

Første obligatoriske innlevering

MEK4420

SIMON LEDERHILGER

28. januar 2025

Addert masse på geometrier

Vi har sett at ved å linearisere BERNOULLIS likning, kan vi skrive om kreftene på en geometri til en akselerasjon ganger en addert masse, som representerer fortrent fluidvolum grunnet geometriens bevegelse,

$$\mathbf{F} = \int_{\partial\Omega} p \hat{\mathbf{n}} \, dS = -\partial_t \mathbf{U} : \varrho \int_{\partial\Omega} \boldsymbol{\phi} \otimes \hat{\mathbf{n}} \, dS.$$

Her er \mathbf{U} geometriens hastighet, og $\hat{\mathbf{n}} = \hat{\mathbf{n}}_i$ er normalvektoren langs alle frihetsgrader. $\boldsymbol{\phi}$ er en vektorisering av fluidpotensialet i to dimensjoner:

$$\boldsymbol{\phi}(\mathbf{x}, t) = \phi_1 \hat{\mathbf{i}} + \phi_2 \hat{\mathbf{j}} + \phi_6 \hat{\boldsymbol{\omega}}_{\mathbf{k}}.$$

Den symmetriske tensoren for addert masse er da i tensorsubskript gitt ved

$$\mathbf{m} = \varrho \int_{\partial\Omega} \phi_j \hat{\mathbf{n}}_i \, dS.$$

Vi har også sett at følgende gjelder for det todimensjonale tilefellet $r = |\mathbf{x} - \boldsymbol{\mathcal{H}}|$.

$$\int_{\partial\Omega} (\phi \partial_n \ln r - \ln r \partial_n \phi) \, dS = \pi \phi(\boldsymbol{\mathcal{H}}),$$

når $\boldsymbol{\mathcal{H}}$ ligger på randa til geometrien.

Integrallikning på diskret form

Integrallikningen har vi vist kan skrives på diskret form

$$-\pi \phi_n - \sum_{m=1}^N \phi_m \boldsymbol{\theta}_{n,m} = \sum_{m=1}^N \partial_{\hat{\mathbf{n}}} \phi_m \mathbf{h}_{m,n},$$

hvor det er underforstått at subskriptet på ϕ indikerer evaluering i tilsvarende punkt— $\phi_n = \phi(\mathbf{x}_n)$, og $\partial_{\hat{\mathbf{n}}} \phi_n = \partial_{\hat{\mathbf{n}}} \phi(\mathbf{x}_n)$. Vi kan løse integrallikningen for å finne potensialfordelingen ved å invertere matrisen, $\boldsymbol{\phi} = \boldsymbol{\theta}^{-1} \mathbf{h}$. Tensoren for addert masse er gitt ved

$$\mathbf{m} \approx \varrho \sum_{m=1}^N |\mathbf{x}_m - \mathbf{x}_{m-1}| \phi_{j_m} \hat{\mathbf{n}}_{i_m}.$$

Addert masse på en sirkel

For å diskretisere sirkelen konstruerer vi to `linspace`, nemlig θ_p og θ_m , for å så lage to tupler med punkter som ligger på sirkelen. Vi definerer dem på følgende vis.

$$\theta_p = \text{linspace}\left(\frac{2\pi}{N}, 2\pi, N\right)$$

$$\theta_m = \text{linspace}\left(0, \frac{2\pi(N-1)}{N}, N\right)$$

Her er N antall noder. Vi kan nå definere punkter på sirkelen ved å definere $\mathbf{x} = R_0(\cos \theta, \sin \theta)$, for både θ_p og θ_m . Vi ser at θ_m er etterslepene, fordi vi da kan finne midtpunktet ved å ta gjennomsnittet mellom dem. Det er i disse midtpunktene vi vil evaluere ϕ .

Det er hensiktsmessig å lage en klasse for å løse integrallikningen, og midtpunktene bør være en metode her. Vi definerer dermed klassen `IntegralEquation` i filen `integralequation.py`. Vi har da metoden $\boldsymbol{\mathcal{H}} \rightarrow \text{tuple}$, som er definert ved $\boldsymbol{\mathcal{H}} = 1/2(\mathbf{x}_p + \mathbf{x}_m)$. For å sette sammen venstre side av integrallikningen, trenger vi en metode for å konstruere $\boldsymbol{\theta} \in \mathbb{R}^{N \times N}$. Vi gjør dette med `for`-løkker over indeksene i og j .

Algorithm 1 Assemble $\boldsymbol{\theta}$

```

for i, j ≤ N do
  if i = j then
     $\theta_{ij} \leftarrow -\pi$ 
  else
     $\mathbf{a}_p, \mathbf{a}_m \leftarrow \mathbf{x}_p - \boldsymbol{\mathcal{H}}, \mathbf{x}_m - \boldsymbol{\mathcal{H}}$ 
     $\text{argument} \leftarrow \frac{\mathbf{a}_p \cdot \mathbf{a}_m}{|\mathbf{a}_p| |\mathbf{a}_m|}$ 
     $\theta_{ij} \leftarrow -\arccos(\text{argument})$ 
  end if
end for

```

Grunnet maskinregningsfeil, kan `argument` ende opp med verdier litt høyere enn 1, si $1 + \varepsilon$, hvor vi typisk får at $\varepsilon = O(10^{-8})$. Som en følge, vil Numpy vise en feilmelding i `numpy.arccos`. Ved

å legge inn en toleranse, unngår vi feilmeldingen, og koden kompilerer. Toleransen bør sjekkes slik at man både ser at avviket er mindre enn toleransen og at argumentet faktisk er større enn 1.

For å regne ut høyresiden av integrallikningen, må vi sette sammen matrisen \mathbf{h} . Den inneholder en tilnærming av integralet til logaritmefunksjonen over sirkelbuen mellom \mathbf{x}_p og \mathbf{x}_m . Denne tilnærmingen er gjort med en to-punkts GAUSSkvadraturmetode. Utledningen og referanser er å finnes i forelesningsnotatene. Denne matrisen skal skalarmultipliseres med moden vi er ute etters normalvektor. Vi lager en metode som finner normalvektoren. Siden alle normalvektorene ligger på todimensjonale linjestykker, vil normalvektoren være gitt ved $\hat{\mathbf{n}} = |(\Delta \mathbf{x}, \Delta \mathbf{y})|^{-1}(-\Delta \mathbf{y}, \Delta \mathbf{x})$. Vi lager en metode som setter sammen høyresiden av likningen, slik at løsningen kan finnes med `numpy.linalg.solve`.

I en ny fil kan vi konstruere koordinatene for en sirkel, og finne addert masse. I figur 1, ser vi at den teoretiske adderte massen i sjettede mode er identisk oppfylt. Dette er grunnet definisjonen på normalvektoren på en sirkelgeometri—den samsvarer med posisjonsvektoren, så deres kryssprodukt blir null. Konvergensen til den adderte massen i første og andre mode er relativt god. Ser man på verdiene, virker det som en dobling i N vil føre til en reduksjon i relativ feil på faktor fire.

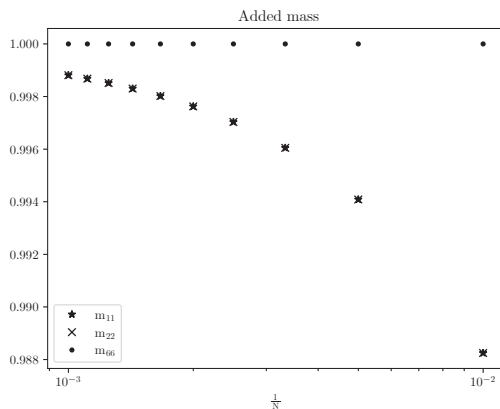


Figure 1: Numerisk utregnet over teoretisk addert masse på en sirkel.

Første modes potensial kan vi finne analytisk, som sett i første forelesningsnotat.

$$\phi_1 = -R_0^2 \partial_x \ln r = -\cos \theta,$$

når vi er på randa til enhetssirkelen. Samsvaret mellom analytisk og numerisk løsning er utmerket for høye verdier av N , som vi ser i figur 2.

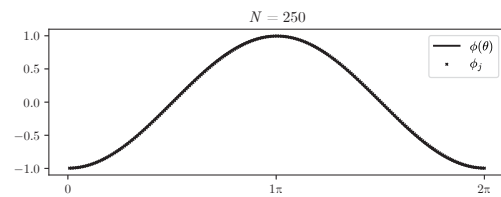


Figure 2: ϕ_1 er gitt analytisk med heltrukket linje, og numerisk med punkter.

Vi kan finne feilen i den numeriske løsningen ved å sjekke L^2 -normen:

$$(\|\phi_j - \phi(\theta)\|_{L^2})^2 = \int_{\Omega} (\phi_j - \phi(\theta))^2 dx.$$

Vi ser at den numeriske løsningen allerede blir veldig god når vi kun har 100 noder på sirkelen.

N	ϕ_1 feil	ϕ_2 feil
10	.547075	.501985
20	.285168	.272773
30	.192806	.187301
40	.145664	.142582
50	.117055	.115091
60	.097843	.096484
70	.084050	.083054
80	.073666	.072906
90	.065566	.064967
100	.059072	.058587

Andre modes potensial kan vi også finne analytisk på samme vis som første modes, dog med deriverte i y . Vi har altså at $\phi_2 = -\sin \theta$.

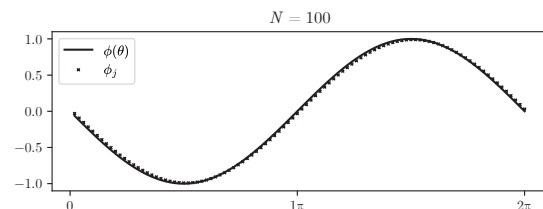


Figure 3: ϕ_2 er gitt analytisk med heltrukket linje, og numerisk med punkter.

Vi formoder at sjettede modes potensial er identisk lik null—vi har ingen heft, så rotasjonen av en rotasjonssymmetrisk geometri bør ikke forflytte fluid. Den numeriske utregningen vår sier seg enig i dette, som sett i figure 4.

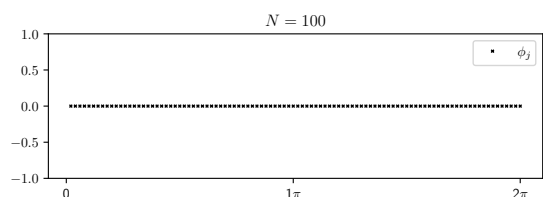


Figure 4: ϕ_6 er gitt numerisk med punkter.

Addert masse på en ellipse

For å regne ut potensial og addert masse på en ellipse kan vi heller modifisere koden brukt til en sirkel, enn å skulle lage en helt ny kode. Vi sier nå at $\mathbf{x} = (a \cos \theta, b \sin \theta)$, hvor a er store halvakse, og b er lille halvakse. Den adderte massen konvergerer fint for både $a/b = 2$ og $a/b = 10$, som vi ser i figurer 5 og 6.

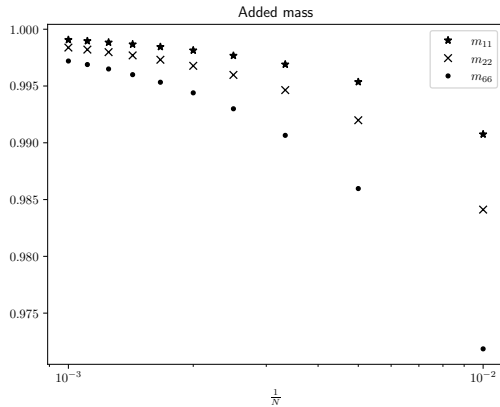


Figure 5: Addert masse for ellipse med $a/b = 2$.

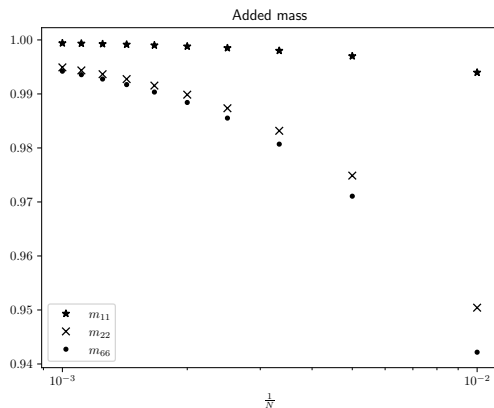


Figure 6: Addert masse for ellipse med $a/b = 10$.

Den dårligere konvergensen for større a kan være grunnet større mellomrom mellom nodene. Siden disse nodene faller nedenfor den faktiske ellipsebuen, vil normalvektoren heller ikke være helt riktig. Dessuten vil diskretiseringen til en sirkel alltid være en regulær flerkant, hvis normalvektorer nødvendigvis peker inn mot origo, som forklarer hvorfor konvergensen er dårligere enn sirkelen. Her ser vi ut til å ha den samme konvergensraten, med at en dobling i N tilsvarer en fjerdedel av relativ feil.

Vi blir ikke spurt om å sammenlikne det numeriske potensialet koden produserer med den analytiske, så vi går ut ifra at de gode resultatene vi så for sirkelen overføres til ellipsen.

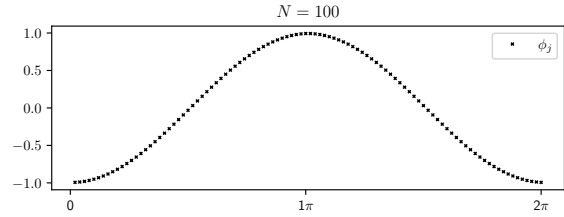


Figure 7: ϕ_1 for ellipse med $a/b = 2$.

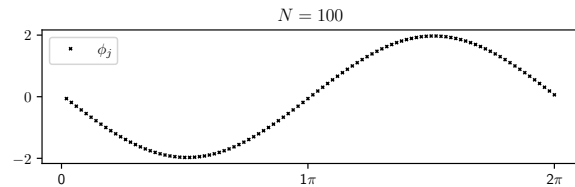


Figure 8: ϕ_2 for ellipse med $a/b = 2$.

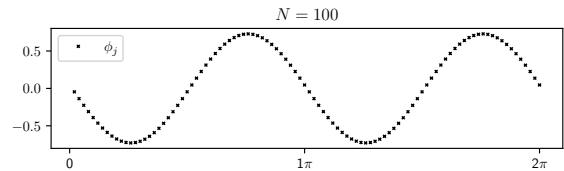


Figure 9: ϕ_6 for ellipse med $a/b = 2$.

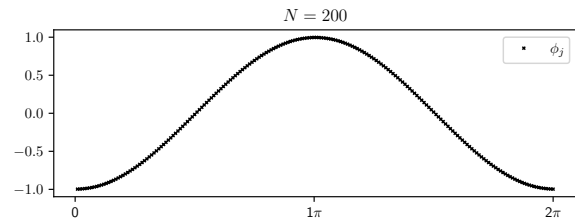


Figure 10: ϕ_1 for ellipse med $a/b = 10$.

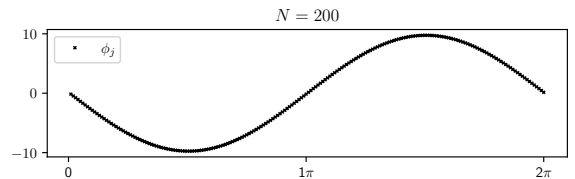


Figure 11: ϕ_2 for ellipse med $a/b = 10$.

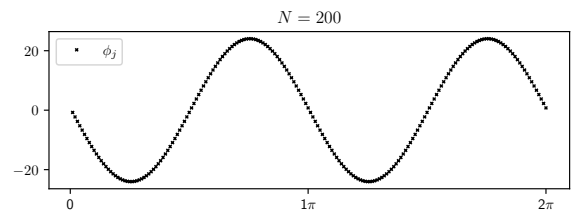


Figure 12: ϕ_6 for ellipse med $a/b = 10$.

Amplituden til første mode ser ut til å tilsvare b , og andre mode a .

Addert masse på et kvadrat

Å skulle diskretisere kvadratet er ikke like trivielt som sirkelen eller ellipsen. Vi ønsker å skulle kunne gjenbruke samme klasse for integrallikningen, så en polar implementering av diskretiseringen er mest opplagt. En kontinuerlig mulighet er en såkalt superellipse, som kan diskretiseres med

$$\mathbf{x}(\theta) = \left(|\cos \theta|^{2/N} \operatorname{sgn}(\cos \theta), |\sin \theta|^{2/N} \operatorname{sgn}(\sin \theta) \right),$$

som vil konvergere raskt mot et kvadrat når N blir stor. Problemet her er at θ ikke faktisk er vinkelen i parametriseringen. Parametriseringsvariablen vil bruke lang tid i hjørnene på superellipsen, og en enorm andel av normalvektorene langs randa vil derfor være fullstendig feil.

En enklere løsning vil være å heller kjøre θ gjennom if-sjekker, som følger.

Algorithm 2 Konstruer kvadrat

```

for  $n \leq N$  do
  if  $\theta_n \in [-\pi/4, \pi/4)$  then
     $\mathbf{x}_n, \mathbf{y}_n \leftarrow 2a, 2a \tan(\theta_n)$ 
  else if  $\theta_n \in [\pi/4, 3\pi/4)$  then
     $\mathbf{x}_n, \mathbf{y}_n \leftarrow 2a \sec(\theta), 2a$ 
  else if  $\theta_n \in [3\pi/4, 5\pi/4)$  then
     $\mathbf{x}_n, \mathbf{y}_n \leftarrow -2a, -2a \tan(\theta_n)$ 
  else if  $\theta_n \in [5\pi/4, 7\pi/4)$  then
     $\mathbf{x}_n, \mathbf{y}_n \leftarrow -2a \sec(\theta), -2a$ 
  end if
end for

```

Denne løsningen er ikke veldig vakker, men den forsikrer at alle nodene faktisk ligger på kvadratet. Dette kan ikke sies for hjørnene i \mathbf{x} , ettersom hjørnenodene kan havne innenfor randa i dersom ikke enten \mathbf{x}_p eller \mathbf{x}_m ligger akkurat i hjørnet for denne N -verdien. Vi ser i figur 13 at dette kan føre til merkelige mønstre i konvergensens til den adderte massen, hvor tilnærmingen faktisk kan bli dårligere for en større N -verdi.

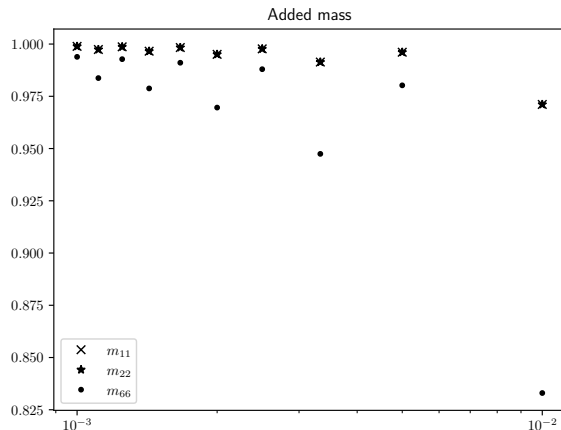


Figure 13: Numerisk utregnet over teoretisk addert masse på et kvadrat.

Samtidig ser vi at den utregnede adderte massen faktisk konvergerer. Vil man unngå å se denne unøyaktigheten, kan man forsikre seg at $\{\pi/4, 3\pi/4, 5\pi/4, 7\pi/4\} \in \theta$ ved å la N alltid være delelig på 8, slik som i figur 14.

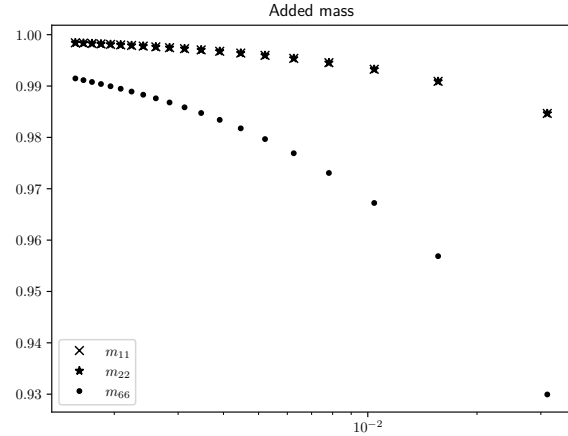


Figure 14: Numerisk utregnet over teoretisk addert masse på et kvadrat med N delelig på 8.

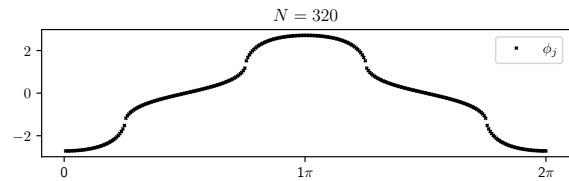


Figure 15: ϕ_1 er gitt numerisk med punkter.

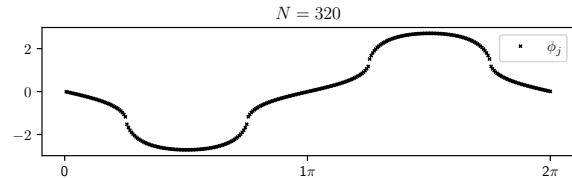


Figure 16: ϕ_2 er gitt numerisk med punkter.

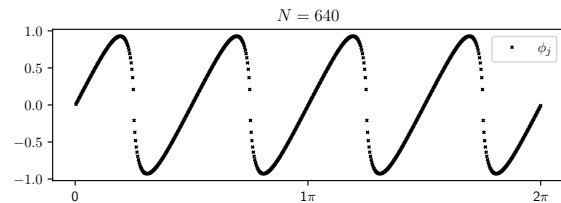


Figure 17: ϕ_6 er gitt numerisk med punkter.