16. februar 2018

MEK1100, vår 2018

Obligatorisk oppgave 1 av 2

Innleveringsfrist

Torsdag 1. mars 2018, klokken 14:30 i Devilry (https://devilry.ifi.uio.no).

Instruksjoner

Du velger selv om du skriver besvarelsen for hånd og scanner besvarelsen eller om du skriver løsningen direkte inn på datamaskin (for eksempel ved bruk av LaTeX). Besvarelsen skal leveres som én PDF-fil. Scannede ark må være godt lesbare. Besvarelsen skal inneholde navn, emne og oblignummer.

Det forventes at man har en klar og ryddig besvarelse med tydelige begrunnelser. Husk å inkludere alle relevante plott og figurer. Studenter som ikke får sin opprinnelige besvarelse godkjent, men som har gjort et reelt forsøk på å løse oppgavene, vil få én mulighet til å levere en revidert besvarelse. Samarbeid og alle slags hjelpemidler er tillatt, men den innleverte besvarelsen skal være skrevet av deg og reflektere din forståelse av stoffet. Er vi i tvil om du virkelig har forstått det du har levert inn, kan vi be deg om en muntlig redegjørelse.

I oppgaver der du blir bedt om å programmere må du legge ved programkoden og levere den sammen med resten av besvarelsen. Det er viktig at programkoden du leverer inneholder et kjøreeksempel, slik at det er lett å se hvilket resultat programmet gir.

Søknad om utsettelse av innleveringsfrist

Hvis du blir syk eller av andre grunner trenger å søke om utsettelse av innleveringsfristen, må du ta kontakt med studieadministrasjonen ved Matematisk institutt (e-post: studieinfo@math.uio.no) i god tid før innleveringsfristen.

For å få adgang til avsluttende eksamen i dette emnet, må man bestå alle obligatoriske oppgaver i ett og samme semester.

For fullstendige retningslinjer for innlevering av obligatoriske oppgaver, se her:

www.uio.no/studier/admin/obligatoriske-aktiviteter/mn-math-oblig.html

LYKKE TIL!

Krav til innlevering og godkjenning

Hvert punkt gir makismalt 10 poeng. I alt kan du oppnå 130 poeng. Vi krever minimum 70 prosent, eller 91 poeng for å få godkjent. Dersom dette ikke er innfridd ved innlevering, men besvarelsen vurderes som et seriøst forsøk, kan det gis anledning til ny innlevering.

Der det står at man skal bruke Matlab eller Python, er det fritt fram for å bruke andre systemer som man måtte like bedre (Octave, etc.) så lenge man har til disposisjon de samme numeriske og grafiske kapasitetene.

1 Skalering

En ball kastes ut fra origo over en flat horisontal bakke. x-aksen er horisontal langs bakken og y-aksen peker vertikalt oppover. Ballen kastes ut med fart v_0 ved tiden $t = 0$. Kastet kan innstilles med utkastvinkel θ i forhold til den horisontale x-aksen. Ballen vil følge en bane gitt ved

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

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

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.

c) Bruk Matlab eller Python for å tegne baner (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.

2 Strømlinjer til et todimensjonal hastighetsfelt

Vi skal nå se på hastighetsfeltet

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

a) Finn strømlinjene.

   HINT 1: Du må løse ei differensiallikning som viser seg å være separabel, det vil si at den kan skrives om på formen $f(x)dx = g(y)dy$.

   HINT 2: Fanger du opp at x-aksen også er løsning av den opprinnelige differensiallikninga?

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, ved å bruke kommandoen `contour`. Dette kan kanskje vise seg å være en utfordring nær $x = 0$.

c) Vis at det ikke finnes en strømfunksjon ψ.

HINT: En strømfunksjon $\psi(x, y)$ for et todimensjonalt felt $\boldsymbol{v} = v_x\boldsymbol{i} + v_y\boldsymbol{j}$ i xy-planet har egenskapen $v_x = -\partial\psi/\partial y$ og $v_y = \partial\psi/\partial x$. Dersom en slik strømfunksjon eksisterer så er strømlinjene gitt ved ekviskalarkurvene til strømfunksjonen, $\psi(x, y) =$ konstant.

Dersom et vektorfelt er todimensjonalt i xy-planet, $\boldsymbol{v} = v_x\boldsymbol{i}+v_y\boldsymbol{j}$ og divergensfritt, $\partial v_x/\partial x + \partial v_y/\partial y = 0$, så eksisterer det en strømfunksjon som angitt ovenfor. Dersom et vektorfelt har divergens forskjellig fra null, så eksisterer det ikke en strømfunksjon for feltet.

For å vise at et felt ikke har en strømfunksjon, kan man enten vise at forsøk på å regne den ut ender i en selvimotsigelse, eller man kan vise at divergensen til feltet er ulik null.

3 Et annet todimensjonalt strømfelt

Et hastighetsfelt i xy-planet er gitt ved $\boldsymbol{v} = v_x\boldsymbol{i} + v_y\boldsymbol{j}$ der

$$v_x = \cos(x)\sin(y), \quad v_y = -\sin(x)\cos(y). \tag{1}$$

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

b) Tegn opp strømvektorer langs x- og y-aksen.

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

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

$$\psi = \cos(x)\cos(y). \tag{2}$$

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

4 Strømlinjer og hastighetsfelt i Matlab eller Python

I denne oppgaven skal du skrive noen funskjoner og skript i Matlab eller Python. Disse og de plott som produseres skal leveres inn som en del av besvarelsen. Det er viktig at de filene du lager har de navn som oppgis. Alle figurer skal ha tittel og tekst på akser.

Sammen med oppgaveteksten, på hjemmesiden, finner du filene `streamfun.m` og `streamfun.py` med en funksjon som beregner strømfunksjonen (2) i området. Et kall på funksjonen kan i Matlab se ut som

```
>> [x,y,psi]=streamfun(n);
```

og kan i Python se ut som

```
>>> x,y,psi=streamfun(n)
```

der `x,y,psi` er arrayer for henholdsvis x-verdier, y-verdier og ψ. Inngangsparameteren `n` er antall punkter som brukes i hver retning. Avstanden mellom punktene, i begge retninger, blir da $\Delta x = \Delta y = \frac{\pi}{n-1}$.

a) Bruk funksjonen `streamfun` i et skript, `strlin.m` eller `strlin.py`, som plotter konturlinjer for ψ når

   (i) $n = 5$

   (ii) $n = 30$

   Framstill tilfellene (i) og (ii) i hvert sitt diagram. Sammenhold plottene med punkt e) i forrige oppgave og diskuter de valgte verdier for n.

b) Skriv en funksjon (filnavn `velfield.m` eller `velfield.py`) som beregner hastigheter utfra likning (1) ved kallet

```
>> [x,y,u,v]=velfield(n);
```

   eller

```
>>> 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 velge et passende antall punkter for lesbarheten av plottet.

Matlab-hint

- En array med n jevnt fordelte punkter i intervallet $[a, b]$ lages best med `x=linspace(a,b,n);`.

- Den enkleste måten å lage en utskrift av en figur er å bruke det grafiske grensesnittet til plottevinduet. Man kan også lage en pdf fil med gjeldende figur med
  ```
  print("-dpdf", filnavn);
  ```
  Ønsker du en annen filtype bytter du ut `"-dpdf"` med `"-dXXX"` hvor XXX er ønsket filtype.

- I figurtitler kan det være gunstig å kombinere tekster og verdien av variable som brukes i Matlabkoden. Er s en streng og r et reelt tall og n et heltall kan man f.eks. benytte
  ```
  tit=sprintf("String %s float %.2f int %d", s, r, n); title(tit);
  ```
  Her indikeres hvilken type som kommer med `%s` for streng, `%f` for reelt tall hvor `%.2f` betyr reelt tall med 2 desimaler, og `%d` for heltall.

Python-hint

- En array med n jevnt fordelte punkter i intervallet $[a, b]$ lages best med
  ```
  import numpy as np
  x=np.linspace(a,b,n)}
  ```

- Den enkleste måten å lage en utskrift av en figur er å bruke det grafiske grensesnittet til plottevinduet. Man kan også lage en pdf fil med gjeldende figur med
  ```
  import matplotlib.pyplot as plt
  plt.savefig("fil.pdf")
  ```
  Ønsker du annen filtype bytter du ut `"fil.pdf"` med `"fil.XXX"` hvor XXX er ønsket filtype.

- I figurtitler kan det være gunstig å kombinere tekster og verdien av variable som brukes i Pythonkoden. Er s en streng og r et reelt tall og n et heltall kan man f.eks. benytte
  ```
  plt.title("String {0:s} float {1:.2f} int {2:d}".format(s, r, i))
  ```
  Funksjonen `format` kan brukes på strenger. Sløyfeparentesene definerer hvor i teksten man setter inn noe annet. Tallet bestemmer hvilken verdi som puttes inn, ellers går det fortløpende. Så må man indikere hvilken type som kommer: `s` for streng, `f` for reelt tall hvor `.2f` betyr reelt tall med 2 desimaler, og `d` for heltall.

LaTeX-hint

- For de som ønsker å skrive hele, eller deler, av obligen i LaTeX kan det være nyttig å inkludere alle `.m` eller `.py` filer i LaTeX koden. Dette gjøres enkelt ved verbatimkommandoen
  `\verbatiminput{streamfun.m}`
  En må huske å inkludere verbatim i starten på LaTeX fila
  `\usepackage{verbatim}`