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 preprosto 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.

Pri vseh podatkovnih tipih smo pazili, da program preveri, ali so velikosti seznamov pravilne (npr. če je dolžina glavne diagonale n, potem mora biti dolžina zgornje diagonale $n-1$). 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 je potrebno paziti, da spreminjamo le 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.

3 Rezultati