UNIVERZA V LJUBLJANI | FRI Fakulteta za računalništvo in informatiko

QR razcep simetrične tridiagonalne matrike

Numerična matematika

—

1. domača naloga

Avtor: Vito Levstik

Akademsko leto 2024/2025

1 Uvod

Cilj te naloge je implementirati učinkovit QR razcep (simetrične) tridiagonalne matrike z Givensovimi rotacijami in ga uporabiti pri QR iteraciji za izračun lastnih vrednosti in lastnih vektorjev simetrične tridiagonalne matrike.

Da lahko to dosežemo, je najprej potrebno implementirati tipe `Tridiag`, ki predstavlja tridiagonalno matriko, `SimTridiag`, ki predstavlja simetrično tridiagonalno matriko, `Givens`, ki predstavlja matriko Givensovih rotacij in `ZgornjeTridiag`, ki predstavlja matriko z neničelnimi elementi na diagonali in dveh zgornjih diagonalah.

Glavna naloga je tako implementacija funkcije `qr(`T`)`, ki tridiagonalno matriko T razcepi v produkt matrik Q in R, kjer je Q tipa `Givens` in R tipa `ZgornjeTridiag` in implementacija funkcije `eigen(`T`)`, ki izračuna lastne vrednosti in lastne vektorje simetrične tridiagonalne matrike T z uporabo QR iteracije.

2 Implementacija

Podatkovni tip `Tridiag` sprejme tri sezname in sicer seznam spodnje diagonale, seznam glavne diagonale in seznam zgornje diagonale. Tale podatkovni tip smo že implementirali na vajah, zato je koda povzeta iz vaj. Podatkovni tip `SimTridiag` je podobno definiran, le da sprejme dva seznama, seznam glavne diagonale in seznam pod-diagonale, saj sta zgornja in spodnja diagonala enaka. Podatkovni tip `ZgornjeTridiag` je podoben, le da sprejme seznam glavne diagonale in dva seznama zgornjih diagonal.

Do elementov vseh treh matrik lahko dostopamo z indeksi, kot pri običajni matriki, kar smo storili z implementacijo funkcij `getindex()`. Elemente matrik lahko tudi spreminjamo s funkcijo `setindex()`, vendar to lahko naredimo le za elemente, ki so v matriki definirani. V nasprotnem primeru, bo program vrnil napako.

Pomemben podatkovni tip, ki si zasluži bolj podroben opis je `Givens`. Ta podatkovni sprejme seznam 4-dimenzionalnih vektorjev. Prvi dve komponenti sta kosinus in sinus kota rotacije, medtem ko sta tretja in četrta komponenta indeksa stolpcev/vrstic, ki jih rotiramo. Za boljšo predstavo si poglejmo matrično reprezentacijo Givensove rotacije

$$G(c,s,i,j) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

kjer se $c = \cos(\theta)$ in $s = \sin(\theta)$, kjer je θ kot rotacije, pojavita na presečišču i in j-te vrstice in stolpca.

Zadnji korak, ki ga moramo narediti preden se lotimo implementacije `qr(`T`)`, je implementacija funkcije množenja med matrikami tipa `Givens` in `ZgornjeTridiag`. Ko bodisi z leve ali desne pomnožimo matriko tipa `ZgornjeTridiag` z matriko tipa `Givens`, ni nujno da je dobljena matrika tipa `ZgornjeTridiag`. Množenje teh dveh matrik smo zato implementirali tako, da smo matriko tipa `ZgornjeTridiag` najprej pretvorili v polno matriko ter nato izvedli množenje. Če je rezultat množenja tridiagonalna matrika, smo jo pretvorili v tip `Tridiag`, sicer pa smo jo pustili v polni obliki.

Sedaj se lahko lotimo QR razcepa. V splošnem moramo eliminirati elemente pod glavno diagonalo. Ključno opažanje pri razcepu tridiagonalne matrike je, da so elementi potrebni eliminacije samo elementi spodnje diagonale. To nam da motivacijo, da implementiramo funkcijo v časovni zahtevnosti $O(n)$, saj imamo samo $n-1$ elementov za eliminirati. Le-te lahko eliminiramo z uporabo Givensovih rotacij. Konkretno, če želimo eliminirati $a_{i+1,i}$ v matriki A, se ga lahko znebimo tako da pomnožimo A iz leve

z Givensovo rotacijo $G(c, s, i, i+1)$. Pri tem je potrebno izbrati ustrezen c is s. Lahko se preveri, da je potreben kot rotacije enak $\theta = \arctan\left(\frac{a_{i+1,i}}{a_{i,i}}\right)$. Tako dobimo

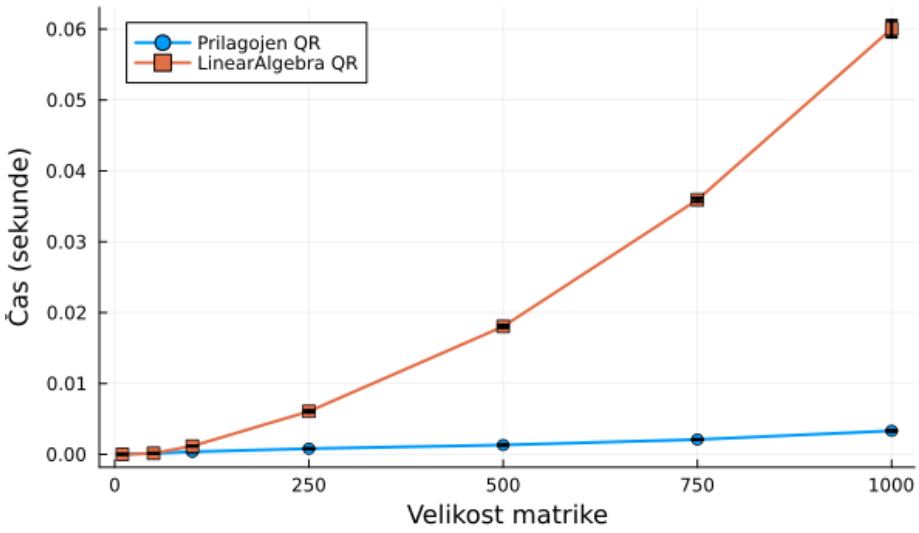
$$c = \cos(\theta) = \frac{a_{i,i}}{\sqrt{a_{i,i}^2 + a_{i+1,i}^2}}, \qquad s = \sin(\theta) = \frac{a_{i+1,i}}{\sqrt{a_{i,i}^2 + a_{i+1,i}^2}}.$$

Ko pomnožimo A z Givensovo rotacijo $G(c, s, i, i+1)$ smo s tem izničili $a_{i+1,i}$, spremenili elemente $a_{i,i}$, $a_{i+1,i+1}$, $a_{i,i+1}$ in $a_{i+1,i+2}$, ter ustvarili nov neničeln element $a_{i,i+2}$ (pod predpostavko da so vsi elementi spodnje diagonale pred $a_{i+1,i}$ že eliminirani). Ta novo nastali element je element zgornje druge nad-diagonale. To pomeni, da celoten algoritem QR razcepa tridiagonalne matrike operira le na spodnji diagonali, glavni diagonali in dveh zgornjih diagonalah. Algoritem je tako preposto enojna zanka po vseh elementih spodnje diagonale, kjer za vsak element izračunamo ustrezno Givensovo rotacijo, posodobimo elemente po množenju in dodamo Givensovo rotacijo v seznam rotacij. Končen rezultat je tako matrika R tipa `ZgornjeTridiag` in matrika Q tipa `Givens`, ki je produkt vseh Givensovih rotacij, ki smo jih uporabili pri eliminaciji.

Lastne vrednosti in lastne vektornje simetrične tridiagonalne matrike dobimo z uporabo QR iteracije. Algoritem QR iteracije je sledeč. Simetrično matriko T razcepimo v produkt $T = QR$, z že implementirano funkcijo `qr(`T`)`. Pri tem si shranimo matriko Q in nato izračunamo $T = RQ$. Ta postopek ponavljamo poljubno število korakov. Elementi na diagonali matrike T konvergirajo v lastne vrednosti, medtem ko so pripadajoči lastni vektorji stolpci matrike dobljene kot produkt vseh matrik Q.

3 Rezultati

Primerjali smo QR razcep simetrične tridiagonalne matrike različnih velikosti z QR razcepom iz knjižice `LinearAlgebra`, ki operira na polni matriki. Rezultati so prizakani na sliki 1.



Slika 1: Primerjava našega QR razcepa simetrične tridiagonalne matrike z QR razcepom iz knjižice LinearAlgebra.

Vidimo, da je naš algoritem QR razcepa bistveno hitrejši pri večjih matrikah. Razlog je v tem, da je naš algoritem optimiziran za tridiagonalne matrike in deluje v $O(n)$, medtem ko QR razcep iz knjižice `LinearAlgebra` deluje za poljubne matrike, kar pomeni, da deluje v $O(n^3)$. Po sliki sodeč naš algoritem resnično deluje v linearnem času, medtem ko QR razcep iz knjižice `LinearAlgebra` izgleda da deluje v bolj kompleksnem času (verjetno $O(n^3)$).