Vormoptimalisatie in softrobotics

Vladomeare Obolonskyy
Begeleider: Wim Vanroose

27 mei 2024

Universiteit Antwerpen
Faculteit Wetenschappen

Inhoud

1. Probleemstelling

Probleemstelling

Inspiratiebron

Inspiratie soft robot

Probleemstelling
Formulering

- Vormoptimalisatie
- Benadering van een aparte onderdeel

2. Theoretische uitwerking

Theoretische uitwerking

- Drukkracht
- Laplaciaan $\sim$ "veerkracht"

2. Theoretische uitwerking

2D model

Theoretische uitwerking

Druk

Stel p druk, $F_\perp$ kracht orthogonaal op oppervlak A of omtrek S

$$p = \frac{F_\perp}{A} \Leftrightarrow F_\perp = p \cdot A$$

Theoretische uitwerking

Druk

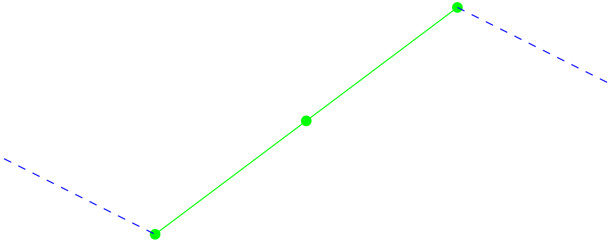
Stel p druk, $F_\perp$ kracht orthogonaal op oppervlak A of omtrek S

$$p = \frac{F_\perp}{A} \Leftrightarrow F_\perp = p \cdot A$$
$$\xrightarrow[\text{Doorsnede}]{} p = \frac{F_\perp}{S} \Leftrightarrow F_\perp = p \cdot S$$

Universiteit Antwerpen
Faculteit Wetenschappen

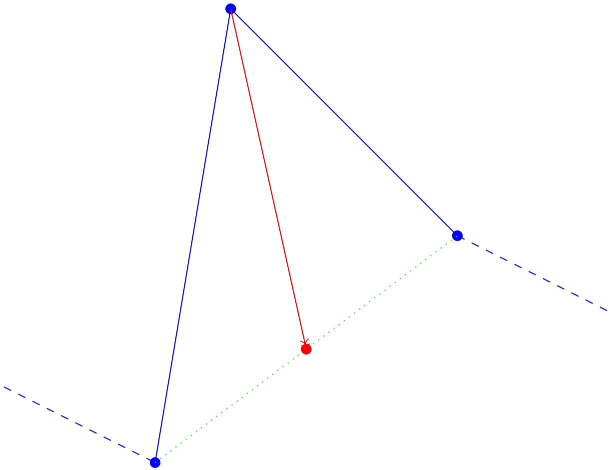
Theoretische uitwerking

Laplaciaan



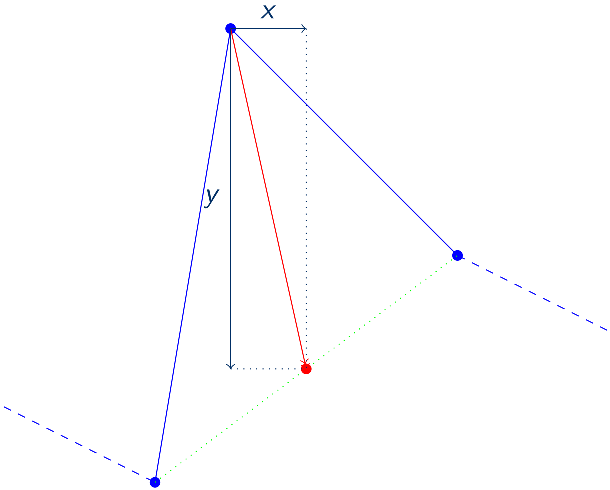
Theoretische uitwerking

Laplaciaan



Theoretische uitwerking

Laplaciaan



Theoretische uitwerking

Laplaciaan

Neem 3 opeenvolgende punten $(x_{i-1}, y_{i-1}), (x_i, y_i)$ en (x_{i+1}, y_{i+1}).

Minimalisatieprobleem $\sim$ Gemiddelde zoeken

$$\begin{cases} x_i = \frac{x_{i-1} + x_{i+1}}{2} \\ y_i = \frac{y_{i-1} + y_{i+1}}{2} \end{cases}$$

x hangt niet af van y en andersom

Theoretische uitwerking

Laplaciaan

We streven naar eentjes op de diagonaal!

$$L_{1D} = \frac{1}{2}\begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix}$$

Theoretische uitwerking

Laplaciaan

Als resultaat krijgen we volgende Laplaciaan matrix voor een twee dimensionaal geval

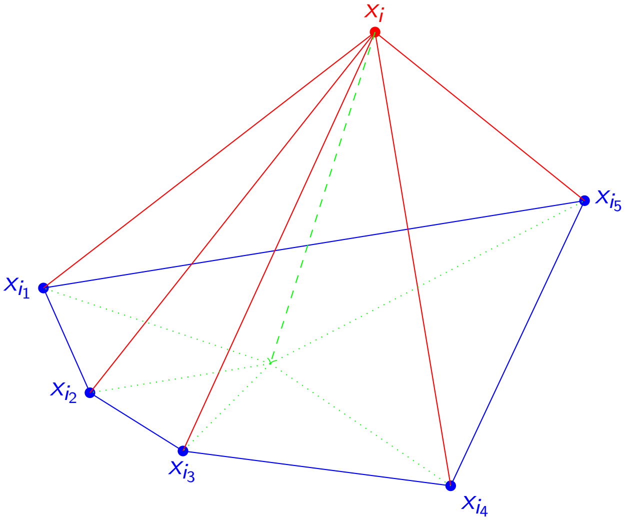
$$L_{2D} = \begin{bmatrix} L_{1D} & 0 \\ 0 & L_{1D} \end{bmatrix}$$

2. Theoretische uitwerking

uitbreiding naar 3D model

Theoretische uitwerking

Laplaciaan



Theoretische uitwerking

Laplaciaan

Elke punt heeft meerdere buren, wat aanleiding geeft tot

$$L_{1D}(i,j) = \begin{cases} -1 & i = j \\ \frac{1}{k_i} & \text{als } j\text{-de punt een buur is} \\ 0 & \text{anders} \end{cases}$$

Theoretische uitwerking

Laplaciaan

Resulterende Laplaciaan matrix is

$$L_{3D} = \begin{bmatrix} L_{1D} & 0 & 0 \\ 0 & L_{1D} & 0 \\ 0 & 0 & L_{1D} \end{bmatrix}$$

3. Praktische uitwerking

Praktische uitwerking

Restricties

- Convexe problemen
- Geordende verzameling punten (mesh structuur)

Praktische uitwerking

Algoritme

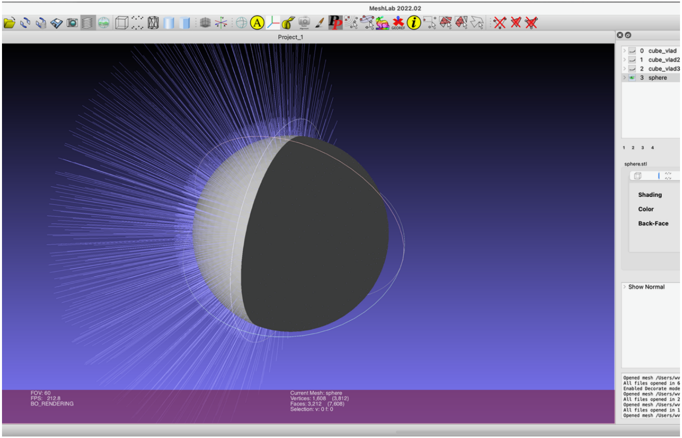
Dit algoritme is algemeen, alles wat overbodig is voor 2 dimensionaal geval staat in rood

- Voorafgaand aan de hoofdprogramma
  - (Vergladding)
  - Structuurvorming (mesh/geordende lijst)
  - Laplaciaan samenstellen
- Hoofdprogramma
  - Massamiddelpunt zoeken
  - Volume/oppervlakte zoeken
  - Drukvectoren zoeken (!!!)
  - Resulterende vector berekenen
  - Plotten

Praktische uitwerking

Probleempunten: Mesh

- Hangt af van programma waar die is opgemaakt
- Definitie binnen- en buitenkant



Praktische uitwerking

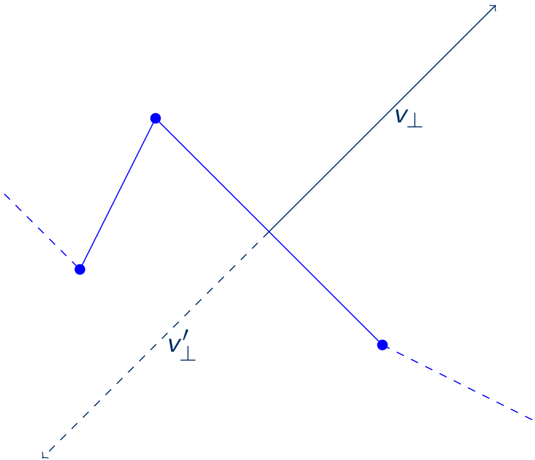
Probleempunten: definitie normaal

Hoe definieren we normaal op een discrete verzamleing?

Praktische uitwerking

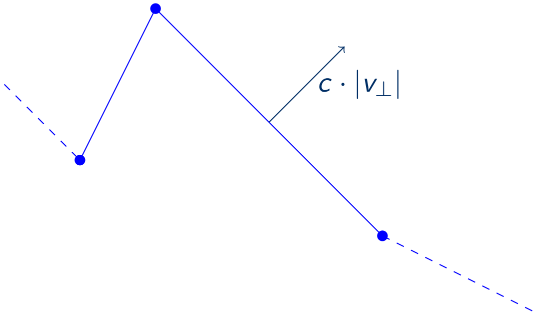
Probleempunten: definitie normaal

Hoe definieren we normaal op een discrete verzamleing?



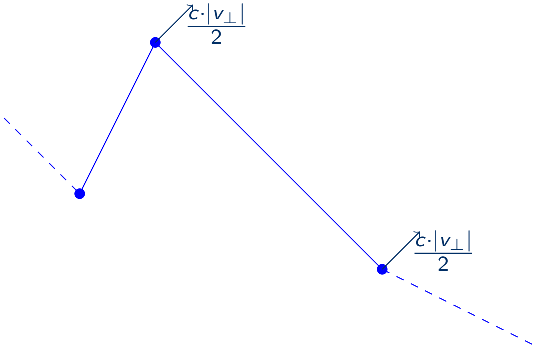
Praktische uitwerking

Probleempunten: definitie normaal



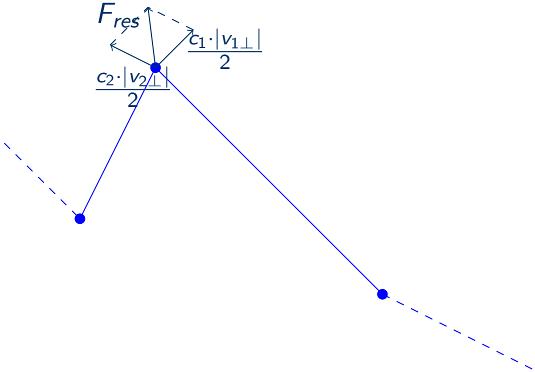
Praktische uitwerking

Probleempunten: definitie normaal



Praktische uitwerking

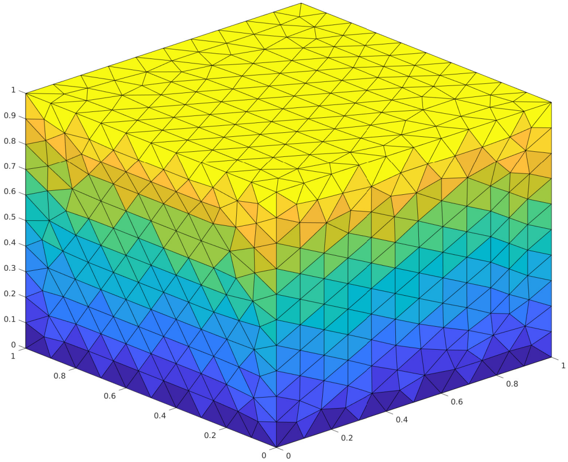
Probleempunten: definitie normaal



4. Resultaten

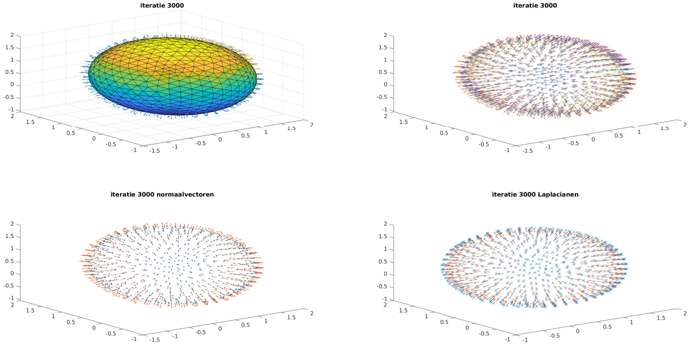
Resultaten

Startsituatie



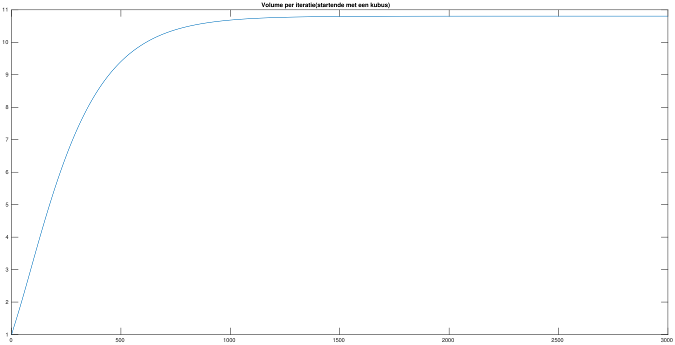
Resultaten

Eindsituatie (iteratie 3000)



Resultaten

Convergentie van onze berekeningen



5. Vragen?