Numerieke Modellering en Benadering: Practicum 1

14 maart 2017

1 QR factorisatie en kleinste kwadraten problemen

In deze sectie onderzoeken we eerst enkele theoretische eigenschappen van de Householder transformatie, waarna deze Householder transformatie getest zal worden in implementaties van de QR factorisatie in Matlab. Deze worden uiteindelijk toegepast voor het modelleren van een systeem aan de hand van een kleinste kwadraten benadering.

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. 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 2. 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^Tb 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 aantal willekeurige vierkante matrices $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$ en bereken $b = Ax$. Variëer hierbij de waarde van n $(= 10, 100, 1000)$ en de conditie van de willeukerige matrix A $(\kappa(A) = 1, 10^4, 10^8)$. De oplossing van beide werkwijzes kan je dan op het eind vergelijken met deze originele x. **Hint:** maak gebruik van de svd om willekeurige matrices met een bepaalde conditie te genereren.

Opgave 3. Op Toledo vind je de bestanden `model.mat` en `validate.mat`. Deze bevatten beide metingen van de in- ($u \in \mathbb{R}^m$) en uitgang ($y \in \mathbb{R}^m$) van eenzelfde onbekend systeem. In deze oefening gaan we de parameters $x \in \mathbb{R}^n$ van volgend model bepalen:

$$\hat{y}_k = x_1\, u_k + x_2\, u_{k-1} + \ldots + x_n\, u_{k-n+1}.$$

Dit wordt ook wel een *voortschrijdend gemiddelde* model met $n-1$ tijdsvertragingen genoemd. Schrijf het probleem van het bepalen van x die de fout,

$$E = \left(\sum_{k=n}^{m} (\hat{y}_k - y_k)^2\right)^{\frac{1}{2}},$$

minimaliseert als de oplossing van een kleinste kwadraten probleem $\|b - A\,x\|_2$ met $A \in \mathbb{R}^{m-n+1\times n}$, $b \in \mathbb{R}^{m-n+1}$. Stel dit probleem op in MATLAB voor de data uit `model.mat` en los het op voor respectievelijk $0, 1, \ldots, 30$ tijdsvertragingen met behulp van één van de twee methodes uit de vorige oefening. Valideer de verkregen modellen aan de hand van de data in `validate.mat`. Maak op basis van dit experiment een keuze voor n en motiveer kort.

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 $A \in \mathbb{R}^{1000 \times 1000}$ 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. In dit gedeelte van het practicum bekijken we de bisectie-methode in meer detail.

Opgave 9. De bisectie-methode steunt op de 'interlacing'-eigenschap van de eigenwaarden van een tridiagonale, symmetrische matrix. Wat is die eigenschap en illustreer voor een zelf gekozen voorbeeld. Geef ook aan welk voorbeeld je gebruikt hebt.

Opgave 10. Schrijf een algoritme uit in pseudo-code om met de bisectie-methode al de eigenwaarden van een symmetrische matrix A in een interval $[a, b)$ te bepalen. Elke eigenwaarde moet bepaald worden met een fout kleiner dan een opgeven tolerantie `tol`. Implementeer dit algoritme als een functie in MATLAB: `function E = bisection(A,a,b,tol)`, met `E` de vector van gevonden eigenwaarden. Toon de correctheid van je implementatie aan door je code toe te passen op enkele 'interessante' testproblemen. Beschrijf duidelijk welke testproblemen je gekozen hebt en waarom.

Praktische richtlijnen

- Dit practicum wordt gemaakt in groepjes van twee.
- Het verslag kan je elektronisch inleveren via mail ten laatste op **21 april 2017** 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!

Daan Camps (200A 02.25)
daan.camps@cs.kuleuven.be
http://toledo.kuleuven.be