Prirodoslovno matematički fakultet

Matematički odsjek

Autor**:** Ivan Leverić

**Simetrični QR algoritam**

Uvod i pretpostavke

Problem: Kako izračunati svojstvene vrijednosti matrice A, za koju u našem slučaju pretpostavljamo da je simetrična, odnosno A = A\* ?

Postoji puno metoda koje daju rješenje navedenog problema, a jedan od najpoznatijih i najefikasnijih jest QR algoritam. Klasični QR algoritam za simetričnu matricu A ima sljedeći oblik

Algoritam 1. (Klasični QR algoritam) Dana je simetrična matrica A€Rnxn i ortogonalna matrica U0€Rnxn. Algoritam računa Schurovu formu matrice A.

T0 = U0TAU0

for k = 1,2...

Tk + 1 = UkRk

Tk = RkUk

End

Problem navedenog algoritma jest efikasnost. Naime, za jednu QR iteraciju potrebno je O(n3) operacija – prvo računamo QR faktorizaciju nxn matrice, a onda mnozimo dvije nxn matrice. Srećom, postoji način da se ove poteškoće prevaziđu. Kako, objašnjeno je u sljedećim paragrafima.

Svojstva problema i ciljevi

Poznati rezultati:

* Sve svojstvene vrijednosti simetrične matrice A su realne
* Postoji ortonormalna baza svojstvenih vektora
* Schurova forma T = QTAQ je dijagonalna matrica čiji su dijagonalni elementi svojstvene vrijednosti matrice A
* Cilj: Producirati niz {Qk} ortogonalnih transformacija takvih da su matrice QkTAQk

sve vise dijagonalne

Simetrični QR algoritam

Simetrične QR iteracije mogu se implementirati puno efikasnije ako problem podijelimo u dva dijela:

1. Svođenje matrice A na tridijagonalnu formu
2. Na toj matrici izvodimo poboljšane QR iteracije

Redukcija na tridijagonalnu formu

Prema teoremu o redukciji na Hessenbergovu formu znamo da je moguće matricu A ortogonalnim tranformacijama reducirati na Hessenbergovu formu. U našem slučaju je, zbog simetričnosti matrice A, ta Hessenbergova forma tridijagonalna.

Tu redukciju ćemo izvesti pomoću Housholderovih reflektora

Definicija Za zadani jedinični vektor u€Rnxn, matrica H definirana s

H = H(u) = I – 2uuT naziva se Householderov reflektor.

Neka je zadan vektor x. Tražimo vektor u iz gornje definicije t.d. vrijedi

Hx = ce1 , c€R

Računom dobijamo u = x + sign(x1)||x||2e1 uz ekvivalentnu definiciju

H(u) = I – βuuT , β = 2/uTu

Algoritam 2. (Householderova tridijagonalizacija) Za A€Rnxn simetričnu, računamo tridijagonalnu matricu T = QTAQ, gdje je Q = H1...Hn produkt Householderovih reflektora

* Matlab fileovi Tridiagonalization.m i Householder.m

Što ako nemamo strogu tridijagonalnu formu, to jest ako postoji k takav da ki + 1, i = 0? Tada problem dijelimo u dva dijela, odnosno u dvije matrice T(1:k,1:k) i T(k+1:n,k+1:n).

QR iteracije i tridijagonalne matrice

* QR iteracije čuvaju tridijagonalnu formu, odnosno ako je T = QR simetrična tridijagonalna onda je i RQ = QTTQ simetrična tridijagonalna.
* Koristeći shift T – sI, s€R možemo ubrzati konvergenciju
* QR faktorizaciju možemo računati pomoću niza od n – 1 Givensovih rotacija

Tražimo najbolju vrijednost *shifta.* Naime, ako je shift dovoljno dobar, očekujemo da će element na poziciji (n, n – 1) biti malen. Pokazuje se da je najbolji izbor tzv. Wilkinsonov shift koji je dan formulom µ = an + d – sign(d)sqrt(d2 + bn – 12) , d = (an - 1 – an)/2.

QR iteracije sa shiftom provodimo pomoću tehnike bulge chasing.

Algoritam 3. (QR iteracije s Wilkinsonovim shiftom) Za danu simetričnu tridijagonalnu matricu T€Rnxn, algoritam izvodi transformacije ZTTZ gdje je Z = G1...Gn – 1 produkt Givensovih rotacija sa svojstvom ZT(T - µI) gornje trokutasta i µ Wilkinsonov shift.

* Matlab file Qrshifts.m, te pomoćna funkcija Givens.m

Time smo praktički gotovi. Sada koristeći algoritme 2 i 3 implementiramo finalni simetrični QR algoritam, uzimajući u obzir da nam matrice na kojima provodimo QR iteracije sa pomakom ne moraju biti strogo tridijagonalne. U tom slučaju moramo problem podijeliti, te izvršavati prethodni algoritam na tim, manjim strogo tridijagonalnim matricama. Seciranje na takve matrice napravljeno je u pomoćnoj matlab funkcijji *Find\_p\_and\_q.m* . Napokon imamo:

Algoritam 4. (Simetrični QR algoritam) Za zadanu simetričnu matricu A€Rnxn i danu toleranciju, algoritam računa aproksimaciju simetrične Schurove dekompozicije D = QTAQ.

Primjeri

1. Prvo definiramo matricu D = diag(1, 2, 3, 4). Nakon toga pomoću QR faktorizacije generiramo slučajnu ortogonalnu matricu Q. Sada znamo da je matrica A = QDQT simetrična sa svojstvenim vrijednostima na dijagonali od D.
2. Slično kao u prvom primjeru, opet uzimamo A = QDQT, samo ovoga puta imamo dijagonalnu matricu veličine 100x100 koja ima nakupine istih svojstvenih vrijednosti. Preciznije, D na dijagonali ima 10 različitih svojstvenih vrijednosti kratnosti 10. D = diag(1,2,3,4,5,6,7,8,9,10).
3. Zadnji primjer jest Stijeltjesova matrica reda 100. Za ovaj primjer dodatno mjerimo potrebno vrijeme pomoću matlab tic toc funkcije, i to vrijeme uspoređujemo sa klasičnim QR algoritmom i Jacobijevom metodom s cikličkim pivotiranjem.

Literatura

* G. H. Golub, C. F. Van Loan, ”Matrix computations”, The Johns Hopkins University Press, Baltimore and London, 1989.
* 8. I 9. predavanje “Numerička analiza”