Oefenzitting 2: QR-factorisatie en Kleinste Kwadraten

Nico Vervliet[*], *KU Leuven*

6 maart 2018

1 Herhaling algoritmen

Klassieke Gram-Schmidt:

for $j = 1, \ldots, n$ **do**
  $v_j = a_j$
  for $i = 1, \ldots, j-1$ **do**
    $r_{ij} = q_i^* a_j$
    $v_j = v_j - r_{ij} q_i$
  $r_{jj} = \|v_j\|_2$
  $q_j = v_j / r_{jj}$

Gewijzigde Gram-Schmidt:

for $i = 1, \ldots, n$ **do**
  $v_i = a_i$
for $i = 1, \ldots, n$ **do**
  $r_{ii} = \|v_i\|$
  $q_i = v_i / r_{ii}$
  for $j = i+1, \ldots, n$ **do**
    $r_{ij} = q_i^* v_j$
    $v_j = v_j - r_{ij} q_i$

QR via Householder-reflectoren:

for $k = 1, \ldots, n$ **do**
  $x = A_{k:m,k}$
  $v_k = \mathrm{sign}(x_1)\|x\|e_1 + x$
  $v_k = v_k / \|v_k\|$
  $A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^* A_{k:m,k:n})$

[*] Nico.Vervliet@esat.kuleuven.be; Lichte wijziging van versie Hendrik Speleers.

2 Pen en papier

Opgave 1. (a) Hoe ziet de QR-factorisatie van een bandmatrix met bovenbandbreedte q en onderbandbreedte p eruit (zie Figuur 1)? (b) Hoeveel bewerkingen vraagt het berekenen van een Householder respectievelijk een Givens QR-factorisatie? (Schrijf eerst het Givens QR algoritme neer.) Je mag veronderstellen dat $p, q \ll n$. Je kan voor de complexiteit gebruik maken van het rooster in Tabel 1.

$$\begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times \end{bmatrix}$$

Figuur 1: Bandmatrix met bovenbandbreedte $q = 2$ en onderbandbreedte $p = 2$.

Tabel 1: Voorbeeldrooster voor complexiteit algoritmen.

	$+-$	$*/$	$\sqrt{}$
Opbouw transformatie			
Aanpassen matrix			

Opgave 2. Gegeven de QR-factorisatie

$$QR = A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}, \quad a_i \in \mathbb{R}^m$$

Hoe zou je te werk gaan om de QR-factorisaties van $\tilde{A}$ te berekenen in de volgende gevallen?

a) kolom k verwijderd

$$\tilde{A} = \begin{bmatrix} a_1 & \cdots & a_{k-1} & a_{k+1} & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{m\times(n-1)}$$

b) een kolom $z \in \mathbb{R}^m$ toegevoegd

$$\tilde{A} = \begin{bmatrix} a_1 & \cdots & a_k & z & a_{k+1} & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{m\times(n+1)}$$

c) een rij $w^T \in \mathbb{R}^n$ toegevoegd

$$\tilde{A} = \begin{bmatrix} w^T \\ A \end{bmatrix} \in \mathbb{R}^{(m+1)\times n}$$

3 Op de computer

3.1 Householdertransformatie

Opgave 3. Maak de volgende matrix

```
X = rand(5,2); X(2:5,1) = X(2:5,1)*1.e-8
```

Pas de Householderspiegelingen

$$H_k = I - 2u_k u_k^T/(u_k^T u_k) \qquad k = 1, 2$$

toe met

$$\begin{aligned} u_1 &= X(:,1) + \|X(:,1)\|_2 e_1 \\ u_2 &= X(:,1) - \|X(:,1)\|_2 e_1 \end{aligned}$$

Bereken $H_1 X$ en $H_2 X$. Wat zie je? En waarom?

3.2 Kleinste kwadratenproblemen: normaalvergelijkingen versus QR-factorisatie

Opgave 4. Construeer een voorbeeld waarvoor de normaalvergelijkingen falen en Householder niet. Kies een $A \in \mathbb{R}^{3\times 2}$ bovendriehoeks. Werk zoveel mogelijk op papier, dan maak je geen afrondingsfouten.

3.3 Snelle Givenstransformatie

Met snelle Givenstransformaties kan je een (M, D)-representatie van Q berekenen. Meer bepaald geldt voor een matrix $A \in \mathbb{R}^{m\times n}$ dat $M^T A = T$ met T een bovendriehoeksmatrix en $M^T M = D$ diagonaal. Bovendien geldt dat $Q = MD^{-1/2}$ een orthogonale matrix is en

$$Q^T A = D^{-1/2} T \equiv R.$$

Gegeven $x \in \mathbb{R}^2$ en positieve $d \in \mathbb{R}^2$. De routine `fastgivens1.m` berekent een 2×2 snelle Givenstransformatie M zodat de tweede component van $M^T x$ nul is en $M^T D M = D_1$ diagonaal is met $D = \mathrm{diag}(d_1, d_2)$. In de implementatie wordt enkel gebruik gemaakt van de vorm

$$\begin{bmatrix} \beta & 1 \\ 1 & \alpha \end{bmatrix}.$$

De routine `fastgivensQR1.m` maakt gebruik van routine `fastgivens1.m` om een matrix $A \in \mathbb{R}^{m \times n}$ te transformeren naar een bovendriehoeksmatrix T en geeft de bijbehorende matrix D terug.

Opgave 5.

a) Ga de groei na van een element van D (bijvoorbeeld het laatste) voor een willekeurige matrix. Gebruik de grafische mogelijkheden van Matlab om een kwalitatief idee te krijgen. Wat verwacht je? Komt dit overeen met je experimenten?

b) Wat is een manier om de groei te beperken? Pas de Matlab-routines aan zodat ze deze oplossing gebruiken. (Test de correctheid van je implementatie!)

c) Vergelijk de groei van een element van D met deze oplossing ten opzichte van de originele. Verbetering?

d) Kunnen er zich met deze nieuwe implementatie nog problemen voordoen? Wat zou je kunnen doen om ze te verhelpen?

e) Genereer een willekeurige matrix `A` $\in \mathbb{R}^{m \times n}$ en een willekeurige kolomvector `b` $\in \mathbb{R}^m$ en gebruik de snelle Givensfactorisatie om een kleinste kwadratenoplossing te vinden voor het stelsel `A*x = b`.