Forelesning 5/4–2018 ved Karsten Trulsen

Litt regning med del-operatoren
Rottmann s.64, M s.82

Eksempel: Se på uttrykket

$$\nabla \times (\overbrace{\boldsymbol{a} \times \boldsymbol{b}}^{\downarrow})$$

hvor pila som peker ned på krøllparentesen indikerer at del-operatoren sin derivasjonsoperasjon skal anvendes på alle variablene som står under krøllparentesen. Legg merke til at parentesen ikke kan sløyfes fordi svaret blir forskjellig dersom vi tar kryssproduktet mellom del og $\boldsymbol{a}$ før kryssproduktet mellom $\boldsymbol{a}$ og $\boldsymbol{b}$.

Det første vi gjør er er å anvende produktregelen for derivasjon

$$\nabla \times (\overbrace{\boldsymbol{a} \times \boldsymbol{b}}^{\downarrow}) = \nabla \times (\overset{\downarrow}{\boldsymbol{a}} \times \boldsymbol{b}) + \nabla \times (\boldsymbol{a} \times \overset{\downarrow}{\boldsymbol{b}})$$

La oss repetere en av de nyttigste formlene vi har

$$\boldsymbol{A} \times (\boldsymbol{B} \times \boldsymbol{C}) = \boldsymbol{B}\boldsymbol{A} \cdot \boldsymbol{C} - \boldsymbol{C}\boldsymbol{A} \cdot \boldsymbol{B}$$

hvor det ikke er nødvendig å skrive parenteser på høyre side fordi svaret blir det samme uansett om vi tar det dyadiske eller prikk produktet først.

Dersom vi bruker denne nyttige formelen "blindt", og beholder pilene for å indikere hva som skal deriveres, får vi

$$\nabla \times (\overbrace{\boldsymbol{a} \times \boldsymbol{b}}^{\downarrow}) = \nabla \times (\overset{\downarrow}{\boldsymbol{a}} \times \boldsymbol{b}) + \nabla \times (\boldsymbol{a} \times \overset{\downarrow}{\boldsymbol{b}}) = \overset{\downarrow}{\boldsymbol{a}}\nabla \cdot \boldsymbol{b} - \boldsymbol{b}\nabla \cdot \overset{\downarrow}{\boldsymbol{a}} + \boldsymbol{a}\nabla \cdot \overset{\downarrow}{\boldsymbol{b}} - \overset{\downarrow}{\boldsymbol{b}}\nabla \cdot \boldsymbol{a}$$

Dette er en unaturlig måte å skrive svaret fordi vi "liker" at del-operatoren virker på det som står til høyre for seg slik at vi slipper å skrive pilene ned. Dersom vi benytter at prikk-produktet mellom to vektorer kommuterer, at prikk-produktet mellom to vektorer er en skalar, og at produktet av en skalar og en vektor kommuterer, så kan vi forenkle uttrykket til

$$\nabla \times (\overbrace{\boldsymbol{a} \times \boldsymbol{b}}^{\downarrow}) = \nabla \times (\overset{\downarrow}{\boldsymbol{a}} \times \boldsymbol{b}) + \nabla \times (\boldsymbol{a} \times \overset{\downarrow}{\boldsymbol{b}}) = \boldsymbol{b} \cdot \nabla\boldsymbol{a} - \boldsymbol{b}\nabla \cdot \boldsymbol{a} + \boldsymbol{a}\nabla \cdot \boldsymbol{b} - \boldsymbol{a} \cdot \nabla\boldsymbol{b}$$

hvor det ikke lenger er behov for piler for å indikere hva som skal deriveres.

Eksempel: Se på uttrykket

$$\nabla(\overbrace{\boldsymbol{a} \cdot \boldsymbol{b}}^{\downarrow})$$

hvor pila som peker ned på krøllparentesen indikerer at del-operatoren sin derivasjonsoperasjon skal anvendes på alle variablene som står under krøllparentesen. Legg merke til at parentesen egentlig er overflødig for å angi rekkefølgen av dyadisk og prikk produkt, men parentesen likevel er nødvendig for å indikere at del-operatoren skal derivere et produkt.

Det første vi gjør er er å anvende produktregelen for derivasjon

$$\nabla(\overbrace{\boldsymbol{a}\cdot\boldsymbol{b}}^{\downarrow}) = \nabla\overset{\downarrow}{\boldsymbol{a}}\cdot\boldsymbol{b} + \nabla\boldsymbol{a}\cdot\overset{\downarrow}{\boldsymbol{b}}$$

Dersom vi nå gjør et par side-regninger

$$\boldsymbol{a}\times(\nabla\times\overset{\downarrow}{\boldsymbol{b}}) = \nabla\boldsymbol{a}\cdot\overset{\downarrow}{\boldsymbol{b}} - \overset{\downarrow}{\boldsymbol{b}}\boldsymbol{a}\cdot\nabla = \nabla\boldsymbol{a}\cdot\overset{\downarrow}{\boldsymbol{b}} - \boldsymbol{a}\cdot\nabla\overset{\downarrow}{\boldsymbol{b}}$$

og

$$\boldsymbol{b}\times(\nabla\times\overset{\downarrow}{\boldsymbol{a}}) = \nabla\boldsymbol{b}\cdot\overset{\downarrow}{\boldsymbol{a}} - \overset{\downarrow}{\boldsymbol{a}}\boldsymbol{b}\cdot\nabla = \nabla\overset{\downarrow}{\boldsymbol{a}}\cdot\boldsymbol{b} - \boldsymbol{b}\cdot\nabla\overset{\downarrow}{\boldsymbol{a}}$$

så ser vi at vi kan skrive

$$\nabla(\overbrace{\boldsymbol{a}\cdot\boldsymbol{b}}^{\downarrow}) = \nabla\overset{\downarrow}{\boldsymbol{a}}\cdot\boldsymbol{b} + \nabla\boldsymbol{a}\cdot\overset{\downarrow}{\boldsymbol{b}} = \boldsymbol{b}\times(\nabla\times\boldsymbol{a}) + \boldsymbol{b}\cdot\nabla\boldsymbol{a} + \boldsymbol{a}\times(\nabla\times\boldsymbol{b}) + \boldsymbol{a}\cdot\nabla\boldsymbol{b}$$

hvor det ikke lenger er behov for piler for å indikere hva som skal deriveres.

Hvordan regner vi ut $\boldsymbol{A}\cdot\boldsymbol{BC}$ for hånd?

For å forenkle så begrenser vi oss til $\boldsymbol{A} = A_x\boldsymbol{i} + A_y\boldsymbol{j}$, $\boldsymbol{B} = B_x\boldsymbol{i} + B_y\boldsymbol{j}$ og $\boldsymbol{C} = C_x\boldsymbol{i} + C_y\boldsymbol{j}$.

Dersom vi regner ut prikk-produktet først får vi

$$\boldsymbol{A}\cdot\boldsymbol{BC} = (A_xB_x + A_yB_y)(C_x\boldsymbol{i} + C_y\boldsymbol{j}) = A_xB_xC_x\boldsymbol{i} + A_xB_xC_y\boldsymbol{j} + A_yB_yC_x\boldsymbol{i} + A_yB_yC_y\boldsymbol{j}$$

Dersom vi regner ut det dyadiske produktet først får vi

$$\boldsymbol{A}\cdot\boldsymbol{BC} = (A_x\boldsymbol{i} + A_y\boldsymbol{j})\cdot(B_xC_x\boldsymbol{ii} + B_xC_y\boldsymbol{ij} + B_yC_x\boldsymbol{ji} + B_yC_y\boldsymbol{jj})$$

Husk at dyadiske produkter ikke kommuterer, og husk at prikk-produktet alltid leter etter første vektor på høyre og venstre side av seg, følgelig får vi

$$\boldsymbol{A}\cdot\boldsymbol{BC} = A_xB_xC_x\boldsymbol{i} + A_xB_xC_y\boldsymbol{j} + A_yB_yC_x\boldsymbol{i} + A_yB_yC_y\boldsymbol{j}$$

altså får vi samme svar uansett om vi gjør prikk eller dyadisk produkt først.

Legg merke til at arbeidsmengeden er størst dersom vi tar det dyadiske produktet først, så det lønner seg alltid å regne ut prikk-produktet først dersom det er mulig.

Hvordan regner vi ut $\boldsymbol{A}\cdot\boldsymbol{BC}$ på datamaskin?

Følgende kode i Python viser hvordan vi kan utføre prikk-produkt med `dot` og dyadisk produkt med `outer`, begge fra `numpy`:

```
>>> import numpy as np
>>> a = np.array([1, 2])
>>> b = np.array([3, 4])
>>> c = np.array([5, 6])
>>> print (a, b, c)
```

```
[1 2] [3 4] [5 6]
>>> print (np.dot (a, b))
11
>>> print (np.dot (a, b) * c)
[55 66]
>>> print (np.outer (b, c))
[[15 18]
 [20 24]]
>>> print (np.dot (a, np.outer (b, c)))
[55 66]
```

Legg merke til at det også finnes en kommando `inner` som kan fungere som erstatning for `dot` i skalarproduktet mellom $\boldsymbol{a}$ og $\boldsymbol{b}$, men som ikke fungerer som erstatning for `dot` i den aller siste linja.

Følgende kode i Matlab/Octave viser hvordan vi kan utføre prikk-produkt som matrisemultiplikasjon mellom en liggende og en stående vektor, og dyadisk produkt som matrisemultiplikasjon mellom en stående og en liggende vektor:

```
octave:1> a = [1, 2]
a =
   1   2
octave:2> b = [3, 4]
b =
   3   4
octave:3> c = [5, 6]
c =
   5   6
octave:4> a*b.'
ans =  11
octave:5> a*b.' * c
ans =
   55   66
octave:6> b.' * c
ans =
   15   18
   20   24
octave:7> a * b.' * c
ans =
   55   66
```

Legg merke til at i Matlab/Octave så betyr kommandoen `.'` å transponere uten å komplekskonjugere, mens kommandoen `'` betyr å transponere og komplekskonjugere. Forskjellen er uvesentlig her fordi våre tall er alle reelle.

Hvordan regner vi ut $\boldsymbol{A} \cdot \nabla \boldsymbol{C}$ for hånd?

Vi begrenser oss til $\boldsymbol{A} = A_x\boldsymbol{i} + A_y\boldsymbol{j}$, $\nabla = \boldsymbol{i}\frac{\partial}{\partial x} + \boldsymbol{j}\frac{\partial}{\partial y}$ og $\boldsymbol{C} = C_x\boldsymbol{i} + C_y\boldsymbol{j}$.

Dersom vi regner ut prikk-produktet først (anbefales) får vi

$$\boldsymbol{A}\cdot\nabla\boldsymbol{C} = (A_x\frac{\partial}{\partial x} + A_y\frac{\partial}{\partial y})(C_x\boldsymbol{i} + C_y\boldsymbol{j}) = A_x\frac{\partial C_x}{\partial x}\boldsymbol{i} + A_x\frac{\partial C_y}{\partial x}\boldsymbol{j} + A_y\frac{\partial C_x}{\partial y}\boldsymbol{i} + A_y\frac{\partial C_y}{\partial y}\boldsymbol{j}$$

Advarsel!!! LH kap. 2.6 definerer Jacobimatrisen på en måte som ser "transponert" ut i forhold til vår notasjon $\nabla\boldsymbol{C}$.

LH bruker matrisenotasjon, og hadde startet med å definere A og C som søylevektorer

$$A = \begin{pmatrix} A_x \\ A_y \end{pmatrix}, \qquad C = \begin{pmatrix} C_x \\ C_y \end{pmatrix},$$

og definert Jacobimatrisen som

$$C' = \begin{pmatrix} \frac{\partial C_x}{\partial x} & \frac{\partial C_x}{\partial y} \\ \frac{\partial C_y}{\partial x} & \frac{\partial C_y}{\partial y} \end{pmatrix},$$

og hadde følgelig skrevet

$$C'A = \begin{pmatrix} \frac{\partial C_x}{\partial x} & \frac{\partial C_x}{\partial y} \\ \frac{\partial C_y}{\partial x} & \frac{\partial C_y}{\partial y} \end{pmatrix} \begin{pmatrix} A_x \\ A_y \end{pmatrix} = \begin{pmatrix} A_x\frac{\partial C_x}{\partial x} + A_y\frac{\partial C_x}{\partial y} \\ A_x\frac{\partial C_y}{\partial x} + A_y\frac{\partial C_y}{\partial y} \end{pmatrix}$$

som tilsvarer vårt yttrykk $\boldsymbol{A}\cdot\nabla\boldsymbol{C}$. LH skriver altså Jacobimatrisen på en form som tilsvarer $\overset{\downarrow}{\boldsymbol{C}}\nabla = (\nabla\boldsymbol{C})^T$, hvor $()^T$ betyr transponert, og hadde i praksis regnet ut vårt uttrykk som $\overset{\downarrow}{\boldsymbol{C}}\nabla\cdot\boldsymbol{A}$ som er ekvivalent med $\boldsymbol{A}\cdot\nabla\boldsymbol{C}$.

Partikkelakselerasjon

I uttrykket for partikkelakselerasjon

$$\boldsymbol{a} = \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{d}t} = \frac{\partial\boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v}$$

ser vi at det konvektivt deriverte leddet er av den typen vi har studert ovenfor. Vi skal se på noen eksempler:

Eksempel: Rotasjon som stivt legeme, konstant rotasjonshastighet rundt z-aksen

La vinkelhastigheten være $\boldsymbol{\omega} = \omega\boldsymbol{k}$ og se på hastighetsfeltet $\boldsymbol{v} = \boldsymbol{\omega}\times\boldsymbol{r} = -\omega y\boldsymbol{i} + \omega x\boldsymbol{j}$. Partikkelakselerasjonen er rent konvektiv fordi vinkelhastigheten antas å være konstant. Det anbefales å regne ut akselerasjonen ved å ta prikk-produktet først

$$\begin{aligned}\boldsymbol{a} = \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{d}t} = \frac{\partial\boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v} &= \boldsymbol{0} + (-\omega y\frac{\partial}{\partial x} + \omega x\frac{\partial}{\partial y})(-\omega y\boldsymbol{i} + \omega x\boldsymbol{j}) \\ &= (-\omega y)(\omega\boldsymbol{j}) + (\omega x)(-\omega\boldsymbol{i}) = -\omega^2(x\boldsymbol{i} + y\boldsymbol{j})\end{aligned}$$

Eksempel: Vis at

$$\frac{\mathrm{D}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{v}$$

hvor $\boldsymbol{r} = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$ er posisjonsvektor og $\boldsymbol{v} = v_x\boldsymbol{i} + v_y\boldsymbol{j} + v_z\boldsymbol{k}$ er hastighetsvektor.

Posisjonsvektor er opplagt stasjonær, så resultatet må være rent konvektivt. Det anbefales å regne dette ut ved å ta prikk-produktet først

$$\frac{\mathrm{D}\boldsymbol{r}}{\mathrm{d}t} = \frac{\partial \boldsymbol{r}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{r} = \boldsymbol{0} + (v_x\frac{\partial}{\partial x} + v_y\frac{\partial}{\partial y} + v_z\frac{\partial}{\partial z})(x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k})$$
$$= v_x\boldsymbol{i} + v_y\boldsymbol{j} + v_z\boldsymbol{k} = \boldsymbol{v}$$

Eksempel: Rotasjon som stivt legeme, konstant rotasjonshastighet orientert i vilkårlig retning

La vinkelhastigheten $\boldsymbol{\omega}$ være vilkårlig, men stasjonær, og se på hastighetsfeltet $\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{r}$. Vi skal regne dette ut på en litt original måte ved å benytte oss av resultatet fra eksemplet ovenfor:

$$\boldsymbol{a} = \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{d}t} = \frac{\mathrm{D}}{\mathrm{d}t}(\boldsymbol{\omega}\times\boldsymbol{r}) = \boldsymbol{\omega}\times\frac{\mathrm{D}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{\omega}\times\boldsymbol{v} = \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}) = \boldsymbol{\omega}\boldsymbol{\omega}\cdot\boldsymbol{r} - \omega^2\boldsymbol{r}$$

Legg merke til at det siste leddet ikke nødvendigvis peker vinkelrett inn mot rotasjonsaksen, ettersom både vinkelhastigheten $\boldsymbol{\omega}$ og posisjonsvektor $\boldsymbol{r}$ er vilkårlig orientert. Derfor trengs det nest siste leddet for å "korrigere" retningen på partikkelakselerasjonen slik at den peker vinkelrett inn mot rotasjonsaksen.