1 a)

Finn tiden t_m når ballen faller ned på bakken ($y = 0$) og posisjonen $x(t_m) = x_m$ hvor dette skjer.

Vi får oppgitt at:

$$x(t) = v_0 t \cos\theta$$
$$y(t) = v_0 t \sin\theta - \frac{1}{2} g t^2$$

Vi vet at t_m er når ballen treffer bakken. Det er når $y(t) = 0$. Vi løser ligningen over for dette tilfellet:

$$y(t) = v_0 t \sin\theta - \frac{1}{2} g t^2$$
$$0 = v_0 t_m \sin\theta - \frac{1}{2} g t_m^2$$
$$v_0 t_m \sin\theta = \frac{1}{2} g t_m^2$$
$$t_m = 2\frac{v_0}{g} \sin\theta$$

For å finne x_m bruker vi t_m i formelen for x(t):

$$x(t) = v_0 t \cos\theta$$
$$x_m = v_0 t_m \cos\theta$$
$$x_m = 2\frac{v_0^2}{g} \sin\theta \cos\theta$$

1 b)

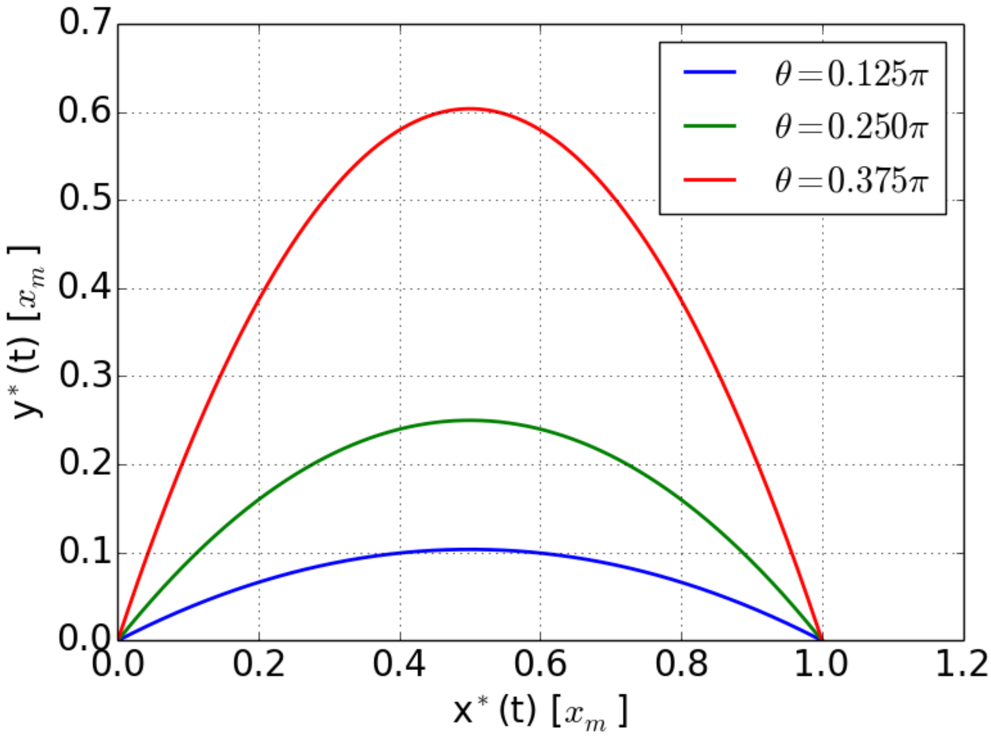
Innfør dimensjonsløse variable (x^*, y^*, t^*) for x,y,t når du skalerer med x_m for lengde og t_m for tid. Forklar hvorfor det ikke er behov for å skalere vinkelen θ.

Vi skalerer x og y med x_m og t med t_m. Det gir:

$$t^* = \frac{t}{t_m} \qquad x^* = \frac{x}{x_m} \qquad y^* = \frac{y}{x_m}$$

1 c)

Bruk Matlab eller Python for å tegne abner (x^*,y^*) for tre utkastvinkler θ_n for n = 1,2,3. Velg $0 < \theta_1 < \frac{\pi}{4}$, $\theta_2 = \frac{\pi}{4}$ og $\frac{\pi}{4} < \theta_3 < \frac{\pi}{2}$. Tegn de tre banene i samme koordinatsystem, og angi hvilken bane som svarer til hvilken utkastvinkel. Forklar hvorfor disse diagrammene kan brukes til å finne ballens baner for forskjellige verdier av utgangsfart v_0 og forskjellige verdier av g.



Figur 1: Grafen viser hvordan banene til ballene med forskjellige utkastvinkel vil være i xy-planet. Programmet som ble brukt for å lage denne figuren finner du bakerst i innleveringen.

Disse diagrammene kan brukes til å finne ballens baner for forskjellige verdier av utgangsfart v_0 og forskjellige verdier av g. Trikset er at informasjonen er lagret i aksene. Altså for at en skal kunne se på effekten av g og v_0, må en pakke ut informasjonen fra aksene. De dimensjonsløse variablene gjør at alle verdier for g og v_0 med samme utkastvinkel gir samme kurve.

2 a)

Finn strømlinjene.

Vi har fått oppgitt hastighetsfeltet:

$$\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} = xy\mathbf{i} + y\mathbf{j}$$

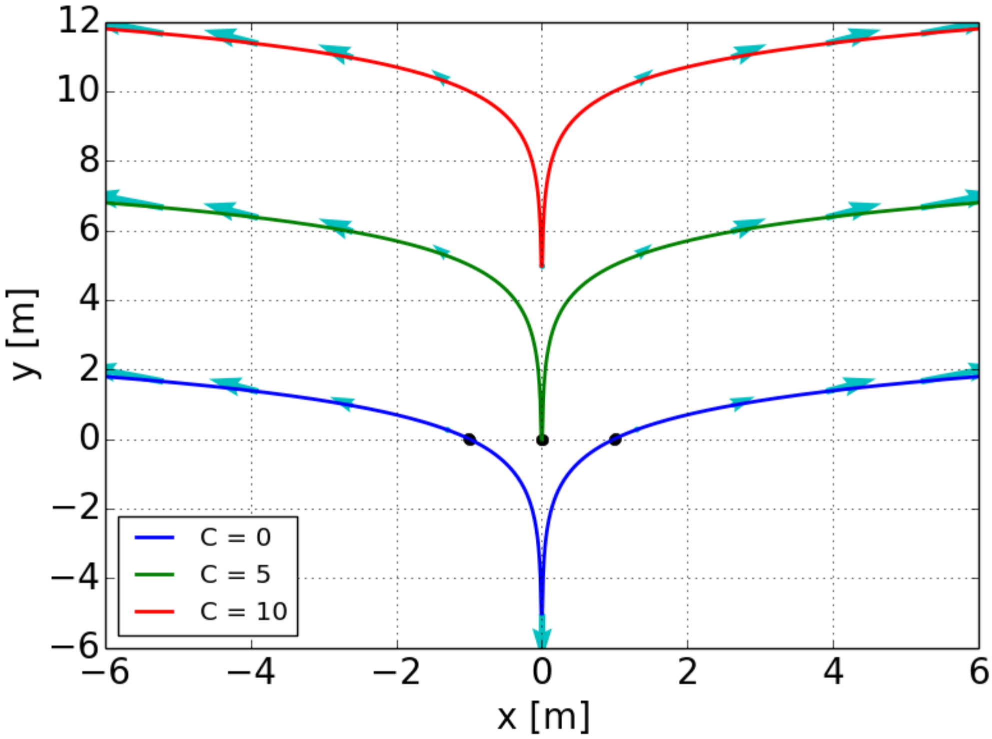
Det er også gitt hint om at denne kan være separabel og vi vet fra GF at $v_x \mathrm{d}x = v_y \mathrm{d}y$ er en nyttig relasjon for å finne strømlinjene:

$$\begin{aligned}
v_x \mathrm{d}y &= v_y \mathrm{d}x \\
xy\mathrm{d}y &= y\mathrm{d}x \\
x\mathrm{d}y &= 1\mathrm{d}x \\
\int 1\mathrm{d}y &= \int \frac{1}{x}\mathrm{d}x \\
y &= \log(x) + C_0 \\
y - \log(x) &= C
\end{aligned}$$

2 b)

Tegn strømlinjene ~~for hånd~~ og sett på piler for å indikere retningen på strømmen. Et stagnasjonspunkt er et punkt hvor hastighetsfeltet er lik null. Finn alle stagnasjonspunktene og identifiser hvor i plottet disse ligger. Det er vanlig å tegne individuelle stagnasjonspunkter som tjukke kulepunkter.

Vis også at du får til å tegne strømlinjene ved hjelp av ~~Matlab eller~~ Python. Dette kan kanskje vise seg å være en utfordring nær x = 0.



Figur 2: Figuren viser tre kurver med forskjellige C. På kurvene er strømvektorene vist som lyseblå vektorer og stagnasjonspunkter er vist som svarte prikker.

Figur 9 viser strømlinjene til hastighetsfeltet som ble oppgitt i oppgave 2. Langs strømlinjene er strømvektorene også tegnet. Retningen til strømvektorene er tangentiell til strømlinjene. Det er uendelig mange stagnasjonspunkter langs x-aksen, hvor to og to hører til en C-verdi.

Jeg glemte å tegne for hånd.

2 c)

Vis at det ikke finnes en strømfunksjon ψ.

Det blir gitt hint om at det er to måter å vise at feltet ikke har en strøm funksjon. Regn ut strømfunksjonen og vis at det er en selvmotsigelse eller hvis at divergensen er ulik null.

Vi skal selvfølgelig gjøre begge:

$$
\begin{aligned}
v_x &= -\partial\psi/\partial y & \qquad v_y &= \partial\psi/\partial x \\
\psi &= -\int v_x \mathrm{d}y & \qquad \psi &= \int v_y \mathrm{d}x \\
\psi &= -\int xy \mathrm{d}y & \qquad \psi &= \int y \mathrm{d}x \\
\psi &= -\frac{1}{2}xy^2 & \qquad \psi &= xy
\end{aligned}
$$

Dette er en åpenbar selvmotsigelse. ψ kan ikke være så bipolar at den er xy og $-\frac{1}{2}xy^2$ samtidig.

Divergensen er lik $\nabla \cdot \mathbf{v} = \partial v_x/\partial x + \partial v_y/\partial y$. Med hastighetsfeltet vårt gir det:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = \frac{\partial xy}{\partial x} + \frac{\partial y}{\partial y}$$

$$\frac{\partial xy}{\partial x} = y \qquad \frac{\partial y}{\partial y} = 1$$

$$\nabla \cdot \mathbf{v} = \underline{\underline{y + 1 \neq 0}}$$

3 a)

Finn divergensen $\nabla \cdot \mathbf{v} = \partial v_x/\partial x + \partial v_y/\partial y$ og virvlingen $\nabla \times \mathbf{v} = (\partial v_y/\partial x - \partial v_x/\partial y)\,\mathbf{k}$ av hastighetsfeltet.

Vi får oppgitt:

$$v_x = \cos(x)\sin(y), \qquad v_y = -\sin(x)\cos(y)$$

Det gir at divergensen, $\nabla \cdot \mathbf{v} = \partial v_x/\partial x + \partial v_y/\partial y$, er lik:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = \frac{\partial \cos(x)\sin(y)}{\partial x} + \frac{\partial - \sin(x)\cos(y)}{\partial y}$$

$$\frac{\partial \cos(x)\sin(y)}{\partial x} = -\sin(x)\sin(y) \qquad \frac{\partial - \sin(x)\cos(y)}{\partial y} = \sin(x)\sin(y)$$

$$\nabla \cdot \mathbf{v} = \sin(x)\sin(y) - \sin(x)\sin(y) = \underline{\underline{0}}$$

Videre kan vi finne virvlingen, $\nabla \times \mathbf{v} = (\partial v_y/\partial x + \partial v_x/\partial y)\mathbf{k}$:

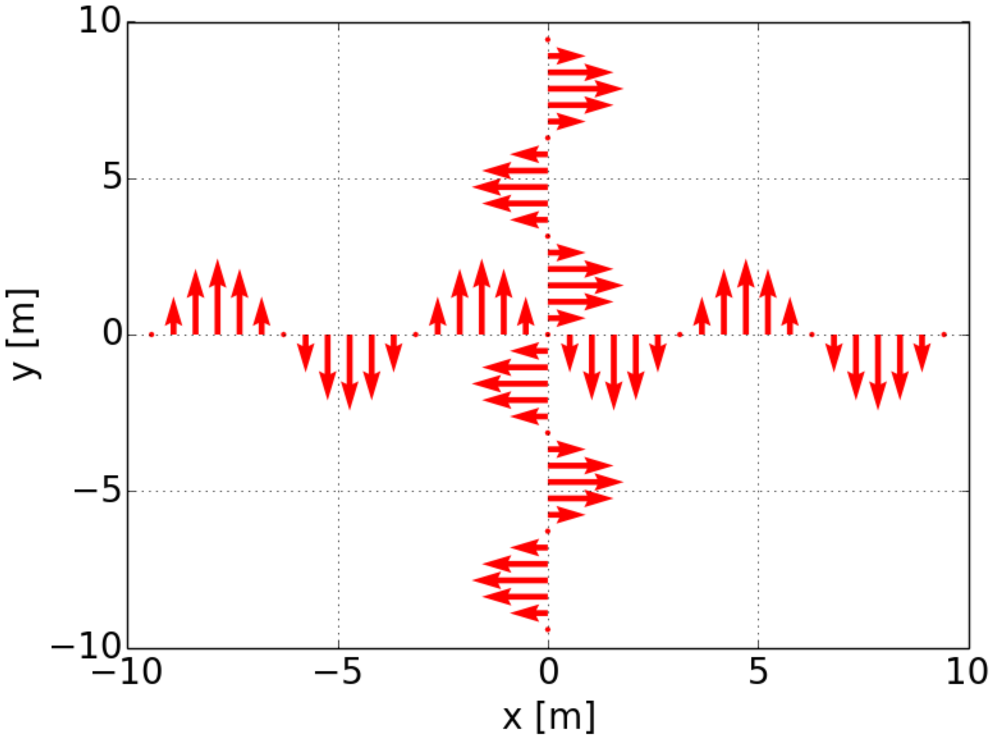
$$\left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right)\mathbf{k} = \left(\frac{\partial - \sin(x)\cos(y)}{\partial x} - \frac{\partial \cos(x)\sin(y)}{\partial y}\right)\mathbf{k}$$

$$\frac{\partial - \sin(x)\cos(y)}{\partial x} = -\cos(x)\cos(y) \qquad \frac{\partial \cos(x)\sin(y)}{\partial y} = \cos(x)\cos(y)$$

$$\nabla \times \mathbf{v} = -\cos(x)\cos(y) - \cos(x)\cos(y)\mathbf{k} = \underline{\underline{-2\cos(x)\cos(y)\mathbf{k}}}$$

3 b)

Tegn opp strømvektorer langs x- og y-aksen.



Figur 3: Vektorene viser hvordan strømvektorene er langs x- og y-aksen. Dette er en sin-funksjon langs aksene.

For å sjekke om figur 3 er korrekt kan en se generelt på hastighetsfeltet langs x-aksen og y-aksen:

Langs x-aksen er $y = 0$

$v_x = \cos(x)\sin(y) = 0$

$v_y = -\sin(x)\cos(y) = -\sin(y)$

Langs y-aksen er $x = 0$

$v_x = \cos(x)\sin(y) = \sin(y)$

$v_y = -\sin(x)\cos(y) = 0$

Dermed vet vi at vi burde få piler langs aksene som varierer som en sin-kurve i størrelse.

3 c)

Finn sirkulasjonen om randa til kvadratet definert ved $-\frac{\pi}{2} \leq x \leq \frac{\pi}{2}$ og $-\frac{\pi}{2} \leq y \leq \frac{\pi}{2}$.

Vi vet fra GF at sirkulasjonen er definert som linjeintegralet over hastighetsfeltet. Vi deler opp integralet i 4 deler. En del for hver sideflate (starter ved nedre sideflate og beveger oss mot klokken):

$$C = \oint \mathbf{v} \mathrm{d}r = \int_a^b \mathbf{v} \cdot \mathrm{d}\mathbf{r} + \int_b^c \mathbf{v} \cdot \mathrm{d}\mathbf{r} + \int_c^d \mathbf{v} \cdot \mathrm{d}\mathbf{r} + \int_d^a \mathbf{v} \cdot \mathrm{d}\mathbf{r}$$

For a til b er $\mathrm{d}\mathbf{r} = \mathrm{d}x\mathbf{i}$. For b til c er $\mathrm{d}\mathbf{r} = \mathrm{d}y\mathbf{j}$. For c til d er $\mathrm{d}\mathbf{r} = \mathrm{d}x{-}\mathbf{i}$. For d til a er $\mathrm{d}\mathbf{r} = \mathrm{d}y{-}\mathbf{j}$. Dette gir oss følgende integraler og resultater:

$$\int_a^b \mathbf{v} \cdot \mathrm{d}\mathbf{r} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathbf{v} \cdot \mathrm{d}x\mathbf{i} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(x)\sin(y)\mathrm{d}x = 2sin(y)$$
$$\int_b^c \mathbf{v} \cdot \mathrm{d}\mathbf{r} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathbf{v} \cdot \mathrm{d}y\mathbf{j} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} -\sin(x)\cos(y)\mathrm{d}y = -2\sin(x)$$
$$\int_c^d \mathbf{v} \cdot \mathrm{d}\mathbf{r} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathbf{v} \cdot \mathrm{d}x{-}\mathbf{i} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} -\cos(x)\sin(y)\mathrm{d}x = -2\sin(y)$$
$$\int_d^a \mathbf{v} \cdot \mathrm{d}\mathbf{r} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathbf{v} \cdot \mathrm{d}y{-}\mathbf{j} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \sin(x)\cos(y)\mathrm{d}y = 2\sin(x)$$
$$\oint \mathbf{v}\mathrm{d}r = 2\sin\left(\frac{\pi}{2}\right) + 2\sin\left(\frac{-\pi}{2}\right) - 2\sin\left(\frac{\pi}{2}\right) - 2\sin\left(\frac{-\pi}{2}\right) = -8$$

3 d)

Forklar hvorfor det eksisterer en strømfunksjon for feltet gitt i likning (1), se hintet i forrige oppgave. Vis at strømfunksjonen kan skrives

$$\psi = cos(x)cos(y) \tag{1}$$

I hintene fra forrige oppgave står det at dersom et vektorfelt er todimensjonalt i xy-planet, $\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j}$ og divergensfritt, $\partial v_x/\partial x + \partial v_y/\partial y = 0$, så eksisterer det en strømfunksjon som angitt ovenfor.

Ovenfor får vi definisjonen $v_x = \partial\psi/\partial y$ og $v_y = \partial\psi/\partial x$. Dermed er det bare å integre v_x med hensyn på y og v_y med hensyn på x.

$$\int v_y \mathrm{d}x = \int -\sin(x)\cos(y)\mathrm{d}x = \underline{\underline{\cos(x)\cos(y)}}$$
$$-\int v_y \mathrm{d}x = -\int \cos(x)\sin(y)\mathrm{d}x = \underline{\underline{\cos(x)\cos(y)}}$$

3 e)

Bruk Taylorutvikling av andre orden til å finne tilnærmede strømlinjer nær origo.

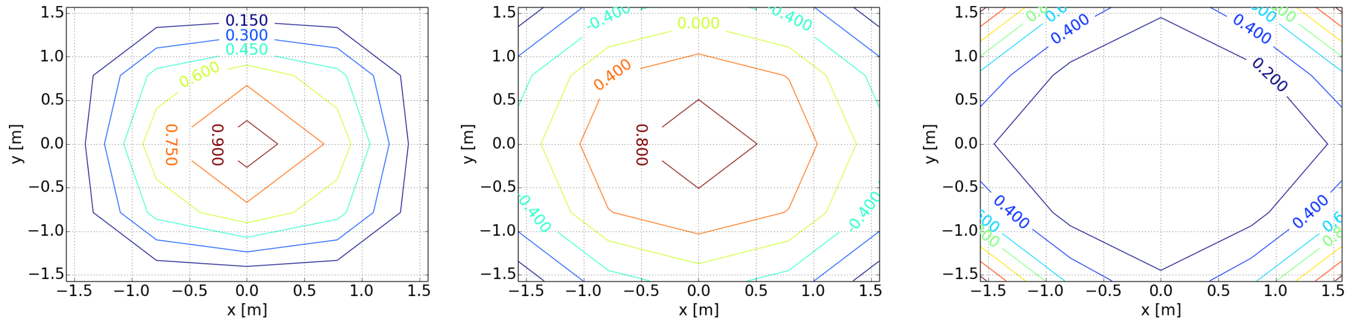
$$\psi(x, y) = T_2(\psi) + \mathcal{O}(x^3, y^3)$$

Vi er kun interessert i $T_2(\psi)$ rundt origo:

$$\begin{aligned}
T_2(\psi)_{0,0} =& \psi(0,0) + \left(\frac{\partial \psi}{\partial x}\right)_{0,0}(x-0) + \left(\frac{\partial \psi}{\partial y}\right)_{0,0}(y-0) + \\
& \frac{1}{2}\left(\frac{\partial^2 \psi}{\partial x^2}\right)_{0,0}(x-0)^2 + \frac{1}{2}\left(\frac{\partial^2 \psi}{\partial y^2}\right)_{0,0}(y-0)^2 + \left(\frac{\partial^2 \psi}{\partial x \partial y}\right)_{0,0}(x-0)(y-0) \\
T_2(\psi)_{0,0} =& \cos(0)\cos(0) - \sin(0)\cos(0)x - \cos(0)\sin(0)y \\
& - \frac{1}{2}\cos(0)\cos(0)x^2 - \frac{1}{2}\cos(0)\cos(0)y^2 + \sin(0)\sin(0)x^2 \\
\\
T_2(\psi)_{0,0} =& \underline{\underline{1 - \frac{1}{2}x^2 - \frac{1}{2}y^2}}
\end{aligned}$$

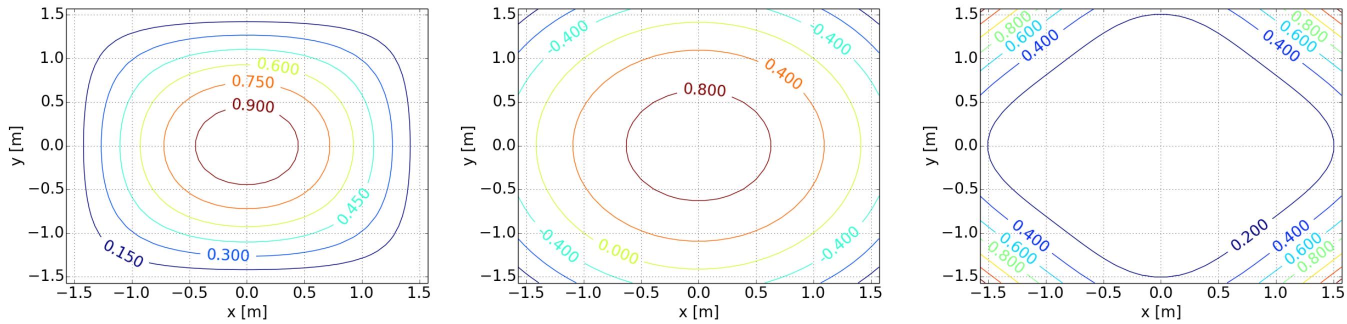
4 a)

Bruk funksjonen streamfun i et skript,~~strlin.m eller~~ strlin.py, som plotter konturlinjer for ψ når $n = 5 \wedge n = 30$. [1]



(a) (b) (c)

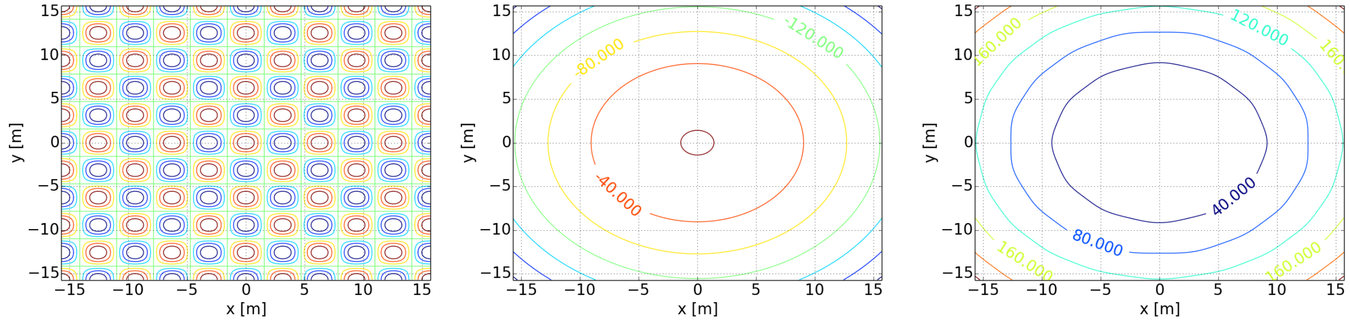
Figur 4: Figurene viser $|x| < \frac{\pi}{2}$ og $|y| < \frac{\pi}{2}$ med $n = 5$. a) viser konturlinjene til ψ. b) viser konturlinjene til Taylor approksimasjon fra 3 e). c) viser konturlinjene til forskjellen mellom a) og b).



(a) (b) (c)

Figur 5: Figurene viser $|x| < \frac{\pi}{2}$ og $|y| < \frac{\pi}{2}$ med $n = 30$. a) viser konturlinjene til ψ. b) viser konturlinjene til Taylor approksimasjon fra 3 e). c) viser konturlinjene til forskjellen mellom a) og b).

[1] $\wedge$ er kul, men feil å bruke i denne sammenhengen.



(a) (b) (c)

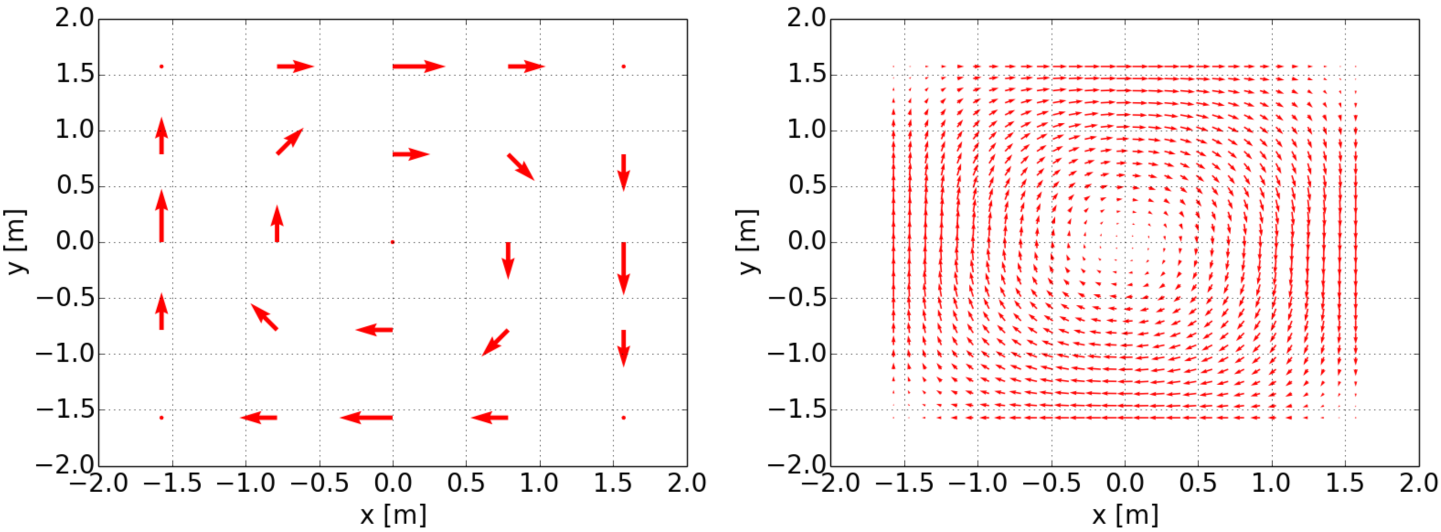
Figur 6: Figurene viser $|x| < \frac{\pi}{2}$ og $|y| < \frac{\pi}{2}$ med $n = 30$. a) viser konturlinjene til ψ. b)viser konturlinjene til Taylor approksimasjon fra 3 e). c) viser konturlinjene til forskjellen mellom a) og b).

Vi ser at for den første svigningen i ψ approksimasjonen fra 3 e) er ganske god. I figur 4 og figur 5 vi kan se at det er kun en feil på sirka 0.2 når vi har kommet til $\frac{\pi}{2}$ på aksene. Dette er å forvente Taylor approksimasjonen skal være best nærmt punktet tilnærmingen er laget. Selv om det ikke er blitt spurt om tar jeg med et plott som viser verdiene til ψ opp til 5π på aksene med n satt til 300. I figur 6 kan vi se resultatet. Jeg tar med dette, fordi det viser hvordan approksimasjonen kun er god rundt toppen i origo, mens den er helt forferdelig på å tilnærme de andre toppene/bunnene.

4 b)

Skriv en funksjon (~~velfield.m eller~~ velfield.py) som bergener hastigheter utfra likning (1) ved kallet $x, y, u, v = velfield(n)$**.**

Bruk denne i et skript, ~~vec.m eller~~ vec.py, som tegner et vektorplott av hastighetsfeltet. Legg vekt på å vlege et passende antall punkter for lesbarheten av plottet.



(a) (b)

Figur 7: Figurene viser vektorplott av hastighetsfeltet ved hjelp av funksjonen definert i velfield.py. a) er vektorplottet med $n = 5$. b) er vektorplottet $n = 30$.

Figur 7 er laget ved scriptene beskrevet i oppgaveteksten. Personlig liker jeg bedre figurer med mange piler (ser mer avansert ut), men jeg legger ved to varianter slik at det er mer lesbart.

```python
# filename: assignment_c1.py
from matplotlib.pyplot import *
from numpy import *

def a1(t,theta = 0.2*pi,v0 = 1):
    x = v0*t*cos(theta)
    y = v0*t*sin(theta) - 0.5 * g*t**2

    #dimensjonslost
    x = x/(2*v0*v0*sin(theta)*cos(theta)/g)
    y = y/(2*v0*v0*sin(theta)*cos(theta)/g)#(0.5*(v0**2)*(sin(theta)**2)/g)
    return x,y


def lag_plot(x,y,label_text,xlabel_text="x$^*$(t) [$x_m$]",\
            ylabel_text="y$^*$(t) [$x_m$]",figfile="test.png"):
    plot(x,y,linewidth=2,label=label_text)
    xlabel(xlabel_text,fontsize=20)
    ylabel(ylabel_text,fontsize=20)
    xticks(fontsize=20)
    yticks(fontsize=20)
    xlim(min(x),max(x)+0.2)
    grid("on")


v0 = 1;     g = 1;     i = 1
for theta in [0.125*pi,0.25*pi,0.375*pi]:
    time = linspace(0,2*v0/g*sin(theta),1000)
    x,y = a1(time,theta,v0)
    lag_plot(x,y,"$\\theta = %.3f\pi$" % (0.125*i),figfile="1a.png")
    i += 1
tight_layout()
legend(loc="best",fontsize=20)
savefig("1a.png" ,bbox_inches="tight")
clf()
```

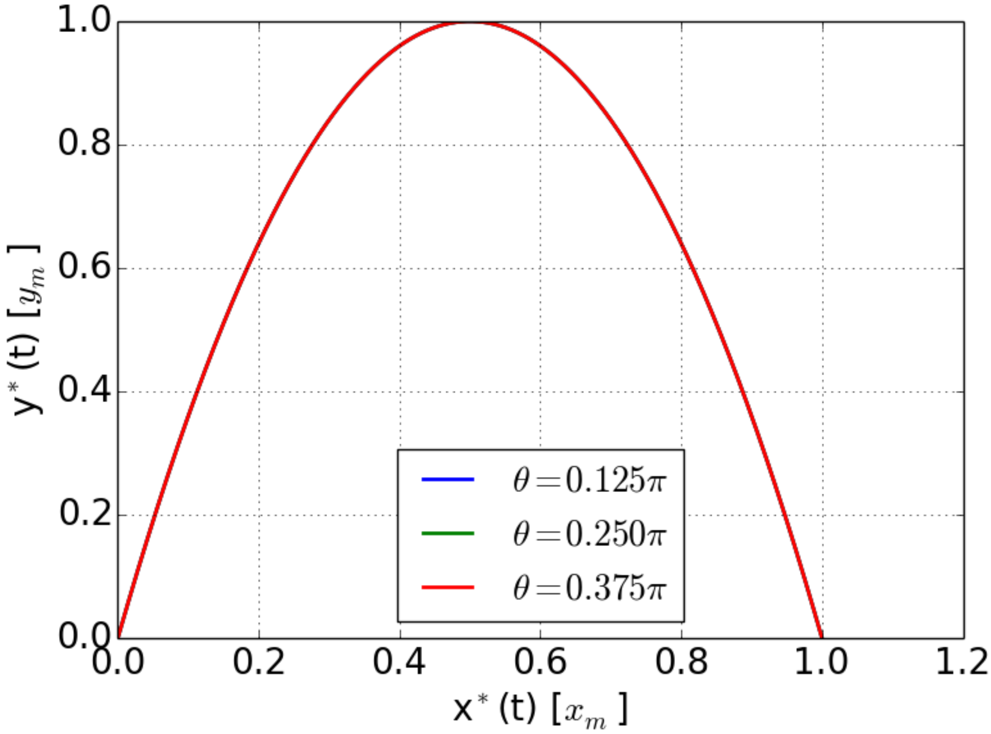
BONUS til oppgave 1b & c:

For y^* kan jeg velge å skalere med toppunktet til $y(t)$. Det finner vi ved å sette inn en halv t_m (fordi dette er en parabel og toppunktet halvveis mellom to ekvivalente verdier, altså $t = 0$ og t_m):

$$y(t) = v_0 t \sin\theta - \frac{1}{2} g t^2$$
$$y(0.5t_m) = v_0 \frac{v_0}{g} \sin\theta \sin\theta - \frac{1}{2} g \frac{v_0^2}{g^2} \sin^2\theta$$
$$y(0.5t_m) = \frac{v_0^2}{g} \sin^2\theta - \frac{1}{2}\frac{v_0^2}{g} \sin^2\theta$$
$$y(0.5t_m) = \frac{1}{2}\frac{v_0^2}{g} \sin^2\theta$$

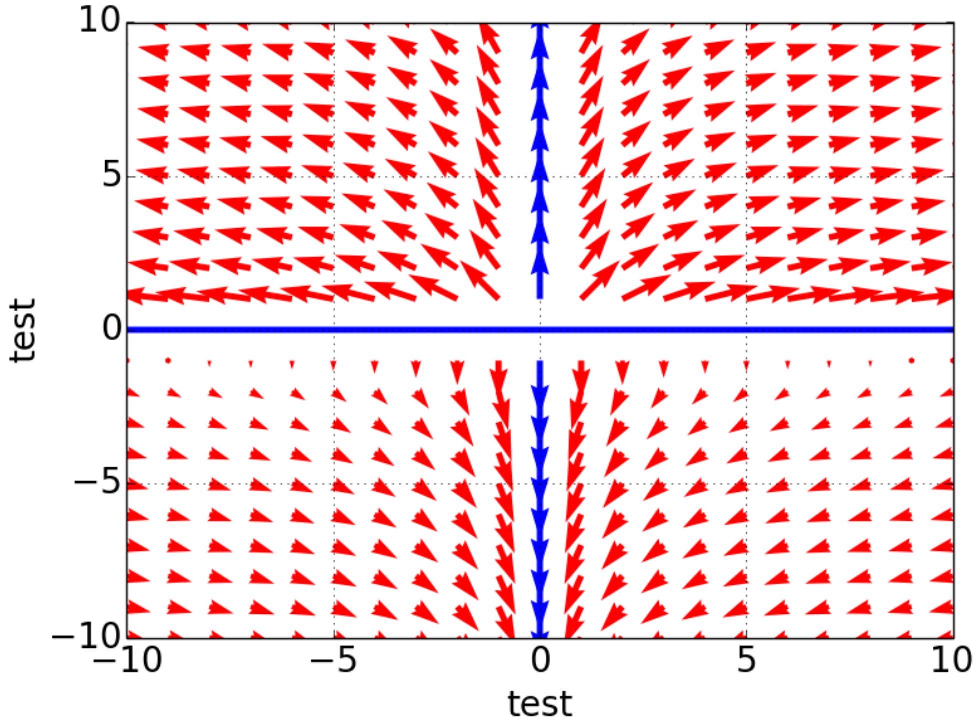
Dermed blir den dimensjonsløse variabelen for y:

$$y^* = y/y(0.5t_m)$$



Figur 8: Grafen viser hvordan banene til ballene med forskjellige utkastvinkel vil være i xy-planet. Programmet som ble brukt for å lage denne figuren finner du bakerst i innleveringen.

Denne innføringen av dimensjonsløse variabler, gjør at alle kast ser like ut. Det er en vis sannhet i det (de uttrykkes ved samme formel). Skaleringen gjør at informasjonen om vinkelen også er pakket inn i aksene, som beskrevet i 1 c).



Figur 9: Dette er et tidligere forsøk på å vise strømvektorene til 2 b). Fargene er for å gjenskape vårt fantastiske flag :).

```
#filename: assignment_b2.py
from matplotlib import pyplot as plt
from numpy import *

def retning(x,y):
        return x*y,y

def y_er_lik(x_verdi,C_verdi):
        return log(abs(x_verdi)) + C_verdi


x = linspace(-6.5,6.5,1000)

ax = plt.subplot(111)
for C in [0,5,10]:
        y_list = []
        vx_list = []
        vy_list = []
        for x_i in x:
                y_i = y_er_lik(x_i,C)
                y_list.append(y_i)
                vx_retning,vy_retning = retning(x_i,y_i)
                vx_list.append(vx_retning)
                vy_list.append(vy_retning)
                if abs(vx_retning) < 0.01+0.02*C and abs(vy_retning) < 0.01+0
                        ax.plot(x_i,y_i,"ko")
        ax.quiver(x[::100],y_list[::100],vx_list[::100],vy_list[::100],color=
        ax.plot(x,y_list,label="C = "+str(C),linewidth=2)
plt.xlim(-6,6)
plt.xlabel("x [m]",fontsize=20)
plt.ylabel("y [m]",fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid("on")
plt.tight_layout()
plt.legend(loc="best")
plt.savefig("2b.png" ,bbox_inches="tight")
plt.show()
```

```
#filename: assignment_b3.py
from matplotlib import pyplot as plt
from numpy import *

def retning(x,y):
        return cos(x)*sin(y),-sin(x)*cos(y)

x_list = []
y_list = []
vx_list = []
vy_list = []
ax = plt.subplot(111)
for x_i in linspace(-3*pi,3*pi,37):
        for y_i in linspace(-3*pi,3*pi,37):
                if x_i == 0 or y_i == 0:
                        vx_retning,vy_retning = retning(x_i,y_i)
                        x_list.append(x_i)
                        y_list.append(y_i)
                        vx_list.append(vx_retning)
                        vy_list.append(vy_retning)
ax.quiver(x_list,y_list,vx_list,vy_list,color="r")
plt.xlabel("x [m]",fontsize=20)
plt.ylabel("y [m]",fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid("on")
plt.tight_layout()
plt.savefig("3b.png" ,bbox_inches="tight")
plt.show()
```

```
#filename: streamfun.py
from numpy import linspace, meshgrid, cos, pi

def streamfun(n=20,func=0,pi_factor=0.5):
    x=linspace(-pi_factor*pi,pi_factor*pi,n)
    [X,Y] = meshgrid(x,x)
    func_list = [cos(X)*cos(Y),(1-((1/2.)*X**2)-((1/2.)*Y**2)),\
                    cos(X)*cos(Y)-(1-((1/2.)*X**2)-((1/2.)*Y**2))]
    psi=func_list[func]
    return X, Y, psi
```

```
#filename: strlin.py
import matplotlib
import numpy as np
import matplotlib.cm as cm
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
from streamfun import *


for func in [0,1,2]:
        for pi_factor in [0.5,5]:
                for n in [5,30,300]:
                        plt.figure()
                        X,Y,Z = streamfun(n,func,pi_factor)
                        CS = plt.contour(X, Y, Z)
                        plt.clabel(CS, inline=1, fontsize=20)
                        plt.xlabel("x [m]",fontsize=20)
                        plt.ylabel("y [m]",fontsize=20)
                        plt.xticks(fontsize=20)
                        plt.yticks(fontsize=20)
                        plt.grid("on")
                        plt.tight_layout()
                        plt.savefig("4a_"+str(func)+"_"+str(pi_factor).replac
                                                "_"+str(n)+".png" ,bbox_inche
                        plt.clf()
```

```
#filename: velfield.py
from numpy import *

def velfield(n=20):
    x=linspace(-0.5*pi,0.5*pi,n)
    [X,Y] = meshgrid(x,x)
    Vx,Vy = cos(X)*sin(Y),-sin(X)*cos(Y)
    return X, Y, Vx, Vy

#filename: vec.py
from matplotlib import pyplot as plt
from numpy import *
from velfield import *

for i in [5,30]:
        x,y,u,v = velfield(i)
        ax = plt.subplot(111)
        ax.quiver(x,y,u,v,color="r")
        plt.xlabel("x [m]",fontsize=20)
        plt.ylabel("y [m]",fontsize=20)
        plt.xticks(fontsize=20)
        plt.yticks(fontsize=20)
        plt.grid("on")
        plt.tight_layout()
        plt.savefig("4b"+str(i)+".png" ,bbox_inches="tight")
        plt.show()
```