Modellering en Simulatie

Oefenzitting 7: Snelle Fourriertransformatie

Academiejaar 2015–2016

1 Zonnevlekken

Zonnevlekken zijn relatief donkere vlekken op het oppervlak van de zon. Deze zonnevlekken hangen samen met koelere plekken op de zon. Hun aantal is een maat voor de activiteit van de zon: hoe meer er te zien zijn, hoe actiever de zon. Een actieve zon produceert korte explosies van energie waarbij geladen deeltjes vrijkomen, de zogenaamde zonnewind. Bij hoog-energetische uitbarstingen kan een zonnewind aanleiding geven tot een geomagnetische storm, veroorzaakt wanneer de zonnewind het magnetisch veld van de Aarde bereikt. Zo'n storm kan interferentie veroorzaken met vele communicatiesystemen en het GPS en kan door tijdelijk verhoogde luchtweerstand satellieten lichtjes uit hun baan brengen. In de volgende opgaven zullen we trachten het dominante periodieke patroon in de zonneactiviteit te bepalen door middel van de Fourriertransformatie.

Opgave 1. Laad het bestand `dailyArea.mat` in. Elke rij van deze matrix stelt een meting voor van de totale oppervlakte van alle zichtbare zonnevlekken, uitgedrukt in miljoenste hemisfeer, op een dag. De eerste kolom is het jaartal, de tweede de maand, de derde de dag van de maand en de vierde kolom, tenslotte, geeft de totale oppervlakte van de zonnevlekken op die dag. De dataset die we hier gebruiken omvat alle 36525 dagelijkse metingen gedurende de 100 jaren van 1 januari 1914 tot 31 december 2013. De 4 ontbrekende waarden werden vervangen door de gemiddelde waarde van de dag voor en de dag na de ontbrekende meting. Verdere technische informatie over deze dataset kun je vinden op `http://solarscience.msfc.nasa.gov/greenwch.shtml`.

Gebruik gedurende deze oefenzitting de Matlab functies `fft` en `ifft`. Het is niet de bedoeling dat je zelf het snelle Fourriertransformatiealgoritme implementeert.

Opgave 2. Bepaal de belangrijkste frequentie en de overeenkomstige periode in de zonnevlekkencyclus met behulp van de snelle Fouriertransformatie. De belangrijkste frequentie is deze waarvoor de bijbehorende amplitude het grootst is. Hint: de DC-component laat je uiteraard buiten beschouwing.

Opgave 3. Filter vervolgens de belangrijkste frequenties uit de ruwe data door in de Fourriertransformatie van het signaal enkel de DC-component en de frequenties horende bij de 10 grootste amplitudes te weerhouden. Hint: denk aan de symmetrie.

2 Snelle vermenigvuldigingsalgoritmen

In het talstelsel met basis b wordt een geheel getal $x \in \mathbb{N}$ geschreven in de vorm

$$x = a_0 \cdot b^n + a_1 \cdot b^{n-1} + \cdots + a_{n-1} \cdot b^1 + a_n \cdot b^0$$

waarbij elke $a_i \in [0, \ldots, b-1]$. In dit talstelsel kunnen we x dus neerschrijven als de string "$a_0 a_1 \ldots a_{n-1} a_n$." Bijvoorbeeld als $x = 8$, dan kunnen we dit in basis $b = 10$ noteren als "8," in basis $b = 8$ als "10," in basis 4 als "20" en in basis $b = 2$ als "100." We zullen in deze opgaven een getal x in basis b steeds voorstellen als de vector $\mathbf{a} = \begin{bmatrix} a_0 & a_1 & \cdots & a_n \end{bmatrix}^T$.

*Begeleider: Nick Vannieuwenhoven (`nick.vannieuwenhoven@cs.kuleuven.be`).
†Opgesteld door: L. Vanherpe (2005–2010), N. Achtsis (2010–2014) en N. Vannieuwenhoven (2014–2016)

Opgave 4. Implementeer het DFT-gebaseerde vermenigvuldigingsalgoritme uit de cursus voor willekeurige basis b. Dit algoritme neemt twee vectoren $\mathbf{x}$ en $\mathbf{y}$ als invoer die twee getallen in basis b voorstellen. Verifiëer de correctheid van je algoritme aan de hand van het voorbeeld uit de cursus.

Opgave 5. Bereken 100!.

3 Beeldcompressie herbekeken

Een zwart-witafbeelding wordt op natuurlijke wijze voorgesteld door een matrix met elementen in het interval $[0, 1]$, dewelke de lichtintensiteit voorstelt. De tweedimensionale snelle discrete Fourriertransformatie kan gebruikt worden om zulke afbeeldingen te comprimeren door enkel de frequenties te weerhouden wiens bijbehorende amplitude groot is.

Opgave 6. Laad het bestand `Lighthouse.mat` in. Je kunt de zwart-witafbeelding voorgesteld door `Z` weergeven in Matlab door middel van de `imagesc` functie en de kleurenafbeelding `A` door middel van de `image` functie. Je zou in beide gevallen de Phare du Cap Béar nabij Port Vendres in Frankrijk moeten waarnemen. Hint: gebruik `colormap gray` voor de zwart-witafbeelding.

Opgave 7. Bereken de tweedimensionale Fourriertransformatie met behulp van de functie `fft2` in Matlab. Bekijk het logaritme van de absolute waarde van de getransformeerde afbeelding door deze met `imagesc` weer te geven. Wat valt je op? Welke frequenties zijn het belangrijkste?

Opgave 8. Schrijf op basis van bovenstaande observatie een algoritme om een afbeelding te comprimeren door waarden uit de Fourriergetransformeerde afbeelding gelijk te stellen aan nul. Wat is de compressiefactor van je algoritme?

Opgave 9. Vergelijk de prestaties van het Fourriertransformatie-gebaseerde en het singulierewaardenontbinding-gebaseerde compressiealgoritme. Welk algoritme leidt tot de beste compressiefactor bij eenzelfde benaderingsfout?

Kleurenafbeeldingen worden vaak voorgesteld door middel van het additieve RGB kleurenmodel. Zo wordt in praktijk met elke pixel drie gehele getallen uit het interval $[0, 255]$ geassocieerd: een roodwaarde, groenwaarde en blauwwaarde. De $m \times n$ pixels kleurenafbeelding zou dus voorgesteld kunnen worden door de volgende 3 matrices: de roodwaardenmatrix $R \in \mathbb{R}^{m \times n}$, de groenwaardenmatrix $G \in \mathbb{R}^{m \times n}$ en de blauwwaardenmatrix $B \in \mathbb{R}^{m \times n}$, dewelke respectievelijk de roodwaarden, de groenwaarden en de blauwwaarden voor elke pixel voorstellen. In Matlab worden deze 3 matrices voorgesteld door een driedimensionale *array* $\mathcal{A} \in \mathbb{R}^{m \times n \times 3}$.

Opgave 10. Breid je compressiealgoritme uit zodat ook kleurenafbeeldingen kunnen gecomprimeerd worden door je algoritme afzonderlijk toe te passen op de rood-, groen- en blauwwaarden.