Numerieke Modellering en Benadering: Practicum 1

18 maart 2016

1 Projectoren en de QR factorisatie

In deze sectie onderzoeken we eerst de theoretische eigenschappen van projectoren en de Householder transformatie, waarna deze Householder transformatie getest zal worden in implementaties van de QR factorisatie in MATLAB.

In MATLAB kan je experimenten uitvoeren door commando's in te geven op de invoerregel, maar het is aangewezen om zelf m-files (scripts en functies) te schrijven. Op die manier kan je gemakkelijk wijzigingen aanbrengen en vermijd je het herhaald intypen van gelijkaardige bevelen.

Enkele nuttige commando's in MATLAB zijn: `help`, `lookfor`, `figure`, `subplot`, `plot`, `semilogy`, `semilogx`, `loglog`, `xlabel`, `ylabel`, `title`, `legend`, `print`, `save`, `load`. Voor nog meer nuttige commando's zie ook de MATLAB inleiding op Toledo.

Opgave 1. Stel dat P een $n \times n$ matrix is die als volgt inwerkt op een vector $x \in \mathbb{R}^n$, met n even:

$$Px = \frac{x + Fx}{2}.$$

Hierbij is F de $n \times n$ matrix die de elementen van x als volgt verandert: het eerste element wordt gewisseld met het tweede, het derde met het vierde, enz. Is deze matrix P een projectiematrix, en zoja, is de projectie orthogonaal of schuin? Toon de matrix P en beschrijf het resultaat Px voor een willekeurige $x \in \mathbb{R}^n$.

Opgave 2. Wat zijn de eigenvectoren en bijhorende eigenwaarden van een Householder transformatiematrix? Toon hoe je aan deze resultaten komt. Wat is de geometrische interpretatie van deze waarden?

Opgave 3. Wanneer je de QR factorisatie van een matrix berekent via Householder transformaties, kan je de matrix Q in deze ontbinding ofwel expliciet berekenen ofwel impliciet bepalen door de vectoren v van de opeenvolgende Householder transformaties op te slaan.

a) Schrijf de volgende functies in MATLAB:

- `[Q,R] = Householder_explicit(A)`
  Deze functie krijgt als input een matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, en geeft als output de QR factorisatie (via Householder transformaties) van de matrix waarbij Q expliciet is berekend.
- `[L,R] = Householder_implicit(A)`
  Deze functie krijgt als input een matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, en geeft als output de QR factorisatie van de matrix waarbij de kolommen van de benedendriehoeksmatrix $L \in \mathbb{R}^{m \times n}$ de vectoren v van de opeenvolgende Householder transformaties bevatten.

- `y = Apply_Q(L,b)`
  Deze functie krijgt als input de benedendriehoeksmatrix $L \in \mathbb{R}^{m\times n}$, $m \geq n$, bekomen met de voorgaande functie `Householder_implicit`, en een vector $b \in \mathbb{R}^m$. De output $y \in \mathbb{R}^m$ stelt de vector $Q^T b$ voor, berekend zonder de matrix Q expliciet te vormen.

b) Vergelijk de volgende twee methoden voor het oplossen van een stelsel $Ax = b$:

- De werkwijze met behulp van `Householder_explicit`;
- De werkwijze met behulp van `Householder_implicit` en `Apply_Q`.

Maak deze vergelijking in Matlab op het gebied van snelheid met behulp van de commando's `tic; jouw programma; timing=toc;`. Bestudeer daarna de achterwaartse stabiliteit van beide werkwijzes aan de hand van volgende formule voor vierkante stelsels

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A)\frac{\|r\|}{\|b\|},$$

met r het residu. Bepaal de individuele elementen in deze ongelijkheid en bespreek de scherpte van de grens. Genereer voor deze testen een willekeurige vierkante matrix $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$ en bereken $b = Ax$. De oplossing van beide werkwijzes kan je dan op het eind vergelijken met deze originele x. Variëer de waarde van n $(= 10, 100, 1000)$ en de conditie van de matrix A $(\kappa(A) = 1, 10^4, 10^8)$.

2 Iteratieve methoden

Op Toledo vind je een aantal Matlab-programma's die je kan gebruiken voor de rest van het practicum. Plaats de bestanden in een directory en gebruik in Matlab het commando `cd` om naar deze directory te gaan. Lees grondig de documentatie bij de Matlab-programma's. Type `help pract` voor meer informatie. Het `help` commando kan je ook gebruiken in combinatie met andere Matlab-functies om meer te weten te komen over de werking ervan. Je mag gerust de code van de gegeven Matlab-programma's aanpassen.

2.1 Rayleigh quotiënt

Opgave 4. Stel $A \in \mathbb{R}^{n\times n}$ een symmetrische matrix. Toon aan dat elk Rayleigh quotiënt van A zich in het interval $[\lambda_{min}, \lambda_{max}]$ bevindt, met λ_{min} de minimum en λ_{max} de maximum eigenwaarde van A. Omgekeerd, toon ook aan dat elke waarde in $[\lambda_{min}, \lambda_{max}]$ geschreven kan worden als het Rayleigh quotiënt van A voor een $x \in \mathbb{R}^n$.

Opgave 5. We bestuderen de relatie tussen het QR-algoritme (zonder shift en met Rayleigh quotiënt shift), de Rayleigh quotiënt iteratie en de gelijktijdige iteratie.

a) Bespreek (zeer kort!) de relatie die in de cursus besproken wordt tussen de vermelde methoden. Hoeveel eigenwaarden berekent elk van deze methoden in hun standaard vorm?

b) "*Het convergentiegedrag van het QR-algoritme met Rayleigh quotiënt shift kan gezien worden als een combinatie van dat van de gelijktijdige iteratie en de Rayleigh quotiënt iteratie.*" Licht deze uitspraak uitgebreid toe en verduidelijk je antwoord door het convergentiegedrag van de methoden te tonen voor de matrix `mat1` die te vinden is in het bestand `mat1.txt` op Toledo. Je kan deze matrix in Matlab inladen met behulp van het commando `load mat1.txt`.

2.2 Conjugate gradients, Arnoldi en GMRES

Opgave 6. Stel dat de methode van de toegevoegde gradiënten (conjugate gradients) toegepast wordt op een matrix A, met resultaten $\|e_0\|_A = 1$ en $\|e_{10}\|_A = 2 \times 2^{-10}$. Welke grens kan je op basis van deze informatie geven

- voor het conditiegetal $\kappa(A)$, en
- voor $\|e_{20}\|_A$?

Opgave 7. Naast (quasi)-directe methoden om eigenwaardenproblemen op te lossen, kunnen we ook iteratieve methoden beschouwen, zoals de Arnoldi methode. Iteratieve methoden zijn in het bijzonder geschikt wanneer de matrix een ijle structuur heeft. Genereer een random, ijle matrix $A \in \mathbb{R}^{1000\times1000}$ met het commando `sprand`. Gebruik de gegeven MATLAB-functie van Toledo om een aantal Arnoldi-iteratiestappen op deze matrix toe te passen (bv. 100) en pas de functie aan zodat in elke iteratiestap de Ritz waarden berekend worden. Daarbij mag je gebruik maken van het ingebouwde MATLAB-commando `eig`. Maak een grafiek waarin je toont hoe de Ritz waarden per iteratiestap convergeren naar de eigenwaarden van A. Toon daarbij enkel de reële delen. Bespreek bondig het convergentiegedrag.

Opgave 8. In deze oefening bestuderen we het convergentiegedrag van het GMRES algoritme. Dit algoritme is niet aan bod gekomen in de hoorcolleges, vandaar dat het aangeraden is om eerst hoofdstuk 35 van het handboek door te nemen voor je aan de oefening begint.

a) Implementeer het GMRES algoritme in MATLAB in een functie `[x,itx]=NMB_gmres(A,b)`, waarbij A en b het gegeven stelsel opmaken, x de oplossing van de GMRES methode is en `itx` een matrix is met als kolommen de schattingen voor x doorheen de iteratiestappen. Vertrek voor het schrijven van deze functie van de Arnoldi methode die je op Toledo kan vinden.

b) Pas je algoritme toe om het stelsel $Ax = b$ iteratief op te lossen, waarbij A en b gegeven zijn door:

```
A = sprand(m,m,0.5);
A = A + alpha*speye(m); A=A/norm(A,1);
b = rand(m,1);
```

Toon in een figuur de norm van het residu, $\|r_n\|$, en de fout ten opzichte van de exacte oplossing in functie van de iteratiestap. Je mag `A\b` gebruiken als exacte oplossing. Gebruik `m = 100` als grootte van het stelsel en toon de resultaten voor `alpha` = 1, 5, 10 en 100.

c) Kan je een kwalitatieve verklaring geven voor de verschillen in convergentiegedrag?

2.3 Alternatieve eigenwaardenalgoritmen

Naast de besproken methoden in lecture 27 t.e.m. lecture 29 van het boek 'Numerical Linear Algebra' (L. Trefethen en D. Bau, 1997) bestaan er nog andere belangrijke methoden voor het oplossen van eigenwaardenproblemen. Enkele voorbeelden zijn Jacobi, bisectie, verdeel- en heers en Arnoldi/Lanczos. In dit gedeelte van het practicum bekijken we de Jacobi-methode in meer detail.

De Jacobi-methode steunt op de diagonalisatie van 2×2 symmetrische matrices met behulp van een orthogonale matrix J,

$$J^T \begin{bmatrix} a & d \\ d & b \end{bmatrix} J = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix}, \tag{1}$$

waarbij de orthogonale matrix J gedefinieerd wordt als een rotatie-matrix of een reflectie-matrix. Hier beschouwen we enkel rotatie,

$$J = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}. \tag{2}$$

Opgave 9. Stel dat we de Jacobi-methode willen toepassen op een matrix $A \in \mathbb{R}^{n \times n}$ met n even. Een populaire manier om algoritmen te versnellen is om onafhankelijke processen in parallel, dus tegelijkertijd, uit te voeren. Bespreek hoe je $n/2$ Jacobi-transformaties (1) in parallel zou uitvoeren, waarbij deze transformaties inwerken op de disjuncte rij-/kolomparen $(1, 2), (3, 4), \ldots, (n-1, n)$. Beschrijf of toon visueel waar er na deze operaties nullen geïntroduceerd zijn in de matrix.

Opgave 10. Schrijf een algoritme uit in pseudo-code om met de Jacobi-methode de eigenwaardenontbinding VDV^T van een symmetrische matrix A te berekenen, waarbij D een diagonaalmatrix is die de eigenwaarden bevat, en V een orthogonale matrix met de overeenkomstige eigenvectoren. Bespreek in welke volgorde je de individuele Jacobi-transformaties uitvoert (de volgorde van de posities waar je nullen introduceert). Als stopcriterium mag je eisen dat de verhouding tussen de norm van de niet-diagonaalelementen enerzijds en het grootste diagonaalelement anderzijds kleiner is dan een opgeven tolerantie `tol`. Implementeer dit algoritme als een functie in MATLAB: `function [V,D] = jacobi(A,tol)`. Dit algoritme moet NIET in parallel geprogrammeerd worden!

Opgave 11. Pas de Jacobi-methode toe om alle eigenwaarden en eigenvectoren van `mat1.txt` op Toledo te berekenen. Bekijk de convergentiesnelheid. Komt de experimentele convergentiesnelheid overeen met de theorie? Wat is de rekenkost van de methode voor een willekeurige symmetrische tridiagonale matrix $A \in \mathbb{R}^{m \times m}$?

Praktische richtlijnen

- Dit practicum wordt gemaakt in groepjes van twee.
- Het verslag kan je elektronisch inleveren via mail ten laatste op **22 april 2016** om 15:00 uur.
- Gebruik figuren en tabellen om je bevindingen te verduidelijken. Gebruik logaritmische grafieken (`semilogx`, `semilogy`, `loglog`) waar nodig. Kies een gepaste schaal voor je figuren, vooral als twee verschillende figuren vergeleken moeten worden.
- Bespreek bondig je resultaten en je aanpak in een duidelijke vergezellende tekst. Hou de lengte evenwel onder de 15 pagina's, inclusief tabellen en figuren.
- Lever ook je MATLAB-code (de m-file) voor de gevraagde functies elektronisch in.

Veel succes!

Ben Jeuris (200A 01.160)
`ben.jeuris@cs.kuleuven.be`
`http://toledo.kuleuven.be`