

Calculus II - Huistaak SymPy

Algemeen

Deze opdracht vervangt het eerder geplande SymPy examen, en moet ingediend worden ten laatste op **vrijdag 3 april 2020 om 18h00** via de daarvoor voorziene assignment op Toledo (Calculus II → assignments → sympy-huistaak 2020), waar je de (ingevulde) worksheet als `.ipynb` bestand moet uploaden. Deze opdracht is vrijblijvend, en kan enkel in positieve zin meetellen voor je eindscore via bonuspunten. We onderscheiden twee gevallen voor het berekenen van je eindscore:

- Geval A: je maakt deze opdracht niet
 - Eerste schriftelijke evaluatie: 8 punten (cesuur < 25%)
 - Online huistaken: 4 punten (cesuur < 37.5%)
 - Online evaluatie 20 april 2020: 8 punten (cesuur < 25%)
- Geval B: je maakt deze opdracht wel
 - Eerste schriftelijke evaluatie: 6 punten (cesuur < 25%)
 - Online huistaken: 4 punten (cesuur < 37.5%)
 - Online evaluatie 20 april 2020: 6 punten (cesuur < 25%)
 - SymPy-opdracht: 4 punten (cesuur < 25%)

Indien je eindscore volgens geval B lager zou zijn dan volgens geval A, dan vervalt de ingediende SymPy-opdracht en wordt je eindscore gegeven volgens geval A.

Instructies

Deze opdracht bestaat voor iedereen uit 3 vragen.

Vraag 1 Het classificeren van kritieke punten voor een functie met meerdere veranderlijken.

Vraag 2 Animatie van een parameterkromme.

Vraag 3 Een lijnintegraal die je moet berekenen met behulp van het pakket `vectorcalculus`.

Voor elke vraag zijn er verschillende opgaven voorzien. **Je lost elke vraag op voor de opgave die jou werd toegewezen.** Deze toewijzing kan je vinden op Toledo. Hier wordt niet van afgeweken: het maken van een ‘verkeerde’ opgave resulteert in geen score voor die vraag. We verduidelijken de verdeling met onderstaand voorbeeld.

Adams	Jonas	0808670	3	Picasso	normaallijn	4
-------	-------	---------	---	---------	-------------	---

Deze student zal voor vraag 1 de derde functie moeten gebruiken. Voor de animatie dient hij de parameterkromme met naam Picasso samen met de normaallijn te animeren. Tot slot lost hij de vierde lijnintegraal op.

Op Toledo kan je een template terugvinden waar imports en symbolen al klaargezet zijn, dit mag je aanvullen op eender welke manier je nodig acht. Je maakt deze opdracht **individueel**. Lees ook goed de extra uitleg bij elke vraag, dit bevat tips om je op weg te helpen.

1 Kritieke punten (3 punten)

Voor deze vraag werken we over de reële getallen. Classificeer voor de gegeven functie f alle kritieke punten en maak een 3D-plot:

- lokaliseer alle zadelpunten;
- lokaliseer alle lokale/globale extrema en bepaal de bijhorende functiewaarden;
- maak 3D-plots waarbij alle extrema en alle zadelpunten zichtbaar zijn.

Staaf al je antwoorden met berekeningen in SymPy en gebruik tekstvakken om je antwoorden te formuleren.

Het nummer dat in de kolom van oefening 1 bij jouw naam staat geeft aan met welke functie je moet werken.

$$1. f(x, y) = \left(1 + \frac{1}{x}\right) \left(1 + \frac{1}{y}\right) \left(\frac{1}{x} + \frac{1}{y}\right)$$

$$2. f(x, y) = xye^{-x^2-y^4}$$

$$3. f(x, y) = \frac{xy}{2+x^4+y^4}$$

$$4. f(x, y) = \frac{1}{x} + \frac{4}{y} + \frac{9}{4-x-y}$$

$$5. f(x, y) = (y^4 - x^2)e^{-(x^2+y^2)}$$

$$6. f(x, y) = (x^2 - y^2 - 9)(x^2 - 25)$$

Opmerking: Het maken van 3D-plots mag je doen via SymPy of via matplotlib, afhankelijk van waar je voorkeur ligt. Voor matplotlib is de makkelijkste methode `plot_surface`, een voorbeeld kan je vinden op <https://matplotlib.org/3.1.0/gallery/mplot3d/surface3d.html>.

Sommige functies f hebben problematische regio's waarbij de plot opblaast en een duidelijke visualisatie moeilijk wordt. In dat geval maak je daarom best meerdere plots waarbij je enkel de regio rond het kritiek punt plot.

2 Animaties van parameterkrommen (5 punten)

In deze vraag animeren we een parameterkromme met bijbehorende normaallijn, raaklijn of kromtecirkel.

2.1 De parameterkrommen

We introduceren 4 verschillende parameterkrommen. Bekijk de verdeling op Toledo om te achterhalen met welke parameterkromme je dient te werken. Het bereik van de parameter t bij elke kromme is $[0, 2\pi]$. Bij elke figuur vermelden we het bereik van het plotvenster om het jezelf makkelijker te maken.

Vlinder. De curve wordt gegeven door

$$\begin{aligned}x(t) &= 2 \cos(t) + \sin(2t) \cos(40t), \\y(t) &= \sin(2t) + \sin(40t).\end{aligned}$$

Plotvenster: het vierkant $[-2.8; 2.8]$.

Epitrochoïde. De curve wordt gegeven door

$$\begin{aligned}x(t) &= \frac{7}{2} \cos(t) - \frac{3}{2} \cos(7t), \\y(t) &= \frac{7}{2} \sin(t) - \frac{3}{2} \sin(7t).\end{aligned}$$

Plotvenster: het vierkant $[-4.5; 4.5]$.

Picasso. De curve wordt gegeven door

$$\begin{aligned}x(t) &= \cos(5t) + \frac{1}{2} \cos(7t) + \frac{1}{2} \sin(14t), \\y(t) &= \sin(5t) + \frac{1}{2} \sin(7t) + \frac{1}{2} \cos(14t).\end{aligned}$$

Plotvenster: het vierkant $[-2; 2]$.

Hart. Om jullie een **hart** onder de riem te steken wordt de laatste curve gegeven door

$$\begin{aligned}x(t) &= 16 \sin^3(t), \\y(t) &= 13 \cos(t) - 5 \cos(2t) - 2 \cos(3t) - \cos(4t).\end{aligned}$$

Plotvenster: $x \in [-17, 17]$ en $y \in [-17, 13]$.

2.2 De animatie

Er zijn 3 verschillende animaties en op de verdeling wordt aangegeven welke animatie je dient te maken. Voor iedere animatie dien je de volledige figuur van de parameterkromme op de achtergrond te laten zien. De code hiervoor hebben we ook al gegeven, je moet deze enkel nog aanpassen naar de variabelen die jij gebruikt. Ook het bereik voor de parameter t hebben we al gedefinieerd, dit staat allemaal in het template.

Normaallijn Maak een animatie waarbij je de curve incrementaal laat tekenen. Teken een blauw bolletje op het eindpunt van de kromme tijdens de animatie. Vervolgens animeer je de normaallijn in het punt dat op dat moment getekend wordt. Indien een parameterkromme gegeven wordt door $x(t)$ en $y(t)$, dan zijn de parametervergelijkingen voor de normaallijn op elk punt t gelijk aan

$$\begin{aligned}x_{nl}(\tau) &= x(t) + y'(t)(\tau - t), \\y_{nl}(\tau) &= y(t) - x'(t)(\tau - t).\end{aligned}$$

Je werkt dus als volgt:

1. De parametervergelijkingen $x(t)$ en $y(t)$ afleiden en evalueren in alle punten t . Gebruik `lambdify` om de SymPy-uitdrukkingen om te zetten naar lijsten van functiewaarden.
2. Bereken de parametervergelijkingen voor de normaallijn op elk punt t , in functie van τ .
3. Vervolgens evalueer je deze voor waarden van τ op elk frame in de animatie.
4. Tenslotte plot je de normaallijn bij op de figuur in het rood.

Raaklijn Maak een animatie waarbij je de curve incrementaal laat tekenen. Teken een blauw bolletje op het eindpunt van de kromme tijdens de animatie. Vervolgens animeer je de raaklijn in het punt dat op dat moment getekend wordt. Indien een parameterkromme gegeven wordt door $x(t)$ en $y(t)$, dan zijn de parametervergelijkingen voor de raaklijn op elk punt t gelijk aan

$$\begin{aligned}x_{rl}(\tau) &= x(t) + x'(t)(\tau - t), \\y_{rl}(\tau) &= y(t) + y'(t)(\tau - t).\end{aligned}$$

Je werkt dus als volgt:

1. De parametervergelijkingen $x(t)$ en $y(t)$ afleiden en evalueren in alle punten t . Gebruik `lambdify` om de SymPy-uitdrukkingen om te zetten naar lijsten van functiewaarden.
2. Bereken de parametervergelijkingen voor de raaklijn op elk punt t , in functie van τ .
3. Vervolgens evalueer je deze voor waarden van τ op elk frame in de animatie.
4. Tenslotte plot je de raaklijn bij op de figuur in het rood.

Kromtecirkel Maak een animatie waarbij je de curve incrementaal laat tekenen. Teken een blauw bolletje op het eindpunt van de kromme tijdens de animatie. Vervolgens animeer je de kromtecirkel in het punt dat op dat moment getekend wordt. Indien een parameterkromme gegeven wordt door $x(t)$ en $y(t)$, dan is het centrum van de kromtecirkel op elk punt t gelijk aan

$$\mathbf{r}_c(t) = \mathbf{r}(t) + \rho(t)\hat{\mathbf{N}}(t).$$

Je werkt dus als volgt:

1. Bepaal de kromming in elk punt, samen met de normaalvector. Denk eraan dat de krommes niet booglente geparametriseerd zijn! Hou ook in gedachten dat SymPy ingebouwde methodes heeft om bijvoorbeeld de norm van een vector te bepalen.
2. Gebruik dit om de kromtestraal af te leiden, en als je weet dat $\mathbf{r}(t)$ de positie is van het punt dat je op dat moment aan het tekenen bent kan je dus het centrum van de kromtecirkel $\mathbf{r}_c(t)$ bepalen.
3. Je weet dat de parametrisatie van een cirkel die niet in de oorsprong ligt wordt gegeven door

$$\begin{aligned}x_o(\theta) &= x_c + \rho \cos(\theta), \\y_o(\theta) &= y_c + \rho \sin(\theta).\end{aligned}$$

Je hebt dus alle gegevens om de kromtecirkel te animeren in elk frame, teken deze in het rood bij op de figuur.

3 Lijnintegralen (2 punten)

In deze vraag bereken je de lijnintegraal die je werd toegewezen met het pakket `vectorcalculus`.

1. Bereken de lijnintegraal van

$$\mathbf{F} = (1+x)e^{x+y} \mathbf{i} + (xe^{x+y} + 2y) \mathbf{j} - 2z \mathbf{k}$$

over de kromme $\mathbf{r}(t) = (1-e)t \mathbf{i} + t \mathbf{j} + 2t \mathbf{k}$ met $0 \leq t \leq 1$.

2. Bereken de lijnintegraal van

$$\mathbf{F} = (xy + 3yz) \mathbf{i} + (x^2 + 3xz + y^2z) \mathbf{j} + (xy + y^3) \mathbf{k}$$

over de halve cirkel met straal 1, middelpunt $(0, 0, 2)$, in het vlak $z = 2$ en waarvoor $y < 0$. We doorlopen hierbij de halve cirkel in tegenwijzerzin.

3. Zoek de arbeid verricht door het krachtveld

$$\mathbf{F} = (y^2 \cos(x) + z^3) \mathbf{i} + (2y \sin(x) - 4) \mathbf{j} + (3xz^2 + 2) \mathbf{k}$$

wanneer een deeltje beweegt langs de curve $x = \sin^{-1}(t)$, $y = 1 - 2t$, $z = 3t - 1$ waarbij $0 \leq t \leq 1$.

4. Bereken de lijnintegraal van

$$\mathbf{F} = \frac{x^2}{y} \mathbf{i} + y \mathbf{j} + \mathbf{k}$$

over de kromme $\mathbf{r}(t) = e^t \mathbf{i} + e^t \mathbf{j} + t \mathbf{k}$ die start in $(1, 1, 0)$ en eindigt in $(e, e, 1)$.

Veel succes!