Innlevering, del 1 *2020*

8. september 2020

Det er lov å spørre om hinter!

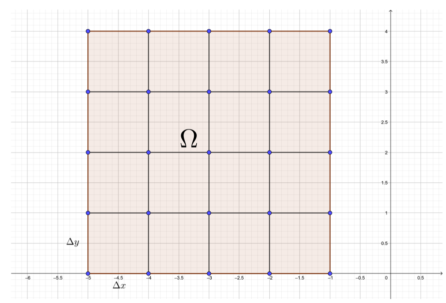
Den fullstendige oppgaven går ut på å beregne ulike typer av integraler numerisk. Første del legger grunnen for dette.

Det kan være bra å notere at jeg bruker begrepet «gitter» litt matematisk tvilsomt, men noen ganger får man ta til poetisk frihet for å forenkle en diskusjon som ellers hadde blitt altfor komplisert. (Et gitter er egentlig alltid parallellogram.)

– Kurver, flater og gitter –

– Rektangulære gitter i $\mathbb{R}^2$ –

Et (rektangulært) *gitter* (eng. *lattice* eller *mesh*) på et område Ω i $\mathbb{R}^2$ er rett og slett bare en utplassering av punkter på Ω (se Figur 1). Observér at linjene mellom punktene ikke er med i selve gitteret. Det skal være samme avstand mellom alle punkter. Normalt forsøker man, så langt som mulig, velge punktene slik at det er samme avstand mellom punktene i x-retning (Δx) og y-retning (Δy), med andre ord, slik at $\Delta x = \Delta y$, men det er ikke alltid mulig.

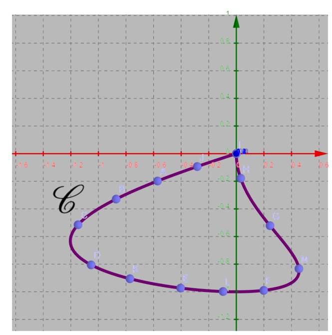


Figur 1: Gitter av $[-5, -1] \times [0, 4]$ i $\mathbb{R}^2$.

Hvordan representerer man et slikt gitter på en PC? Det bør være relativt opplagt: en liste med punkter.

– Gitter på kurver –

Litt vanskeligere er det med et gitter på en kurve (se Figur 2). Årsaken er at vi ikke kan representere en kurve enkelt på en datamaskin. Å andre siden, hvis vi tenker etter litt så er det klart hvordan man skal gjøre dette.



Figur 2: Gitter på kurven $\mathscr{C}$.

La $\mathscr{C}$ ha parametriseringen $\boldsymbol{c} : [a, b] \to \mathbb{R}^2$, der $[a, b]$ er et intervall i $\mathbb{R}$. Tenk på at $\boldsymbol{c}$ er en funksjon

$$\boldsymbol{c} : t \in [a, b] \mapsto \big(c_1(t), c_2(t)\big) \subset \mathbb{R}^2.$$

Hvis vi deler opp $[a, b]$ i

$$a = x_0 < x_1 < x_2 < \cdots < x_n < \cdots < x_{m-1} < x_m = b$$

får vi en mengde tall $\{x_0, x_1, \ldots, x_m\}$ på $[a, b]$ med første punkt lik a, og siste punkt lik b. Hvis vi velger m tilstrekkelig stort (dette verdi er avhengig av $\mathscr{C}$ og hvilken nøyaktighet man trenger) får vi en mengde punkter

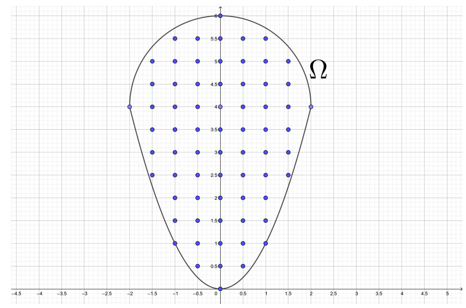
$$\Big\{(c_1(x_0), c_2(x_0)), (c_1(x_1), c_2(x_1)), \ldots, (c_1(x_m), c_2(x_m))\Big\}$$

som er et gitter på $\mathscr{C}$ og er en numerisk representasjon av $\mathscr{C}$.

– Generelle gitter i $\mathbb{R}^2$ –

Vi har sett hva et rektangulart gitter er. Hva da man har et mer generelt område i $\mathbb{R}^2$, si en sirkelskive, eller som i Figur 3?

Ja prinsippet er akkurat det samme: plasser ut punkter på området med jevne mellomrom, et voilà! Observér at det ikke trenger å være punkter på randen overalt.



Figur 3: Gitter på området Ω.

– Gitter på flater –

Det bør være relativt opplagt hva som bør menes med et gitter på en flate, men la meg formulere det matematisk.

La $X(x, y)$ være en parametrisering av en flate $\mathscr{X}$, med andre ord

$$X : U \subset \mathbb{R}^2 \to \mathbb{R}^3, \qquad \mathscr{X} = X(U).$$

Vi trenger ikke anta at X er kontinuerlig.

La G være et gitter på U. Da er $X(G)$ et gitter på $\mathscr{X}$.

– Oppgaver –

Oppgavene er enkle å formulere.

(a) Skriv et program som kan representere rektangulære gitter.

(b) Utvid programmet slik at man kan representere gitter på kurver.

(c) Utvid programmet slik at man kan representere gitter på generelle områder. Observér at dere må fundere ut hvordan man representerer generelle områder i kode.

(d) Utvid programmet slik at man kan representere gitter på flater.

Hvis dere ikke klarer (c) kan dere gjøre (d) for flater der U er et rektangulært område som i (a).

Dere kan bruke vilkårlig programmeringsspråk, men Python er godt egnet til dette.