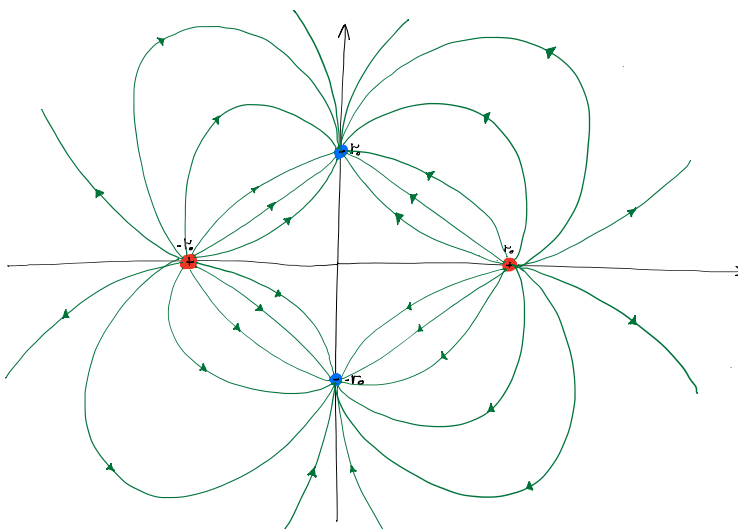
Prosjekt 2 - Introduksjon til Vitenskapelige Beregninger

Studentnr: 755110, 759144 og 753717

Mars 2016

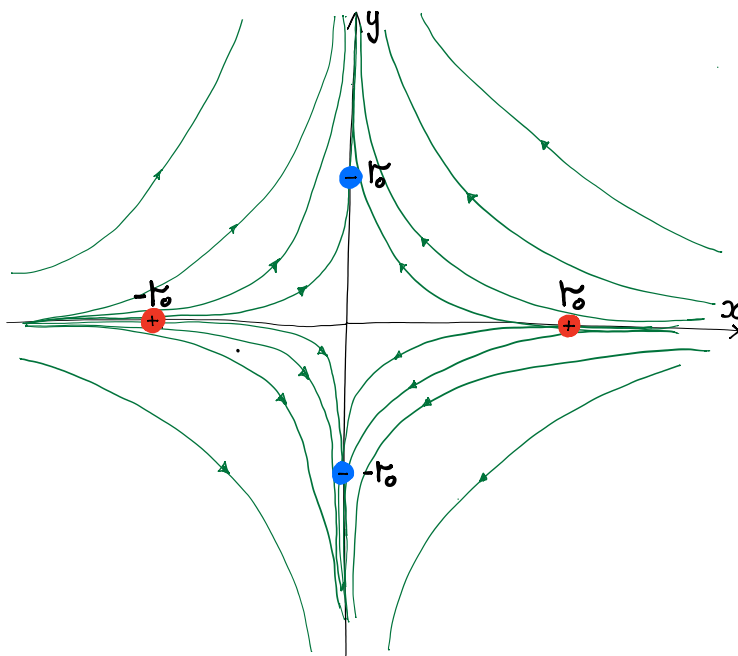
1 Oppgave 1

Feltlinjene for en kvadrupol med positive punktladninger Q langs x-aksen i $x = \pm r_0$ og negative punktladninger $-Q$ langs y-aksen i $y = \pm r_0$ er skissert kvalitativt i figur 1. I figur 2 er tilsvarende feltlinjer plottet for det hyperbolske potensialet.



Figur 1: Elektriske feltlinjer for 4 punktladninger (kvadrupol).

Kvalitativt kan man observere at feltlinjene har kun lik krumming på innsiden av kvadratet med de fire punktladningene som hjørner. Mer spesifikt, vil feltlinjene være mest identiske i området, innad dette kvadratet, som er nærmest origo.



Figur 2: Elektriske feltlinjer for hyperbolsk potensialfelt.

2 Oppgave 2

I funksjonen `electricFieldPlot()` beregnes potensialet og det tilhørende elektriske feltet både ved superposisjon av linjeladninger og med en hyperbolsk tilnærming (se under). Disse feltene tegnes i et kvadratisk område gitt ved $[-2r_0, 2r_0] \times [-2r_0, 2r_0]$. Potensialfeltet, gitt ved superposisjonen av de individuelle potensialfeltene til de fire linjeladningene, er gitt proporsjonalt ved

$$V(\vec{r}) \propto \ln |\vec{r} - \vec{r}_1| - \ln |\vec{r} - \vec{r}_2| + \ln |\vec{r} - \vec{r}_3| - \ln |\vec{r} - \vec{r}_4|, \quad (1)$$

hvor \vec{r}_1 og \vec{r}_3 er posisjonen til de positive linjeladningene, og \vec{r}_2 og \vec{r}_4 er posisjonen til de negative linjeladningene. Det elektriske feltet beregnes så numerisk ved gradienten til potensialfeltet

$$\vec{E} = -\vec{\nabla}V. \quad (2)$$

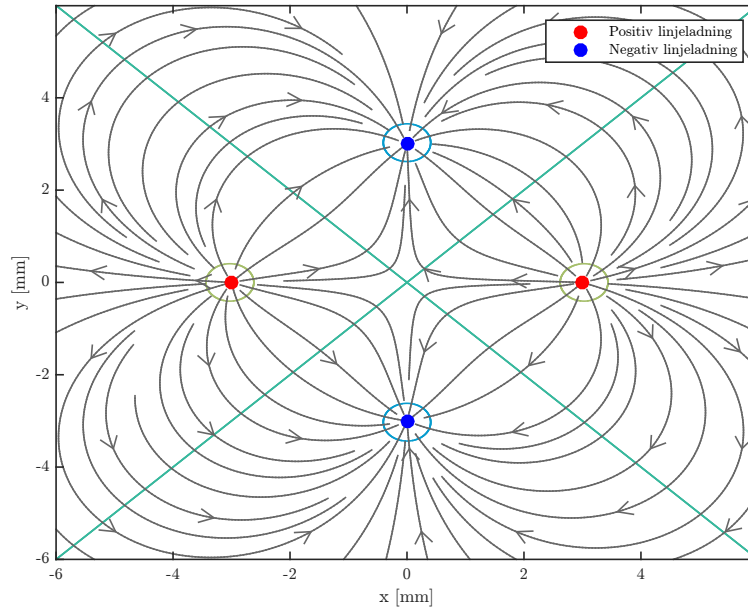
Det hyperbolske modellpotensialet er gitt ved

$$V(x, y, t) = (V_{DC} + V_{AC} \cos \omega t) \frac{x^2 - y^2}{r_0^2}, \quad (3)$$

og det tilsvarende elektriske feltet er derfor

$$\vec{E} = \frac{2}{r_0^2}(V_{DC} + V_{AC} \cos \omega t)(-x\hat{i} + y\hat{j}). \quad (4)$$

Feltlinjer og ekvipotensiallinjer er plottet i figur 3 og 4 for henholdsvis linjeladningene og den hyperbolske tilnærmingen.



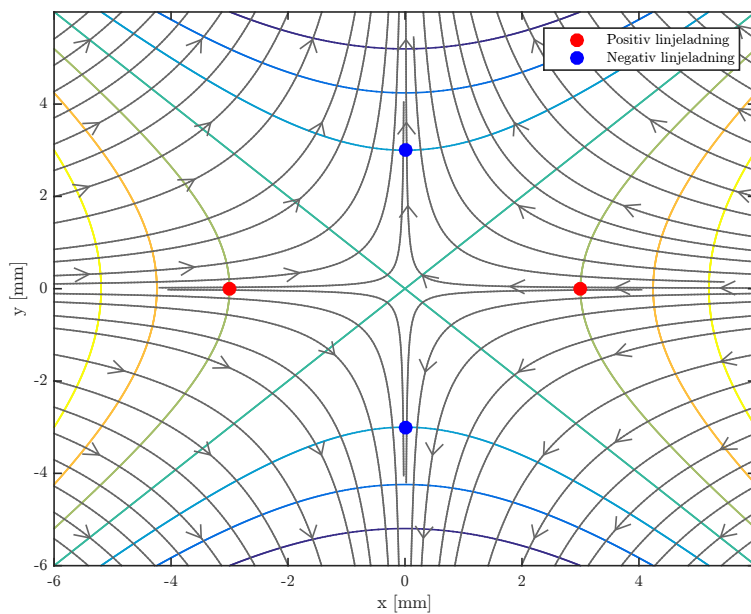
Figur 3: Elektriske feltlinjer for 4 punktladninger (kvadrupol.)

Disse figurene har kvalitativt samme form som i oppgave 1, og konklusjonene trukket i oppgave 1 er derfor fortsatt gyldige. Ekvipotensiallinjene for både linjeladningene og det hyperbolske potensialet er plottet sammen i figur 5 for et innskrenket område rundt origo innenfor de fire punktladningene.

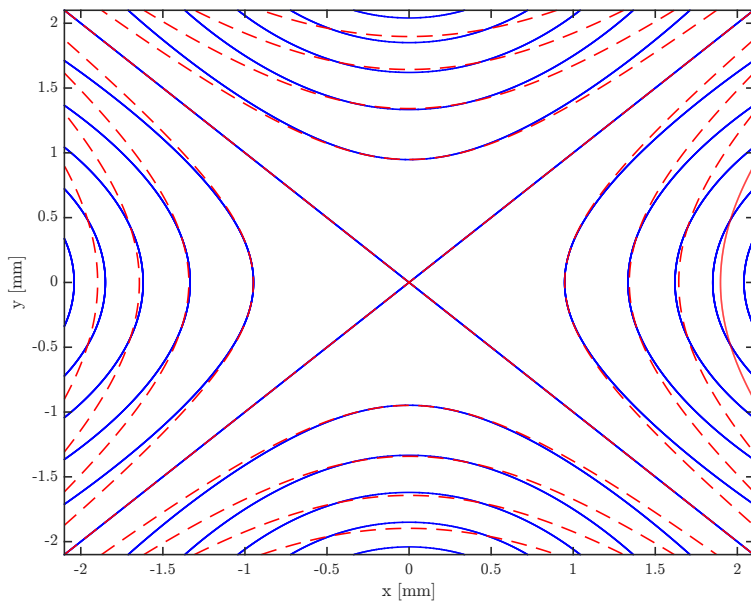
Figur 5 viser at nøyaktigheten til den hyperbolske tilnærmingen avtar lenger unna origo, og nøyaktigheten er spesielt god langs aksene. Til vårt formål er derfor denne tilnærmingen tilstrekkelig, da partiklene vil hovedsakelig befinne seg innenfor et område relativt nærme origo.

3 Oppgave 3

Newtons 2. lov gir den påførte akselerasjonen for en partikkel med masse m og ladning q i et elektrisk felt E som $\mathbf{a} = q\mathbf{E}/m$. Ved å bruke at posisjonsvektor $\ddot{\mathbf{r}} = \dot{\mathbf{v}} = \mathbf{a}$, kan vi sette opp de to følgende lineære differensialligninger av første grad



Figur 4: Elektriske feltlinjer for hyperbolsk potensialfelt.

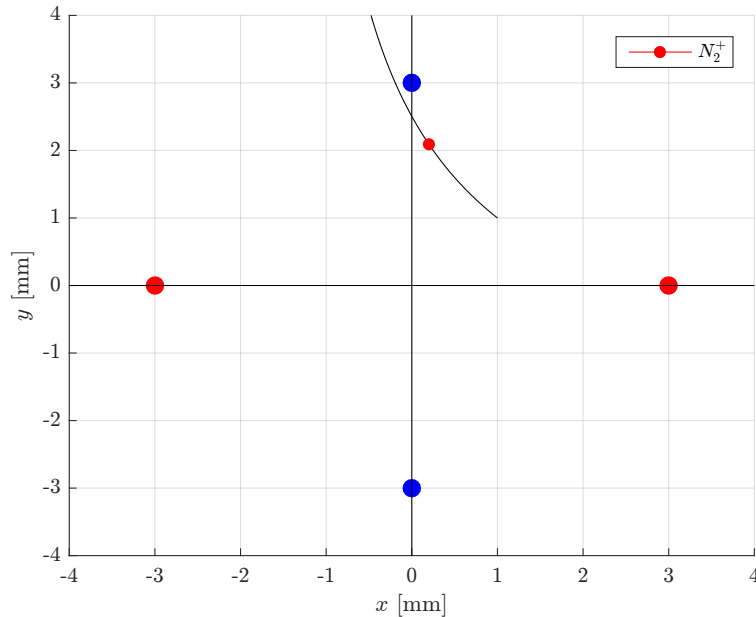


Figur 5: Ekvipotensiallinjer for linjeladninger, gitt i heltrukket blått, og hyperbolsk potensial, gitt i stiplet rødt.

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= q\mathbf{E}/m.\end{aligned}\tag{5}$$

Disse kan løses ved bruk av Eulers metode, noe som gjøres i funksjonen `eulerStep()`. Funksjonen tar i bruk `wdot()` som implementerer ligning (4) og (5) med gitte parametre. Ved å kalle `plotTrajectory2D()` vil N_2^+ -partikkels bane plottes og animeres (med masse $m = 8u$ og ladning $q = e$) i xy -planet mellom elektrodene.

Figur 6 viser et stillbilde av animasjonen for initialverdiene $x_0 = y_0 = 1$ mm. Figur 7 plottes tilsvarende for $y_0 = 0$ mm. Her er $V_{DC} = \pm 5$ V og $V_{AC} = 0$ V, steglendgden $\Delta t = 10^{-9}$ s, og starthastigheten satt til null.

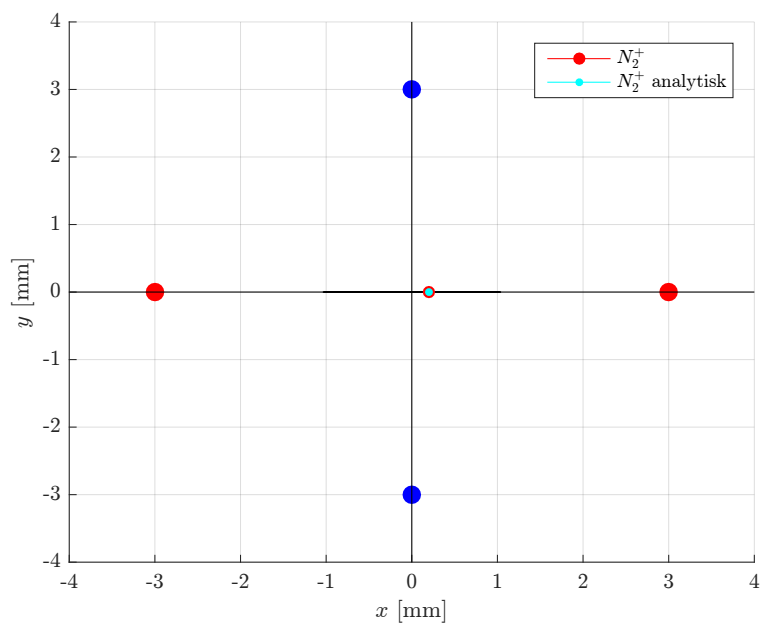


Figur 6: Animasjon av bevegelsen til en N_2^+ -partikkel i området mellom elektrodene (rød markerer positiv elektrode, blå markerer negativ elektrode). Partikkelen beveger seg mot positiv y -retning.

Banen vist i figur 6 går godt overens med feltlinjediagrammet i oppgave 2. Partikkelen oppnår tilstrekkelig hastighet til å krysse y -aksen, mens banen viser en konstant avbøyning mot nærmeste negative elektrode, noe som er å forvente ut fra både feltlinjediagrammet. Akselerasjonen til partikkelen virker derfor å alltid å være rettet langs \vec{E} -feltet, noe som følger teorien fra fysikken.

For $y_0 = 0$ mm eksisterer det en analytisk løsning i formen

$$x(t) = x_0 \cos \sqrt{at}, \text{ der } a = \frac{2qV_{DC}}{mr_0^2}.\tag{6}$$

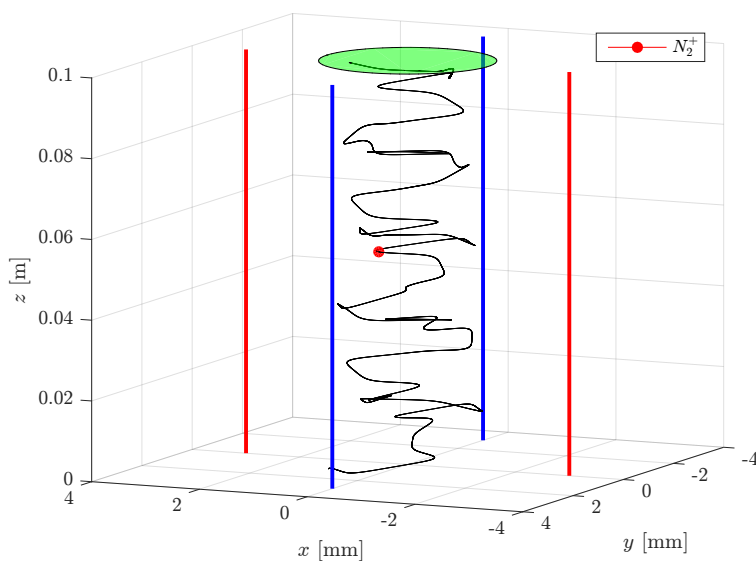


Figur 7: Animasjon av bevegelsen til en N_2^+ -partikkel i området mellom elektrodene (rød markerer positiv elektrode, blå markerer negativ elektrode). Både numerisk og analytisk løsning er animert. Partikkelen beveger seg her mot negativ x-retning.

Denne animeres, sammen med en numerisk løsning, av `plotTrajectory2D()`. Visuelt ser det ut som steglengde Δt på 1 ns gir god nøyaktighet de første 20 μs . Når Δt nærmer seg 10 ns avviker derimot den numeriske løsningen tydelig fra den analytiske. Dette kommer vi tilbake til i oppgave 5.

4 Oppgave 4

Funksjonen `plotTrajectory3D()` gjør tilsvarende som `plotTrajectory2D()` (se oppgave 3) for tre dimensjoner. Figur 8 viser et stillbilde av en animasjon med $V_{AC} = 45 \text{ V}$, $\omega = 10^7 \text{ s}^{-1}$, $v_{z0} = 5000 \text{ m s}^{-1}$, $y_0 = 1 \text{ mm}$ og resten av parametrene som i oppgave 3. Elektrodene lengde er 10 cm.



Figur 8: Animasjon av bevegelsen til en N_2^+ -partikkel i området mellom elektrodene (rød markerer positiv og blå negativ elektrode). Partikkelen beveger seg her oppover.

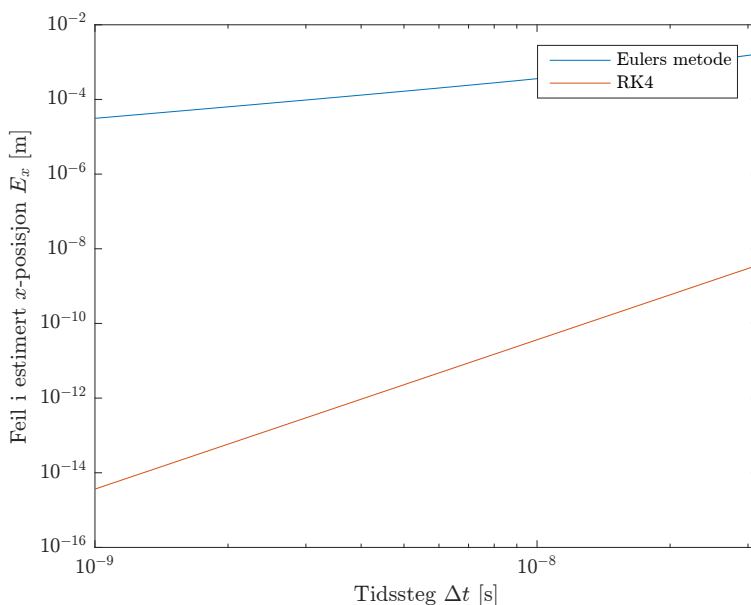
Partikkelbanen i figur 8 viser at partikkelen ikke går tapt underveis (den er ikke lenger fra z -aksen enn 3 mm), og den registreres av detektoren med radius $r_0/2$ ved elektrodene øvre ende. Dette gjelder også for $y_0 = 0 \text{ mm}$. Denne slutningen, basert på visuell observasjon av figur 8, bekreftes også numerisk. Funksjonen `getTrajectoryStability()` sjekker steg for steg at partikkelen holder seg innenfor det aksepterte området.

5 Oppgave 5

Runge-Kutta metoden av fjerde orden (RK4) er implementert i funksjonen `rk4Step()`. Som `eulerStep()` (se oppgave 3), benytter den seg av `wdot()`

for de problem-spesifikke ligningene og konstantene.

`plotMaxError()` generer et log-log plot av den maksimale feilen over fem perioder i den numeriske løsningen i forhold til den analytiske løsningen, for $y_0 = 0$ mm, som funksjon av tidssteget Δt . Resultatet av kjøring med Δt -verdier mellom 10^{-9} s og $10^{-7.5}$ s er vist i figur 9. Stigningstallene for feilen med bruk av hhv. Eulers metode og RK4 er henholdsvis tilnærmet lik 1 og 4, slik som forventet av første- og fjerde ordens metoder for iterativ løsning av differensialligninger på denne formen.



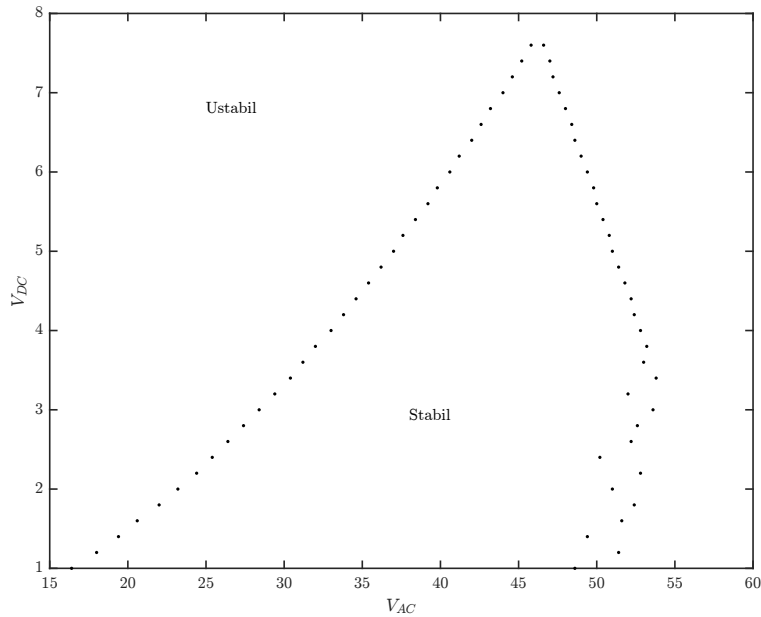
Figur 9: Feil i estimert x -posisjon som funksjon av tidssteg Δt .

Med $\Delta t = 10^{-7.5}$ s er den maksimale relative numeriske feilen ved bruk av RK4 lik $E_{x,rel} \approx 0.0035\%$. Dette er godt under kravet om maksimal relativ numerisk feil på 0.01% . Ved å bruke denne steglengden, og `rk4Step()` i stedet for `eulerStep` under kjøring av `plotTrajectory2D()` og `plotTrajectory3D()`, får vi resultater som er svært like som de generert ved bruk av Eulers metode. Resultatene gir altså fysisk mening.

Det er dog ingen måte å regne ut den eksakte feilen når en analytisk løsning ikke er tilgjengelig, og vi velger derfor $\Delta t = 10^{-8}$ s som steglengde ved bruk av RK4 i senere oppgaver. Med en slik steglengde vil også kjøretidene til funksjonene være rimelige. Denne steglengden er derfor et kompromiss mellom tilstrekkelig nøyaktig og relativt hurtig kjøretid.

6 Oppgave 6

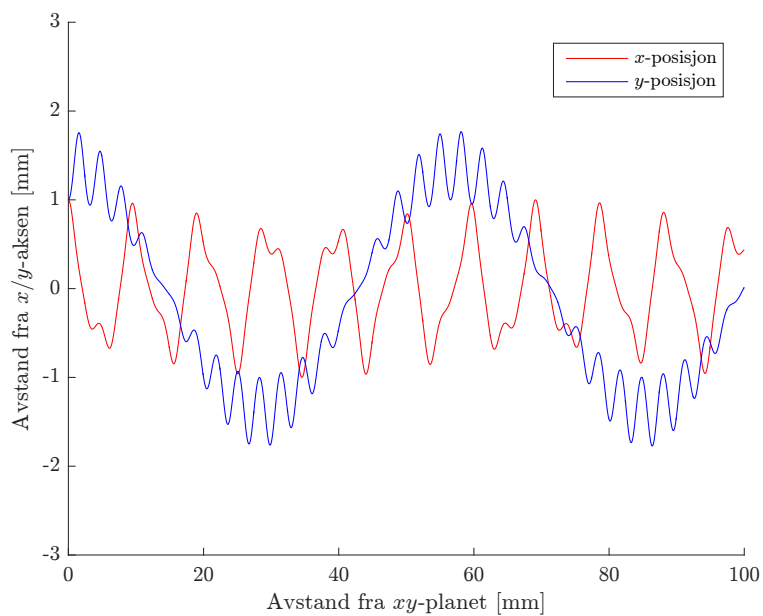
Funksjonen `plotStability()` bruker `getTrajectoryStability()` til å tegne en kurve som skiller de kombinasjonene av V_{DC} og V_{AC} som gir stabile baner fra de kombinasjonene som gir ustabile baner. Figur 10 viser resultatet for N_2^+ med steglengder for V_{DC} og V_{AC} lik 0.2 V. Den optimale kombinasjonen viser seg å være $V_{DC} = 7.7$ V og $V_{AC} = 46.25$ V.



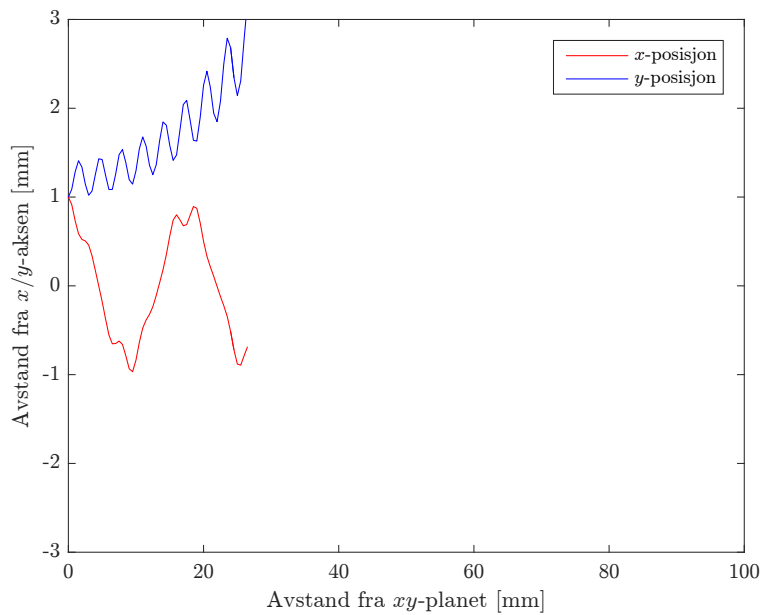
Figur 10: Stabilitets-diagram - plot av grensen mellom stabile og ustabile kombinasjoner av V_{DC} og V_{AC} .

Eksempler på stabil og ustabil bane, fra funksjonene `plotTrajectory_stable()` og `plotTrajectory_unstable()`, er gitt i figuren 11 og 12, med V_{DC} og V_{AC} lik henholdsvis 5 V og 38 V, og 2 V og 22.5 V.

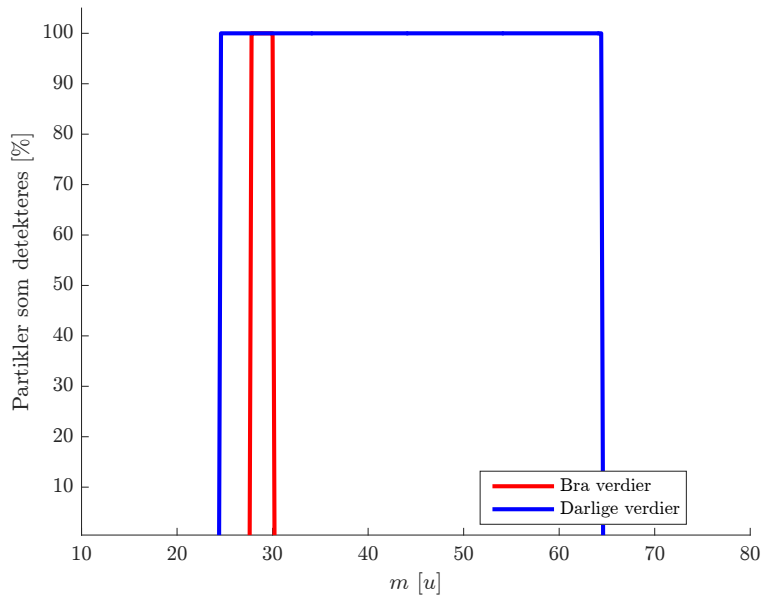
Ved å variere massen til partikkelen som sendes inn i massespektrometeret kan en kurve med hvilke masser som treffer detektoren plottes ved hjelp av funksjonen `testingParticleMasses()`. I figur 13 er det brukt V_{DC} lik 7.25 V for "bra" verdier og 3.5 V for "dårlige" verdier, og V_{AC} lik 46.25 V for begge. Dette stemmer godt overens med Figur 4 i oppgaveteksten. Endringer i spenningsverdiene påvirker masseoppløsningen Δm i stor grad; en liten endring i spenning gir stor endring i Δm .



Figur 11: Plot av partikkel med stabil bane.



Figur 12: Plot av partikkel med ustabil bane.



Figur 13: Partikkel-massene som når detektoren med ”gode” og ”dårlige” verdier av V_{DC} og V_{AC} .

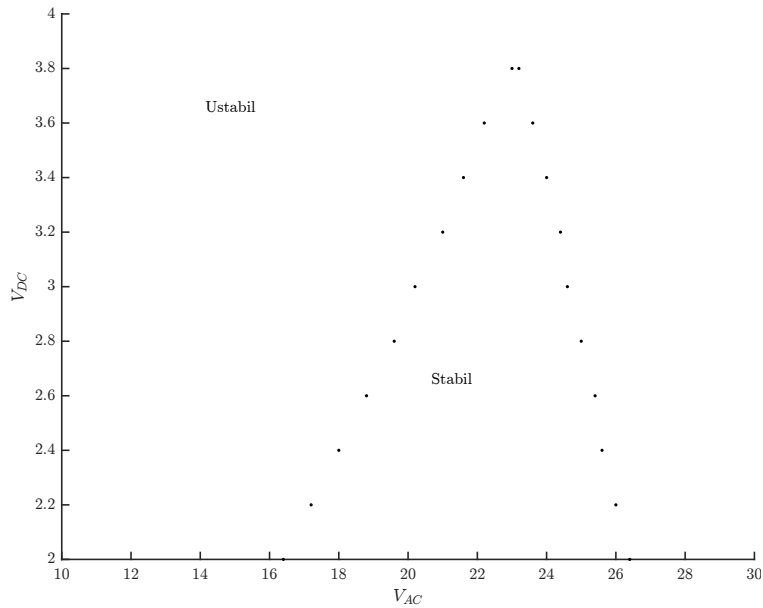
7 Oppgave 7

Massen til en N^+ -partikkel er halvparten av massen til en N_2^+ -partikkel. Siden ladningene til disse partiklene er like, vil den påførte kraften fra det elektriske feltet være lik. Når massen halveres, vil akselerasjonen derfor dobles (Newtons andre lov). For at banen til N^+ skal være lik den til N_2^+ , må V_{DC} og V_{AC} halveres. Derfor vil optimale verdier for spenningen være lik halvparten av de optimale verdiene presentert i oppgave 6.

Ved å endre startverdiene for V_{DC} og V_{AC} , samt massen m til partikkelen i funksjonen `plotStability()` fås i figur 14, et plot av grensen mellom kombinasjoner av V_{DC} og V_{AC} som gir stabile og ustabile baner. Toppunktet er nå omtrent $V_{DC} = 3.8 \text{ V}$ og $V_{AC} = 23 \text{ V}$. Dermed fås det forventede resultatet for optimale spenningsverdier.

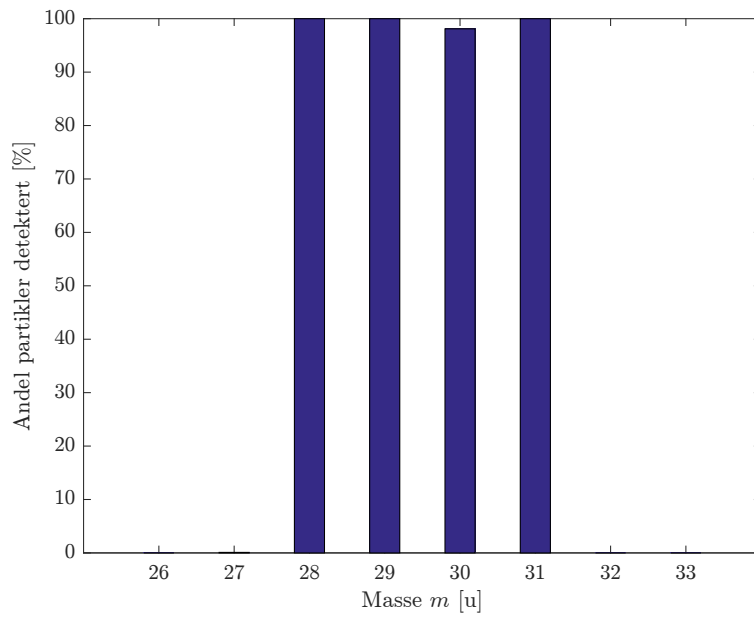
8 Oppgave 8

Det ønskes nå å finne ut hvordan instrumentet påvirkes av at partiklene kommer inn med tilfeldige startposisjoner, gitt ved avstand fra z -aksen $r < r_0/2$, og startvinkler i forhold til z -aksen mellom 0° og 2° . `getRandomInitialConditions` genererer slike tilfeldige fordelte verdier for 1000 partikler. Vi antar her lik v_{z0} (5000 ms^{-1}). `getDetectedPercentage()` tar i bruk `getTrajectoryStability()` (se oppgave 5) for å finne ut hvor stor andel av disse partiklene som detekteres.



Figur 14: Stabilitetsdiagram for N^+ -plot av grensen mellom stabile og ustabile kombinasjoner av V_{DC} og V_{AC} .

I oppgave 6 ble de optimale verdiene $V_{DC} = 7.7\text{ V}$ og $V_{AC} = 46.25\text{ V}$ beregnet. Her velges $V_{DC} = 7.0\text{ V}$, for å detektere partikler med litt større masse. Ved bruk av disse verdiene, og masser i intervallet $[26\text{ u}, 33\text{ u}]$, genererer `plotDetectedPercentage()` stolpediagrammet i figur 15. Man ser at praktisk talt alle partiklene i intervallet $[28\text{ u}, 31\text{ u}]$ detekteres, men ingen utenfor ($\Delta m = 3\text{ u}$). Dette tyder på instrumentet er veldig sensitivt for endringer når man er i grenseområdet for hvilke masser som detekteres (med $V_{DC} = 7.7\text{ V}$ detekteres kun partikler med masse 28 u), noe som er ønskelig.



Figur 15: Andel av 1000 partikler med masse m og tilfeldige startposisjoner og startvinkler detektert.