Wetenschappelijk Rekenen

Project 2021-2022 – Cyclische matrices

Inleiding

Doorheen de cursus Wetenschappelijk rekenen hebben we al enkele soorten matrices leren kennen. Denk hierbij bijvoorbeeld aan inverteerbare matrices of positief definiete matrices. Deze eigenschappen geven ons informatie over welke bewerkingen of manipulaties we met deze matrices kunnen uitvoeren. Wanneer we in de praktijk echter matrices tegenkomen is het vaak niet zo voor de hand liggend om zulke moeilijke eigenschappen te bewijzen. Het is meestal veel eenvoudiger om te vertellen hoe de matrix 'eruit ziet'.

Hiervan hebben we al voorbeelden gezien. Zo zijn er tridiagonale matrices, of zeer ijle matrices, of bandmatrices, ... Deze eigenschappen zijn vaak eenvoudig te herkennen, maar vertellen ons eigenlijk niet zoveel over wat we met de matrix kunnen doen. In dit project gaan we matrices met een speciale vorm bestuderen.

1 Cyclische matrices

Bij het numeriek modelleren van sommige praktische problemen verschijnen al eens speciale matrices. Eén zo een type speciale matrices noemen we de *cyclische matrix*, of *circulant matrix* in het Engels.

Een $n \times n$ cyclische matrix wordt gedefiniëerd aan de hand van de n-dimensionale complexe vector $\begin{pmatrix} c_0 & c_1 & \dots & c_{n-1} \end{pmatrix}^\intercal \in \mathbb{C}^n$. Een matrix $\mathbf{C}$ wordt cyclisch genoemd als hij de volgende vorm heeft

$$\mathbf{C} = \begin{pmatrix} c_0 & c_1 & c_2 & \dots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \dots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \dots & c_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & \dots & c_0 \end{pmatrix}. \tag{1}$$

Let er op dat we in de literatuur twee definities terug vinden, deze, en een definitie waarbij we $\mathbf{C}$ transponeren. Bij beide definities is de essentie dezelfde, maar de wiskundige uitwerking van sommige stellingen kan een beetje verschillend zijn. Wanneer je zelf zult gaan grasduinen, zul je misschien het concept *Toeplitz matrix* tegenkomen, dit is een meer algemene vorm van de cyclische matrix. Met andere woorden, de cyclische matrix is een speciaal soort Toeplitz matrix. Voor dit project hoef je je hier geen zorgen over te maken.

Alvorens we enkele interessante eigenschappen of uitdagende toepassingen kunnen bekijken, moeten we eerst enkele basisbewerkingen bestuderen.

Stelling 1. *De volgende bewerkingen met cyclische matrices van dimensie $n \times n$ geven opnieuw cyclische matrices.*

- *De som $\mathbf{A} + \mathbf{B}$ van twee cyclische matrices is een cyclische matrix.*

- *Het product* $\mathbf{A} \cdot \mathbf{B}$ *van twee cyclische matrices is een cyclische matrix.*
- *De inverse van een cyclische matrix* $\mathbf{A}^{-1}$ *is een cyclische matrix.*

Opdracht 1. *Bewijs de drie delen van stelling 1.*

Met deze bewerkingen in de hand kunnen we starten met cyclische matrices in de computer te implementeren. Je mag hier zelf volledig vrij kiezen welke programmeertaal je gebruikt. Er zijn sommige talen of bibliotheken die zelf al gelijkaardige concepten voorzien, deze gebruik je **niet**. Maak het zeer duidelijk hoeveel bewerkingen jouw functies vereisen door te duidelijke code te schrijven, en geen zwaar rekenwerk uit te besteden naar een magische bibliotheek.

Opdracht 2. *Implementeer een* `Vector`*-object. Zorg ervoor dat je hier de basisbewerkingen van vectoren mee kan uitvoeren (optellen, aftrekken, schalen, de lengte of een element opvragen...). Jouw vectoren moeten ondersteuning bieden voor complexe getallen in dubbele precisie*[1]*.*

Opdracht 3. *Implementeer een* `CirculantMatrix`*-object. Bij constructie neemt dit object de elementen* $c_0, \ldots, c_{n-1}$ *als argumenten. Zorg ervoor dat jouw* `CirculantMatrix`*-object ondersteuning heeft voor elk van de volgende bewerkingen.*

1. *Het optellen van cyclische matrices.*
2. *Het vermenigvuldigen van cyclische matrices.*
3. *Het vermenigvuldigen van een cyclische matrix met een vector.*

1.1 Eigenwaarden en eigenvectoren

We hebben ondertussen enkele algoritmen leren kennen om de eigenwaarde en eigenvectoren van matrices te benaderen. Dit zou ons al eens doen vergeten dat in zeldzame gevallen, we de eigenwaarden en eigenvectoren van een matrix exact kunnen berekenen. Voor cyclische matrices kan dit.

Stelling 2. *De* n *eigenwaarden* λ_j *en genormeerde eigenvectoren* $\mathbf{v_j}$ *van een cyclische* $n \times n$ *matrix* $\mathbf{C}$ *worden gegeven door:*

$$\lambda_j = \sum_{k=0}^{n-1} c_k \omega^{kj}$$

$$\mathbf{v_j} = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & \omega^j & \omega^{2j} & \omega^{3j} & \ldots & \omega^{(n-1)j} \end{pmatrix}^\intercal$$

met $j = 0, 1, \ldots n-1$*. Hierbij is* $(c_0, c_1, \ldots, c_{n-1})$ *de eerste rij van* $\mathbf{C}$*, en* $\omega = \exp\left(\frac{2\pi \mathrm{i}}{n}\right)$*. Deze eigenvectoren vormen een orthonormale basis van* $\mathbb{C}^n$*.*

In deze stelling maken we gretig gebruik van rekenregels voor complexe getallen. In het bijzoner hebben we ω neergeschreven als $\exp\left(\frac{2\pi \mathrm{i}}{n}\right)$. Dit wordt ook wel eens de n^{de} eenheidswortel genoemd. Dit is net omdat $\omega^n = 1$. Wanneer je een zeer beperkte ervaring hebt met complexe getallen, kan het vreemd lijken om de exponentiële van een complex getal te berekenen. Hieronder geven we de bijhorende rekenregel voor het complexe getal $a + b\mathrm{i} \in \mathbb{C}$, met $a, b \in \mathbb{R}$:

$$\exp(a + b\mathrm{i}) = e^a \left(\cos(b) + \mathrm{i}\sin(b)\right).$$

Om het eerste deel van de stelling te bewijzen hoeven we ons niet af te vragen hoe men aan de uitdrukkingen voor λ_j en $\mathbf{v_j}$ komt. Het is voldoende na te rekenen dat deze inderdaad correct zijn.

[1] Een complex getal $a + b\mathrm{i}$ kan je opslaan als één `double` voor het reële deel a en één `double` voor het imaginaire deel b. Of gebruik het ingebouwde type voor complexe getallen.

Opdracht 4. *Reken na dat λ_j en $\mathbf{v_j}$, zoals in stelling 2, inderdaad eigenwaarden en eigenvectoren zijn van de cyclische matrix* $\mathbf{C}$*. Je hoeft niet te bewijzen dat deze eigenvectoren een orthonormale basis vormen.*

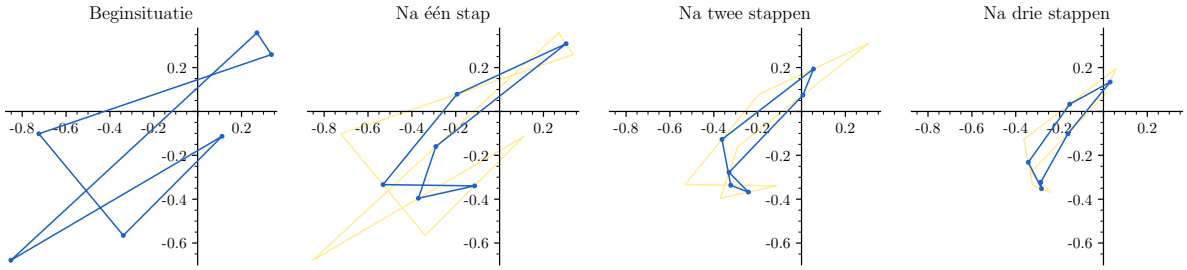
Het kennen van de eigenwaarden en eigenvectoren laat ons toe om andere, op het eerste zicht ongerelateerde, algoritmen te construeren.

Opdracht 5. *Werk een algoritme uit dat de m^{de} macht van een cyclische $n \times n$ matrix* $\mathbf{C}$ *kan berekenen. Wat is de complexiteit? Hierbij mag je veronderstellen dat de* `pow`*-functie in constante tijd uitgevoerd wordt op (complexe) scalairen.*

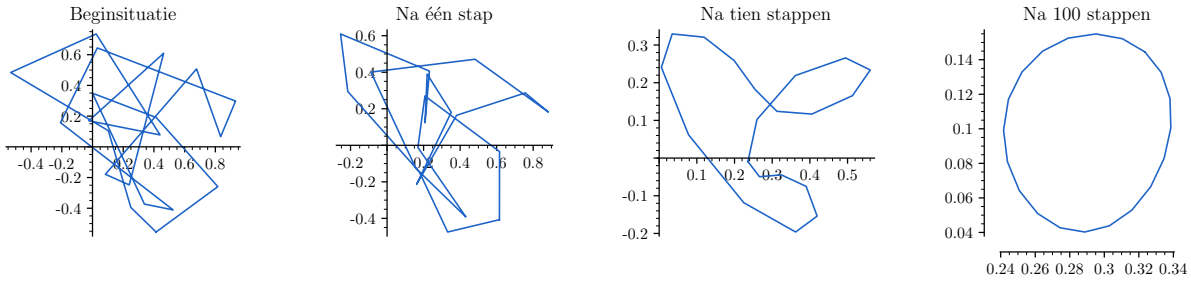
Opdracht 6. *Werk een algoritme uit dat in staat is om* $\mathbf{x}$ *te bepalen in* $\mathbf{Cx} = \mathbf{b}$*, hierbij is* $\mathbf{C}$ *een cyclische $n \times n$ matrix, en* $\mathbf{b}$ *een gegeven n-dimensionale vector.*

2 De middens verbinden

Als een allereerste toepassing van cyclische matrices kijken we naar wiskundig 'spelletje'. We starten met het tekenen van een algemene n-hoek, deze figuur hoeft helemaal niet convex te zijn en mag zichzelf zelfs snijden. We vervolgen het spelletje door de volgende stap te blijven herhalen: verbind de middens van opeenvolgende zijden van de n-hoek, dit geeft een nieuwe n-hoek.



Figuur 1: In elke stap worden de middens van opeenvolgende zijdes verbonden.



Figuur 2: Zelfs wanneer we dit proces starten met een chaotische veelhoek, vereenvoudigt de figuur aanzienlijk.

Figuur 1 illustreert hoe dit proces eruit ziet wanneer we dit toepassen op een willekeurige zeshoek. Het eerste wat ons opvalt, is dat de figuur kleiner wordt. In figuur 2 werken we dit effect tegen, door de assen te wijzigen. Nu valt het ons op dat de vreemde 20-zijdige figuur waarmee we starten steeds eenvoudiger wordt.

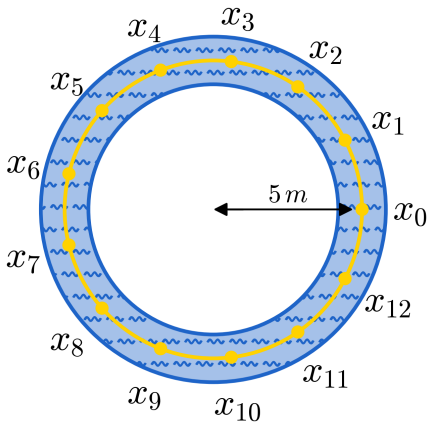
Opdracht 7. *Vertaal dit probleem naar een bewerking op een zekere cyclische matrix.*

Opdracht 8. *Met welke complexiteit kunnen we de situatie voor n punten na m stappen berekenen?*

Opdracht 9. *Bereken de grootste twee eigenwaarden (in modulus) van deze matrix. Wat vertellen de bijhorende eigenvectoren jou over het limietgedrag?*

3 De wildwaterbaan

Vanuit diverse vakgebieden uit de exacte wetenschappen komen vele uitdagingen die zich zeer goed laten vertalen naar een matrixprobleem. Het simuleren van golven in een zwembad is daar een voorbeeld van.



We benaderen een ringvormig zwembad aan de hand van n equidistante punten: $x_0, x_1, \ldots, x_{n-1}$. We noemen Δx de afstand tussen twee opeenvolgende punten, gemeten langs het zwembad.

Figuur 3: Een voorbeeld van een zwembad mat straal $5m$ en een benadering in 13 punten. De afstand tussen twee punten is dus $\Delta x = \frac{10\pi}{13}$ meter.

Om de golven in het zwembad te volgen zeggen we dat $h_i^{(k)}$ de hoogte van het water is in tijdstip k. We noteren de vector $\mathbf{h}^{(k)} = \begin{pmatrix} h_0^{(k)} & h_1^{(k)} & \ldots & h_{n-1}^{(k)} \end{pmatrix}$. Er geldt nu dat de hoogtes op een volgend tijdstip $\mathbf{h}^{(k+1)}$ bepaald kan worden uit de twee vorige tijdstippen $\mathbf{h}^{(k-1)}$ en $\mathbf{h}^{(k)}$:

$$\mathbf{h}^{(k-1)} - 2\mathbf{h}^{(k)} + \mathbf{h}^{(k+1)} = \frac{\Delta t^2}{\Delta x^2}\mathbf{A}\mathbf{h}^{(k+1)} \tag{2}$$

Hierbij is Δt de tijd tussen twee tijdstippen en $\mathbf{A}$ de cyclische $n \times n$ matrix:

$$\mathbf{A} := \frac{1}{12}\begin{pmatrix} -30 & 16 & -1 & 0 & 0 & \ldots & 0 & -1 & 16 \\ 16 & -30 & 16 & -1 & 0 & \ldots & 0 & 0 & -1 \\ -1 & 16 & -30 & 16 & -1 & \ldots & 0 & 0 & 0 \\ 0 & -1 & 16 & -30 & 16 & \ldots & 0 & 0 & 0 \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ -1 & 0 & 0 & 0 & 0 & \ldots & 16 & -30 & 16 \\ 16 & -1 & 0 & 0 & 0 & \ldots & -1 & 16 & -30 \end{pmatrix}.$$

Aan de hand van vergelijking (2) kunnen nu simulaties gemaakt worden van de golven in zo een ringvormig zwembad. Om een nauwkeurige simulatie te verkrijgen is het belangrijk dat Δt voldoende klein is. Stel dus even dat $\Delta t = 0.01$, om de golven op $t = 2$ te berekenen kunnen we het proces uit vergelijking (2) 200 keer herhalen.

Opdracht 10. *Benader de golven in een ringvormig zwembad, met straal 5 meter, net zoals in figuur 3. Experimenteer hiervoor met n, de beginwaarde $\mathbf{h}^{(-1)} = \mathbf{h}^{(0)}$ en Δt.*

3.1 Bonus: exotische zwembaden

Stel dat de vorm van een ringvormig zwembad beschreven wordt door de functie $f : [0, 1] \to \mathbb{R}^2 : t \to f(t)$, met de extra voorwaarde dat $f(0) = f(1)$. Het is, in algemeenheid, helemaal niet meer zo eenvoudig om equidistante punten, gemeten langs de kromme, te kiezen. Om dit toch mogelijk te maken kunnen we gebruikmaken van het feit dat de lengte tussen $f(t_0)$ en $f(t_1)$, als we ze meten langs de kromme, gegeven wordt door:

$$\int_{t_0}^{t_1} \left\| \frac{\mathrm{d}f}{\mathrm{d}t} \right\| \mathrm{d}t. \tag{3}$$

Opdracht 11. *Gebruik* (3) *om 50 equidistante punten te bepalen langs:*

$$f(t) = (2 + \cos(4\pi t)) \begin{pmatrix} \cos(2\pi t) \\ \sin(2\pi t) \end{pmatrix} \qquad \textit{voor } t \in [0, 1].$$

4 De opdracht

Van dit project moet je een verslag schrijven. Dit verslag, en niet jouw code, zal geëvalueerd worden. Je noteert hierin jouw bevindingen en resultaten. Dit verslag zal ook een overzicht bevatten van alle wendingen die jouw onderzoek genomen heeft. Dit project is *individueel*. Het delen van ideeën mag, het delen van code of oplossingen niet.

Je dient dit verslag, samen met je code, in op Ufora, niet later dan **27 mei 2022**.

In deze opgave zijn er twee soorten opdrachten:

- **Implementaties van algoritmen**

  De code hiervoor dien je in. In je verslag noteer je jouw resultaten en bevindingen. Je verantwoordt ook grondig je gemaakte keuzes en geeft de motivering achterliggend aan jouw implementatie.

  Het is aan jou om te illusteren dat dit inderdaad de correcte resultaten geeft. Doe dit door jouw algoritme grondig te beschrijven, maar ook voldoende testen uit te voeren en de resultaten hiervan te rapporteren in het verslag.

- **Wiskundige bewijzen**

  In jouw verslag noteer je de gevraagde bewijzen, duidelijk en net.

Onderzoek

Naast de opdrachten in deze opgave wordt je van harte aangemoedigd om zelf te experimenteren en te onderzoeken. Rapporteer dit ook zeker in jouw verslag.