

Znanstveno računanje 1 (akad. god. 2019./2020.)

Završni projektni zadaci

Upute ili “pravila igre”:

- Treba izabrati i riješiti 1 (**jedan**) od ponuđenih zadataka. Svaki student bira svoj zadatak, po principu “tko prije izabere — njegov je”.
- Rješenje zadatka je **program** i dobiveni **rezultati** programa, s popratnom dokumentacijom i interpretacijom rezultata. Rezultati se predaju u obliku “kraćeg seminarskog rada”.
- Program može biti napisan u programskom jeziku C, u nekom drugom standardnom programskom jeziku (Fortran, Pascal), ili nekom standardnom alatu za numeričko računanje (Matlab, Octave). Dozvoljeno je koristiti posebne programske pakete za vizualizaciju postupka i rezultata.
- Zadnji rok za predaju završnog projektnog zadatka je **ponedjeljak, 17. veljače 2020.**

Kodeks ponašanja: Ista ili vrlo slična rješenja se **poništavaju**.

Napomena. Zadaci su **namjerno** sažeto formulirani, tako da propisuju samo nužni dio posla. Sve ostalo: planiranje ulaza, izlaza, implementacije, testiranja i sl., je “slobodno” i nagrađuje se kao dio rješenja. Dozvoljeno je koristiti standardne programske biblioteke (poput BLAS, LAPACK i sl.). Za rješenje smijete koristiti i dodatnu literaturu (“sve što nađete”), samo propisno citirajte. Također, dozvoljeno je napraviti i više od onog što se traži, — na primjer, usporediti nekoliko metoda za rješenje problema, i sl.

Kontakt: Ako negdje “zapnete”, imate pitanja, trebate savjet, želite saznati i/ili napraviti i više od onog što se traži — **slobodno** se javite.

- Najbolje e-mailom, pa ćemo se dogovoriti za daljnji kontakt (po potrebi), ili me potražite u doba konzultacija.

Isto vrijedi i za dogovor o terminu predaje.

Zadatak 1. Iterativne metode za linearne sustave — blok–Gauss–Seidel.

Promatramo sustav linearnih jednadžbi u kojem su matrica i vektori podijeljeni na blokove na sljedeći način

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix},$$

s tim da su dijagonalni blokovi A i D regularne kvadratne matrice (ne moraju biti istog reda).

Onda možemo definirati **blok–Gauss–Seidelovu** iterativnu metodu po ugledu na običnu Gauss–Seidelovu iterativnu metodu, tako da jedna iteracija ima sljedeća 2 koraka:

1. Uzmemo neku početnu aproksimaciju za y i riješimo linearni sustav za x iz prve blok–jednadžbe $Ax = b - By$.
2. Dobiveni x uvrstimo u drugu blok–jednadžbu i riješimo linearni sustav $Dy = c - Cx$. Dobiveni y je nova aproksimacija za y u sljedećoj iteraciji.

Napišite ovu iterativnu metodu u standardnom matricnom obliku i nađite matricu koja određuje konvergenciju metode. U kojim slučajevima ima smisla koristiti ovu metodu?

Generalizirajte ovaj postupak na blok tridijagonalne matrice oblika

$$\begin{bmatrix} D_1 & U_1 & & & \\ L_1 & \ddots & \ddots & & \\ & \ddots & \ddots & U_{m-1} & \\ & & L_{m-1} & D_m & \end{bmatrix},$$

gdje su D_i regularne kvadratne matrice (ne nužno istih redova).

Napišite program za blok–Gauss–Seidelovu iterativnu metodu za linearne sustave s blok tridijagonalnim matricama. Testirajte na matricama koje se dobivaju diskretizacijom Poissonove jednadžbe na jediničnom kvadratu. Uporedite brzinu konvergencije obične i blok Gauss–Seidelove metode na nekoliko primjera.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 2. Iterativne metode za linearne sustave — tridijagonalni blok.

Iterativnu metodu za rješavanje sustava linearnih jednadžbi $Ax = b$ možemo konstruirati podjelom matrice A u 3 bloka: T je tridijagonalni dio od A , U je dio od A iznad T , a L je dio od A ispod T , tako da je $A = T + L + U$. Sustav $Ax = b$ možemo napisati u obliku

$$Tx = -(L + U)x + b.$$

Pretpostavimo da je T regularna matrica. Iterativnu metodu dobivamo tako da krenemo od neke početne aproksimacije za vektor x , a jedna iteracija ima sljedeća 2 koraka:

1. Trenutnu aproksimaciju za x uvrstimo na desnu stranu gornje jednadžbe.
2. Riješimo dobiveni linearni sustav s tridijagonalnom matricom T i tako dobijemo novu aproksimaciju za x .

Dodatno, nakon ova dva koraka, u svakoj iteraciji možemo još napraviti relaksaciju s nekim parametrom ω , što odgovara ekstrapolaciji promjene vektora x za faktor ω .

Napišite obje varijante ove iterativne metode u standardnom matricnom obliku i nađite matrice koje određuju konvergenciju ovih varijanti. U kojim slučajevima ima smisla koristiti ovu metodu? Nađite broj operacija za jednu iteraciju (u obje varijante) i usporedite ga s brojem operacija u standardnim iterativnim metodama (Jacobi, Gauss–Seidel, SOR).

Generalizirajte ovaj postupak na blok tridijagonalne matrice oblika

$$\begin{bmatrix} T_1 & U_1 & & & \\ L_1 & \ddots & \ddots & & \\ & \ddots & \ddots & U_{m-1} & \\ & & L_{m-1} & T_m & \end{bmatrix},$$

gdje su T_i regularne kvadratne tridijagonalne matrice (ne nužno istih redova). U jednoj iteraciji blok–metode, redom se rješavaju linearni sustavi s matricama T_i , za $i = 1, \dots, m$ i tako dobivaju novi “poboljšani” blokovi u vektoru x .

Napišite program za razne varijante ove iterativne metode za linearne sustave s blok tridijagonalnim matricama. Testirajte na matricama koje se dobivaju diskretizacijom Poissonove jednadžbe na jediničnom kvadratu. Usporedite brzinu konvergencije ovih i standardnih iterativnih metoda na nekoliko primjera. Ako realizirate i relaksacijsku varijantu, probajte naći dobar izbor parametra ω .

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 3. Iterativne metode za linearne sustave — kombinirana metoda za skalirane sustave.

Promatramo sustav linearnih jednadžbi u kojem su matrica i vektori podijeljeni na blokove na sljedeći način

$$\begin{bmatrix} A & E \\ H & B \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}.$$

s tim da su dijagonalni blokovi A i B regularne kvadratne matrice, gdje je A reda n , a B reda m . Dodatno, pretpostavimo da su izvandijagonalni blokovi E i H “mali” po veličini elemenata (ili po normi) u usporedbi s A i B , tako da se ovaj sustav “skoro” raspada u dva dijela.

Neka je n mnogo manji od m , tj. A je mnogo manja matrica od B , i neka je B “jako” dijagonalno dominantna. Onda ima smisla definirati kombiniranu **blok** direktno–iterativnu metodu, u kojoj jedna iteracija ima sljedeća 2 koraka:

1. Uzmemo neku početnu aproksimaciju za y i nekom direktnom metodom riješimo linearni sustav za x iz prve blok–jednadžbe $Ax = b - Ey$, na pr. Gaussovimi eliminacijama (s parcijalnim pivotiranjem).
2. Dobiveni x uvrstimo u drugu blok–jednadžbu i dobijemo linearni sustav $By = c - Hx$ za y . U ovom sustavu napravimo jednu iteraciju Gauss–Seidelove metode (ili $SOR(\omega)$ metode). Time dobivamo poboljšanu aproksimaciju za y koju koristimo u sljedećoj iteraciji.

Ova dva koraka transformiraju par vektora $(x^{(k)}, y^{(k)})$ u novu iteraciju $(x^{(k+1)}, y^{(k+1)})$ tog para.

Napišite ovu iterativnu metodu u standardnom matričnom obliku i nađite matricu koja određuje konvergenciju metode.

Napišite program za ovu kombiniranu direktno–iterativnu metodu za linearne sustave s odgovarajućom strukturom matrica. Testirajte na nekom izabranom skupu matrica i usporedite rezultate s nekim drugim metodama (točnost, brzina).

Testirajte program i na matricama koje se dobivaju diskretizacijom Poissonove jednadžbe na jediničnom kvadratu uz “crveno–crnu” numeraciju točaka. Jesu li te matrice dobar primjer za ovu metodu?

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 4. Iterativne metode za linearne sustave — optimalni parametar za SOR.

Sustav linearnih jednadžbi $Ax = b$ iterativno rješavamo $\text{SOR}(\omega)$ metodom. Za neke klase matrica A postoji teorija na osnovu koje možemo analitički odrediti optimalnu vrijednost parametra ω koji daje najbržu konvergenciju metode (najmanji spektralni radijus). S druge strane, za mnoge praktične probleme optimalni parametar ω treba odrediti eksperimentalno.

Napišite program koji za zadanu simetričnu matricu A eksperimentalno nalazi približnu optimalnu vrijednost parametra ω za SOR metodu. Koraci za konstrukciju algoritma:

- Uzmite da je vektor $x = (1, 1, \dots, 1)^T$ egzaktno rješenje sustava $Ax = b$ i izračunajte pripadnu desnu stranu b .
- Za startnu aproksimaciju uzmite $x^{(0)} = 0$, tako da je greška ove aproksimacije $\|e^{(0)}\|_\infty = 1$.
- Za razne vrijednosti od $\omega \in [1, 2]$ izračunajte približno rješenje sustava $\text{SOR}(\omega)$ metodom i procijenite brzinu konvergencije (detaljniji opis je malo niže). Krenite od ekvidistantne mreže za ω s korakom 0.1, a zatim profinjavajte tamo gdje je konvergencija brža. Osim obične trisekcije (ili sličnog profinjavanja na više dijelova), možete koristiti i numeričke tehnike minimizacije.
- Za procjenu brzine konvergencije generirajte niz od N aproksimacija (iteracija metode) i izračunajte grešku $\|e^{(N)}\|_\infty$ zadnje aproksimacije. Možete uzeti $N = 100$ za prvi test.
- Broj $p(\omega) = (\|e^{(N)}\|_\infty)^{1/N}$ možemo uzeti kao “prosječnu” brzinu konvergencije ovih iteracija, jer se greška smanjuje za taj faktor u svakoj iteraciji. Naravno, manji faktor znači veću brzinu, tj. traži se točka minimuma za $p(\omega)$, po raznim vrijednostima ω .

Testirajte ovaj program za matrice raznih redova n . Zgodni primjeri su matrice:

- A_1 je tridijagonalna, s elementima 2 na dijagonali i -1 iznad i ispod dijagonale (diskretni Poisson u 1D),
- A_2 je peterodijagonalna, s elementima 2 na dijagonali, -1 iznad i ispod dijagonale, i $1/2$ na sljedećoj donjoj i gornjoj dijagonali.

Testirajte program i na matricama koje se dobivaju diskretizacijom Poissonove jednadžbe na jediničnom kvadratu uz običnu i “crveno–crnu” numeraciju točaka.

Komentirajte trajanje ovog postupka u ovisnosti o numeričkoj točnosti nađenih optimalnih vrijednosti za ω .

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 5. Iterativne metode za linearne sustave — usporedba Jacobi, Gauss–Seidel.

Sustav linearnih jednadžbi $Ax = b$ rješavamo Jacobijevom i Gauss–Seidelovom iterativnom metodom. Prema konstrukciji ovih iterativnih metoda, brzina konvergencije Jacobijeve metode ovisi o tome koliko je matrica A “blizu” dijagonalne, a brzina Gauss–Seidelove o tome koliko je A “blizu” donje trokutaste matrice.

Napišite program za eksperimentalnu usporedbu ovih dviju metoda na posebno generiranim slučajnim matricama A raznih redova n . Svaka pojedina matrica A generira se u obliku $A = L + U$, gdje je L donja trokutasta (s dijagonalom), a U strogo gornja trokutasta matrica (bez dijagonale). Algoritam za generiranje ovih matrica kontrolira se s dva parametra MODE i σ , tako da budu zadovoljena sljedeće relacije:

- Ako je $\text{MODE} = 1$, onda mora vrijediti

$$\sum_{i,j=1}^n |l_{ij}| = \sigma \cdot \sum_{i,j=1}^n |u_{ij}|.$$

- Ako je $\text{MODE} = 2$, onda mora vrijediti

$$\sum_{j=1}^i |l_{ij}| = \sigma \cdot \sum_{j=i+1}^n |u_{ij}|, \quad \text{za } i = 1, \dots, n.$$

- Ako je $\text{MODE} = 3$, onda mora vrijediti

$$|l_{ij}| = \sigma \cdot |u_{ji}|, \quad \text{za sve } i \neq j.$$

Do na ove relacije, elementi matrica L i U su slučajno generirani. Ako uzmemo $\sigma \geq 1$, onda L “dominira” nad U na tri različita načina. Po želji, smijete dodati još neki MODE za drugačiju kontrolu “dominacije”.

Program treba provjeriti konvergenciju navedenih iterativnih metoda i usporediti brzinu konvergencije (ako konvergiraju), u funkciji parametara MODE i σ . Upute za konstrukciju algoritma za testiranje pojedine metode na zadanoj matrici A :

- Uzmite da je vektor $x = (1, 1, \dots, 1)^T$ egzaktno rješenje sustava $Ax = b$ i izračunajte pripadnu desnu stranu b .
- Za startnu aproksimaciju uzmite $x^{(0)} = 0$, tako da je greška ove aproksimacije $\|e^{(0)}\|_\infty = 1$.
- Za procjenu brzine konvergencije generirajte niz od N aproksimacija (iteracija metode) i izračunajte grešku $\|e^{(N)}\|_\infty$ zadnje aproksimacije. Možete uzeti $N \leq 100$, ali kontrolirajte ponašanje u procesu iteracija, jer metoda ne mora konvergirati.
- Broj $p = (\|e^{(N)}\|_\infty)^{1/N}$ možemo uzeti kao “prosječnu” brzinu konvergencije ovih iteracija, jer se greška smanjuje (ili povećava) za taj faktor u svakoj iteraciji. Naravno, manji faktor znači veću brzinu, ako metoda konvergira.

Planirajte eksperimentalno testiranje efekta parametara MODE i σ na konvergenciju navedenih metoda. Izaberite nekoliko (razumno malih) vrijednosti za n i neki raspon vrijednosti za σ , za svaku vrijednost MODE , i provedite eksperiment. Diskutirajte dobivene rezultate.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 6. Iterativne metode za linearne sustave — usmjereni Gauss–Seidel.

Sustav linearnih jednadžbi $Ax = b$ rješavamo Gauss–Seidelovom iterativnom metodom. U običnoj (standardnoj) varijanti metode, u svakoj iteraciji imamo n koraka i

- prolazimo **redom** kroz sve jednadžbe sustava, onim redom kojim su zadane, i popravljamo jednu po jednu komponentu rješenja (opet, redom, od prve prema zadnjoj).

Cijela iteracija ima oblik

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n.$$

Uočimo da svaki korak **poništava** i -tu komponentu trenutnog reziduala $r = Ax - b$, gdje je x trenutno stanje komponenti “radnog” vektora x .

U **usmjereoju** varijanti Gauss-Seidelove metode, u svakom sljedećem **koraku**

- biramo onu jednadžbu i koja tog trena ima **najveću** komponentu reziduala (po apsolutnoj vrijednosti), obzirom na “radni” vektor x .

Zatim, poništavamo tu komponentu reziduala **popravljanjem** odgovarajuće komponente radnog vektora x , kao u i -tom koraku obične Gauss-Seidelove metode.

Za opće probleme s punom matricom A , preskupo je računati **sve** komponente reziduala nakon **svakog** pojedinog koraka usmjerene metode. Međutim, za dovoljno šuplje matrice s posebnom strukturom, to se može isplatiti.

Opišite takve situacije u kojima se **isplati** ponovno računati rezidual nakon svakog koraka iterativne metode. Posebno, sastavite efikasne pripadne algoritme za linearne sustave koji nastaju standardnom diskretizacijom Poissonove jednadžbe u jednoj i dvije dimenzije (na jediničnom segmentu i kvadratu). Opišite kako se efikasno može voditi evidencija o tome koja jednadžba daje najveći rezidual, što je ključno za izbor sljedećeg koraka, Uputa: koristite hrpu (“heap”).

Napišite program za eksperimentalnu usporedbu obične i usmjerene Gauss-Seidelove metode za zadani linearni sustav $Ax = b$. Varijante programa:

- Napravite program za “opće” matrice A . Za testiranje možete koristiti matrice A koje se generiraju kao u Zadatku 5. Također, uporedbu brzine konvergencije obje metode možete napraviti na isti način kao u Zadatku 5. Rezidual možete računati direktnim algoritmom, bez posebnih “trikova” za specijalne matrice.
- Umjesto toga, možete napraviti usporedbu na matricama koje nastaju diskretizacijom Poissonove jednadžbe u jednoj i dvije dimenzije. U tom slučaju, probajte što efikasnije implementirati računanje reziduala i vođenje “evidencije” o tome koja jednadžba daje najveći rezidual korištenjem hrpe (“heap”).

Diskutirajte dobitak u brzini konvergencije kod usmjerene metode i pokušajte odrediti koliko dodatnog posla se smije potrošiti na “usmjeravanje”, a da efikasnost bude veća nego kod obične Gauss–Seidelove metode.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 7. Multigrid metoda za Poissonovu jednadžbu u 1D, 2D.

Promatramo sustave linearnih jednadžbi koji nastaju diskretizacijom Poissonove jednadžbe u jednoj i dvije dimenzije (na jediničnom segmentu, odnosno, na jediničnom kvadratu).

Napišite program za rješavanje ovih sustava Multigrid metodom. Napravite posebnu funkciju za realizaciju jednog V-ciklusa (zadane “dubine”), a za rješavanje sustava iskoristite puni V-ciklus, sve dok ne dobijete željenu točnost. Testirajte s raznim desnim stranama sustava i usporedite rezultate s nekom drugom metodom za rješavanje istih sustava.

U funkciji za jedan V-ciklus Multigrid metode treba realizirati 3 operatora. Za operator rješenja $S(i)$ na oba mjesta u funkciji uzmite jedan korak JOR metode s optimalnim parametrom $w = 2/3$. Po želji, smijete probati i razne druge varijante operatora rješenja (na jednom ili oba mjesta u funkciji) i usporediti dobivene rezultate u pogledu točnosti i efikasnosti (broja operacija). Ideje za varijacije:

- Umjesto jednog koraka iterativne metode, koristiti neki zadani broj koraka metode, s tim da brojevi koraka mogu biti različiti na raznim mjestima.
- Probati i druge težine.
- Probati i drugačije metode, poput Gauss–Seidelove.

Za operatore restrikcije i interpolacije iskoristite usrednjavanje vrijednosti preko susjednih točaka u mreži diskretizacije. Detaljniji opis usrednjavanja dan je u literaturi.

Literatura: predavanja “Multigrid” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera), i Nela Bosner, “Iterativne metode za rješavanje linearnih sustava”, magistarski rad (dostupno na mojoj web stranici).

Zadatak 8. Iterativne metode za linearne sustave — MINRES metoda

Metoda konjugiranih smjerova za simetrične pozitivno definitne matrice A u svakoj iteraciji bira skalar α_k takav da za $x_{k+1} = x_k + \alpha_k d_k$ vektor greške $e_{k+1} = x - x_{k+1}$, $x = A^{-1}b$ ima minimalnu A -normu $\|e_{k+1}\|_A = \sqrt{e_{k+1}^T A e_{k+1}}$, tj.

$$\|e_{k+1}\|_A = \min\{\|e\|_A : e = e_k - \alpha d_k, \alpha \in \mathbb{R}\}.$$

Pri tome smjerovi traganja d_k biraju se tako da je $d_0 = r_0$ i da su međusobno A -ortogonalni ili *konjugirani*:

$$d_j^T A d_i = 0, \quad i \neq j.$$

Vektor d_{k+1} dobiva se iz vektora reziduala $r_{k+1} = r_k - \alpha_k A d_k$ primjenom Gram–Schmidtove metode A -ortogonalnosti, odakle dobivamo da je $d_{k+1} = r_{k+1} + \beta_{k+1} d_k$.

Vaš zadatak je da definirate sličnu metodu za opću simetričnu matricu A pod imenom MINRES. Ona također bira sljedeću aproksimaciju rješenja kao i metoda konjugiranih smjerova: $x_{k+1} = x_k + \alpha_k d_k$, samo što se skalar α_k bira tako da pripadajući rezidual $r_{k+1} = r_k - \alpha_k A d_k$ ima minimalnu euklidsku normu. U ovom slučaju smjerovi traganja su A^2 -ortogonalni. Također se dobivaju iz vektora reziduala primjenom Gram–Schmidtove metode A^2 -ortogonalnosti, pri čemu je $d_0 = r_0$, te kao i kod metode konjugiranih smjerova zadovoljavaju jednostavnu rekurziju $d_{k+1} = r_{k+1} + \beta_{k+1} d_k$. Dakle, jedina je razlika u izboru skalara α_k i β_{k+1} .

Napišite postupak pomoću kojeg se dolazi do skalara α_k i β_{k+1} u MINRES metodi. Postupak mora predstavljati valjan matematički dokaz vaše tvrdnje. Napišite potprogram koji implementira MINRES metodu i primijenite je na raznim simetričnim matricama. Brzinu konvergencije MINRES metode primijenjene na indefinitnim matricama usporedite sa brzinom konvergencije GMRES metode. Brzinu konvergencije MINRES metode primijenjene na pozitivno definitnim matricama usporedite sa brzinom konvergencije metode konjugiranih smjerova. Za svaku matricu ispišite njene svojstvene vrijednosti. Kako raspored svojstvenih vrijednosti utječe na brzinu konvergencije MINRES metode? Da li se može dogoditi da se metoda prekine bez da smo došli do rješenja?

Literatura: Nela Bosner, “Iterativne metode za rješavanje linearnih sustava”, magistarski rad (dostupno na mojoj web stranici)

Zadatak 9. Iterativne metode za linearne sustave — metoda bikonjugiranih gradijenata (BCG)

Budući da GMRES metoda za nehermitske probleme svakom iteracijom povećava količinu posla kojeg se treba obaviti i memorije koju treba zauzeti, pojavila se potreba za razvojem drugačijih metoda koje bi zahtijevale fiksnu količinu posla i memorije po iteraciji. Takve metode baziraju se na dvostranom Lanczosovom algoritmu.

Algoritam 1 (Dvostrani Lanczosov algoritam)

v_1 sa $\|v_1\|_2 = 1$, i w_1 sa $w_1^T v_1 = 1$ zadani;

$\psi_0 = 0$; $\xi_0 = 0$;

$v_0 = 0$; $w_0 = 0$;

$j = 1$;

while (!kriterij_zaustavljanja){

$\phi_j = w_j^T A v_j$;

$\tilde{v}_{j+1} = A v_j - \phi_j v_j - \psi_{j-1} v_{j-1}$;

$\tilde{w}_{j+1} = A^T w_j - \phi_j w_j - \xi_{j-1} w_{j-1}$;

$\xi_j = \|\tilde{v}_{j+1}\|_2$;

$v_{j+1} = \frac{\tilde{v}_{j+1}}{\xi_j}$;

$\psi_j = \tilde{w}_{j+1}^T v_{j+1}$;

if ($\psi_j == 0$)

stani;

else

$w_{j+1} = \frac{\tilde{w}_{j+1}}{\psi_j}$;

$j = j + 1$;

}

Za razliku od Arnoldijevog algoritma, ova metoda konstruira *biortogonalne* baze za dva Krylovljeva potprostora, jednog definiranog za matricu A

$$\mathcal{K}_k(A, v_1) = \text{span}\{v_1, A v_1, \dots, A^{k-1} v_1\},$$

i drugog definiranog za matricu A^T

$$\mathcal{K}_k(A^T, w_1) = \text{span}\{w_1, A^T w_1, \dots, (A^T)^{k-1} w_1\}.$$

Biortogonalnost se očituje u svojstvu

$$w_j^T v_i = 0, \quad \text{za } i \neq j.$$

Neka je V_k matrica sa stupcima v_1, \dots, v_k i W_k matrica sa stupcima w_1, \dots, w_k , tada se par rekurzija iz gornjeg algoritma može napisati u matičnom obliku kao

$$\begin{aligned} A V_k &= V_k T_k + \xi_k v_{k+1} e_k^T = V_{k+1} T_{k+1,k}, \\ A^T W_k &= W_k T_k^T + \psi_k w_{k+1} e_k^T = W_{k+1} \hat{T}_{k+1,k}, \end{aligned}$$

gdje je $e_k = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T$ jedinični vektor s 1 u k -toj komponenti, a T_k $k \times k$ tridijagonalna matrica koeficijenata rekurzije

$$T_k = \begin{bmatrix} \phi_1 & \psi_1 & & & \\ \xi_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \psi_{k-1} & \\ & & \xi_{k-1} & \phi_k & \end{bmatrix}.$$

$(k + 1) \times k$ matrice $T_{k+1,k}$ i $\hat{T}_{k+1,k}$ imaju T_k odnosno T_k^T kao gornji $k \times k$ blok, i zadnji red popunjen s nulama osim $(k + 1, k)$ elementa koji je jednak ξ_k odnosno ψ_k . Svojstvo biortogonalnosti matricno se može napisati kao

$$W_k^T V_k = I.$$

Metode bazirane na dvostranom Lanczosovom algoritmu ipak imaju neke nedostatke jer mogu zakazati. Jedna takva metoda je i BCG, za koju je $v_1 = r_0 / \|r_0\|_2$ i $w_1 = \hat{r}_0 / v_1^T \hat{r}_0$. Za nju ćemo kao i kod GMRES-a uzeti x_k oblika

$$x_k = x_0 + V_k y_k.$$

Jedan od prirodnih izbora vektora y_k u k -toj iteraciji metode bio bi takav da odgovarajući $r_k = r_0 - AV_k y_k$ bude ortogonalan na $\mathcal{K}_k(A^T, w_1)$, odnosno na vektore w_1, \dots, w_k . To nas vodi do jednakosti

$$W_k^* r_k = W_k^* r_0 - W_k^* AV_k y_k = 0.$$

Zbog biortogonalnosti Lanczosovih vektora i Lanczosove rekurzije slijedi da je $W_k^* r_0 = \beta \xi_1$, sa $\beta = \|r_0\|_2$, i $W_k^* AV_k = T_k$, stoga jednadžba za y_k izgleda ovako,

$$T_k y_k = \beta \xi_1.$$

Algoritam se dalje pojednostavljuje ma oblik sličan algoritmu konjugiranih gradijenata.

Algoritam 2 (Metoda bikonjugiranih gradijenata (BCG))

x_0 zadan;

$d_0 = r_0 = b - Ax_0$;

Izaberi \hat{r}_0 takav da je $\hat{r}_0^T r_0 \neq 0$;

$p_0 = r_0$;

$\hat{p}_0 = \hat{r}_0$;

$k = 0$;

while (!kriterij_zaustavljanja){

$$\alpha_k = \frac{\hat{r}_k^T r_k}{\hat{p}_k^T A p_k};$$

$$x_{k+1} = x_k + \alpha_k p_k;$$

$$r_{k+1} = r_k - \alpha_k A p_k;$$

$$\hat{r}_{k+1} = \hat{r}_k - \alpha_k A^T \hat{p}_k;$$

$$\beta_{k+1} = \frac{\hat{r}_{k+1}^T r_{k+1}}{\hat{r}_k^T r_k};$$

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k;$$

$$\hat{p}_{k+1} = \hat{r}_{k+1} + \beta_{k+1} \hat{p}_k;$$

$$k = k + 1;$$

}

U kojim slučajevima bi moglo doći do sloma ove metode? Koji uvjet nam garantira da će metoda završiti za $m \leq n$ iteracija pri čemu je n dimenzija sustava?

Napišite potprograme koji implementiraju dvostrani Lanczosov algoritam i BCG metodu i primijenite ih na raznim matricama. Najprije izvršite potprogram za BCG metodu, koji treba završiti za $m \leq n$ iteracija bilo zbog sloma, bilo zbog konvergencije. Zatim izračunajte T_m , V_m i W_m pomoću dvostranog Lanczosovog algoritma. Provjerite da je $W_m^T AV_m = T_m$. Koliki je \tilde{v}_{m+1} , i da li nam on nešto govori o konvergenciji BCG metode?

Usporedite brzinu konvergencije BCG metode s brzinom konvergencije GMRES metode. Posebno pronađite barem jedan primjer u kojem dolazi do sloma metode, i barem jedan primjer u kojem je konvergencija dobra kao i kod GMRES metode. Prokomentirajte sve dobivene rezultate i brzine konvergencije.

Literatura: Nela Bosner, “Iterativne metode za rješavanje linearnih sustava”, magistarski rad (dostupno na mojoj web stranici)

Zadatak 10. Linearni sustav — Laplaceova jednadžba na polukrugu.

Zadana je polukružna ploča radijusa R . Ravni dio ruba (dijametar kroz centar kruga) ima temperaturu 0°C , dok se zakrivljeni (polukružni) dio ruba drži na konstantnoj temperaturi od $T^\circ\text{C}$.

Stacionarna raspodjela temperature u te ploče određena je Laplaceovom jednadžbom

$$\Delta u = 0,$$

unutar ploče, uz zadane rubne uvjete. Geometrija problema sugerira da funkciju temperature u treba pisati u polarnim koordinatama (r, φ) obzirom na centar polukruga, tj. $u = u(r, \varphi)$. U polarnim koordinatama Laplaceov operator ima sljedeći oblik

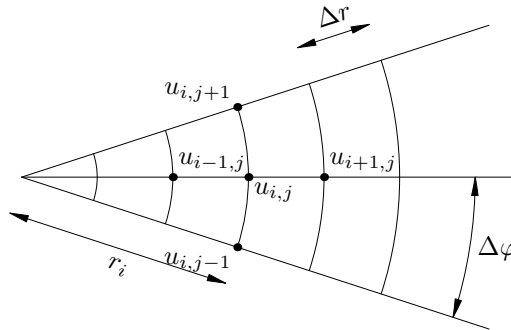
$$\Delta u = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \varphi^2} = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \varphi^2}.$$

Ovu jednadžbu možemo diskretizirati tako da uzmemo “ekvidistantnu” mrežu u svakoj od koordinata s koracima $\Delta r = R/(M+1)$ i $\Delta \varphi = \pi/(N+1)$, gdje su M i N unaprijed zadani prirodni brojevi. Čvorovi mreže su točke

$$(r_i, \varphi_j) = (i \cdot \Delta r, j \cdot \Delta \varphi), \quad i = 0, \dots, M+1, \quad j = 0, \dots, N+1.$$

Uočite da za $i = 0$ imamo samo jednu točku $(0, 0)$, neovisno o j . Uvedimo još skraćenu oznaku za vrijednosti funkcije u u čvorovima mreže $u_{i,j} = u(r_i, \varphi_j)$.

Laplaceov operator u polarnim koordinatama diskretiziramo u unutarnjoj točki mreže (r_i, φ_j) na standardni način — simetričnim razlikama, korištenjem 5 točaka, kao na sljedećoj slici.



Dobivena diskretizirana jednadžba, nakon množenja s $(\Delta r)^2$, ima oblik

$$\left(1 - \frac{\Delta r}{2r_i}\right) u_{i-1,j} - 2 \left[1 + \left(\frac{\Delta r}{r_i \Delta \varphi}\right)^2\right] u_{i,j} + \left(1 + \frac{\Delta r}{2r_i}\right) u_{i+1,j} + \left(\frac{\Delta r}{r_i \Delta \varphi}\right)^2 u_{i,j-1} + \left(\frac{\Delta r}{r_i \Delta \varphi}\right)^2 u_{i,j+1} = 0$$

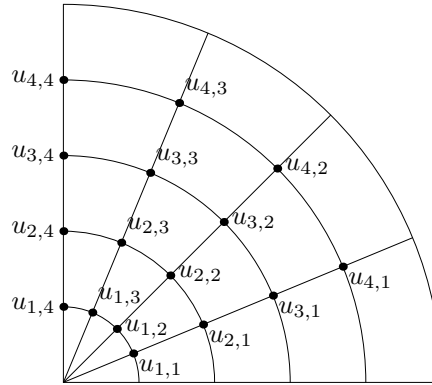
Nakon uvrštavanja rubnih uvjeta $u_{i,0} = u_{i,N+1} = 0$, za $i = 0, \dots, M+1$, i $u_{M+1,j} = T$, za $i = 1, \dots, N$, dobivamo linearni sustav od $M \cdot N$ jednadžbi. Ako uzmemo leksikografsku “pravokutnu” numeraciju čvorova, sustav je sličan onom kod Poissonove jednadžbe na kvadratu, samo što dijagonalni blokovi **nisu** simetrični.

Napišite program koji rješava ovaj sustav jednadžbi, za zadane parametre diskretizacije M i N . Sustav možete riješiti direktno (Gausovim eliminacijama), ili nekom iterativnom metodom (Jacobi, Gauss–Seidel, i sl.), samo budite oprezni s konvergencijom metode.

Za osnovno testiranje pretpostavite da je $R = 1$ m, $T = 100^\circ\text{C}$. Numerički ispitajte ovisnost točnosti rješenja problema o gustoći mreže $(\Delta r, \Delta\varphi)$, tj. o parametrima M i N .

Dodatno, možete realizirati i sljedeće eksperimente:

- Broj nepoznanica se može skoro prepoloviti ako uočimo da rješenje mora biti osno simetrično u φ (zamjena $\varphi \mapsto \pi - \varphi$ ne mijenja rješenje). Iskoristite tu simetriju rješenja i riješite cijeli problem kao problem za četvrtinu kružne ploče, kao na sljedećoj slici.



Kako treba postaviti jednačbe (ili rubne uvjete) u točkama na “okomitoj” osi simetrije? Možete pretpostaviti da je N neparan.

- Može se pokazati (separacijom varijabli) da je analitičko rješenje stacionarnog stanja temperature u točki s polarnim koordinatama (r, φ) dano s

$$u(r, \varphi) = \frac{4T}{\pi} \sum_{n=1}^{\infty} \frac{1}{2n-1} \left(\frac{r}{R}\right)^{2n-1} \sin(2n-1)\varphi.$$

Napišite funkciju koja, za zadane r i φ , nalazi “pravo” rješenje problema sumiranjem ovog reda. Diskutirajte broj članova reda potreban za odgovarajuću točnost računanja, ovisno o r i φ . Usporedite točnost ovakvog “egzaktnog” rješenja s približnim dobivenim iz diskretizacije. Posebno, probajte s diskretizacijom $\Delta r = 0.2$ m, $\Delta\varphi = \pi/8$.

- Što se događa s rješenjem u točki diskontinuiteta rubnih uvjeta — na rubovima dijametra (gdje su, zapravo, zadane dvije različite temperature na rubu)?
- Promijenite rubne uvjete tako da nemate diskontinuitet i tada ispitajte stabilnost rješenja. Na primjer, uzmite da je temperatura na polukružnom dijelu ruba zadana kao $u(R, \varphi) = T \sin \varphi$, za $0 \leq \varphi \leq \pi$. Pripadno egzaktno rješenje je

$$u(r, \varphi) = \frac{r}{R} T \sin \varphi.$$

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Zadatak 11. Linearni sustav — otklon ploče pod pritiskom.

Tanka elastična ploča podvrgnuta je pritisku okomito na površinu ploče, tako da dolazi do poprečnog (transverzalnog) pomaka ili otklona ploče u smjeru pritiska. Uz pretpostavku da je otklon w mali prema ostalim dimenzijama ploče, za w vrijedi klasična jednačba ploče

$$\Delta(D \Delta w) = p,$$

gdje je Δ Laplaceov diferencijalni operator, p je lokalni pritisak na površinu ploče (sila po jedinici površine) u smjeru pomaka w , okomito na površinu, a D je krutost ploče na savijanje, definirana na sljedeći način

$$D = \frac{Et^3}{12(1 - \nu^2)}.$$

U ovoj formuli, E je Youngov modul elastičnosti, t je debljina ploče (treća dimenzija), a ν je tzv. Poissonov omjer za materijal od kojeg je ploča napravljena. Sasvim općenito, pored w , i funkcije D , p mogu ovisiti o položaju, tj. o x i y .

Pretpostavimo još da je ploča homogena i da ima konstantnu debljinu. Onda je krutost na savijanje D konstantna kroz cijelu ploču, pa jednačbu ploče možemo napisati u obliku

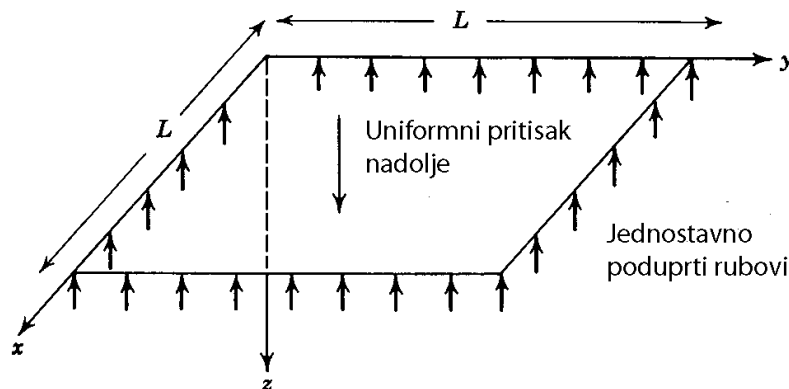
$$\Delta^2 w = \frac{p}{D}.$$

Operator Δ^2 je **biharmonijski** diferencijalni operator.

Kvadratna ploča duljine stranice L jednostavno je učvršćena na svim rubovima, tako da u rubovima nema otklona, ali se rubovi mogu slobodno zakrenuti (saviti) u ravnini okomitoj na rub. Odgovarajući rubni uvjeti su

$$w = 0, \quad \frac{\partial^2 w}{\partial \eta^2} = 0,$$

na sve četiri stranice kvadrata, gdje η označava jedinični vektor vanjske normale na rub kvadrata. Na tu ploču djelujemo uniformnim vanjskim pritiskom p , kao na sljedećoj slici.



Jedan od načina za rješenje ovog problema je uvođenje supstitucije $v = -\Delta w$. Originalni problem svodi se onda na sukcesivno rješavanje dviju Poissonovih jednačbi. Prva jednačba za funkciju v ima oblik

$$-\Delta v = \frac{p}{D},$$

uz rubni uvjet $v = 0$ na rubu kvadrata, što odgovara ranijem rubnom uvjetu na druge normalne derivacije funkcije w . Druga jednačba za traženu funkciju w izlazi iz supstitucije

$$-\Delta w = v,$$

uz rubni uvjet $w = 0$.

Napišite program za nalaženje približnog rješenja jednačbe ploče. Iskoristite standardne metode diskretizacije za Poissonovu jednačbu na kvadratu s homogenim rubnim uvjetima i riješite dobivene sustave nekom od standardnih iterativnih metoda.

Testirajte kvalitetu dobivenih rješenja za otklon ploče u ovisnosti o finoći diskretizacije. Možete koristiti i bolje diskretizacije, ali dobivena matrica može imati više punih dijagonala. Za testiranje možete uzeti da je $L = D = 1$.

Riješite isti problem i za drugačije vanjske pritiske p koji nisu uniformni (konstantni). Interesantno je probati koncentrirani pritisak u jednoj unutarnjoj točki ploče. Uzmite točku koja je čvor diskretizacijske mreže. Probajte za centar ploče i neke druge točke, bliže nekom rubu.

Cijeli problem može se riješiti i **bez** supstitucije, direktnom diskretizacijom biharmonijske jednačbe u originalnoj formulaciji.

Probajte standardnu diskretizaciju s 13 točaka, točnosti $O(h^2)$. Rubni uvjeti na druge normalne derivacije diskretiziraju se onda drugim centralnim razlikama u rubnim čvorovima mreže (u odgovarajućem smjeru, okomito na rub), koristeći po jedan fiktivni čvor “preko” ruba. Vrijednost funkcije u tom fiktivnom čvoru se eliminira iz diskretiziranog rubnog uvjeta. Kako tada izgleda pripadni linearni sustav jednačbi? Pokušajte riješiti dobiveni linearni sustav i usporediti rezultate s prethodnima, za razne vanjske pritiske p .

Literatura: predavanja “Iterativne metode”, “Dodaci” i “Numerička interpolacija, deriviranje i integriranje — Formule i tablice iz knjige Abramowitz–Stegun, 25. poglavlje” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Original: tekst i slika preuzeti su iz knjige

- B. Carnahan, H. A. Luther, J. O. Wilkes, “Applied Numerical Methods”, J. Wiley & Sons, New York, 1969., Example 7.7, str. 491–497.

Zadatak 12. Iterativne metode za linearne sustave — difuzijska jednadžba

Promatramo sustave linearnih jednadžbi koji nastaju diskretizacijom difuzijske jednadžbe

$$\frac{\partial u}{\partial t}(x, t) = \frac{\partial^2 u}{\partial x^2}(x, t), \quad x \in [a, b], \quad t \in [0, T],$$

s Dirichletovim rubnim uvjetima

$$u(a, t) = y_a(t), \quad u(b, t) = u_b(t), \quad u(x, 0) = u_0(x).$$

Standardna metoda za dobivanje aproksimacija rješenja difuzijske jednadžbe je diskretizacija pomoću konačnih diferencija. Za njenu primjenu potrebno je definirati pravokutnu mrežu na $[a, b] \times [0, T]$, i u svakoj točki mreže derivaciju zamijeniti konačnom diferencijom. Raznim tipovima konačnih diferencija dobivaju se različite metode, sa različitim svojstvima.

Mrežu definiramo pomoću ekvidistantnih segmenata na x i t osi

$$\delta x = \frac{b - a}{n}, \quad \delta t = \frac{T}{m},$$

a čvorovi mreže imaju oblik

$$(x_i, t_j) = (a + i\delta x, j\delta t), \quad i = 0, \dots, n, \quad j = 0, \dots, m.$$

Ove metode će računati aproksimativno rješenje samo u čvorovima mreže, pri čemu vrijede oznake

$$u_{i,j} \approx u(a + i\delta x, j\delta t) = u(x_i, t_j).$$

Za odabir aproksimacija parcijalnih derivacija imamo tri mogućnosti.

- Parcijalna derivacija drugog reda po x aproksimira se simetričnom centralnom konačnom diferencijom

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\delta x)^2}.$$

Parcijalna derivacija po t aproksimira se konačnom diferencijom unaprijed

$$\frac{\partial u}{\partial t}(x_i, t_j) \approx \frac{u_{i,j+1} - u_{i,j}}{\delta t}.$$

Ovakav izbor konačnih diferencija definira **eksplicitnu metodu**.

- Parcijalna derivacija drugog reda po x aproksimira se simetričnom centralnom konačnom diferencijom

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\delta x)^2}.$$

Parcijalna derivacija po t aproksimira se konačnom diferencijom unazad

$$\frac{\partial u}{\partial t}(x_i, t_j) \approx \frac{u_{i,j} - u_{i,j-1}}{\delta t}.$$

Ovakav izbor konačnih diferencija definira **potpunu implicitnu metodu**, koja rezultira rješavanjem sustava.

- Parcijalna derivacija drugog reda po x aproksimira se srednjom vrijednošću simetričnih centralnih konačnih diferencija za t_j i t_{j+1}

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_{j+\frac{1}{2}}) \approx \frac{1}{2} \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\delta x)^2} + \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\delta x)^2} \right).$$

Parcijalna derivacija po t aproksimira se centralnom konačnom diferencijom

$$\frac{\partial u}{\partial t}(x_i, t_{j+\frac{1}{2}}) \approx \frac{u_{i,j} - u_{i,j-1}}{\delta t}.$$

Ovakav izbor konačnih diferencija definira **Crank–Nicolsonovu metodu**, koja također rezultira rješavanjem sustava.

Sve tri metode računaju aproksimacije $u_{i,j}$ za $0 < i < n$, $0 < j \leq m$. Rubni uvjeti se koriste za određivanje $u_{0,j}$ i $u_{n,j}$:

$$\begin{aligned} u_{0,j} &= u_a(j\delta t), & 0 < j \leq m, \\ u_{n,j} &= u_b(j\delta t), & 0 < j \leq m. \end{aligned}$$

Za pokretanje ovih iterativnih metoda koristi se inicijalni uvjet

$$u_{i,0} = u_0(a + i\delta x), \quad 0 \leq i \leq n.$$

Glavna petlja iterira $j = 1, \dots, m$ i u svakom koraku rješava aproksimacije za fiksni t_j . Iterativna metoda završava za $j = m$ i rješenjem

$$u_{i,m}, \quad 0 \leq i \leq n,$$

što predstavlja aproksimaciju rješenja za $u(a + i\delta x, T)$, $0 \leq i \leq n$.

Napišite program koji implementira sve tri metode diskretizacije, pri čemu za sustave prvo provjerite da li imaju neku posebnu strukturu a onda za njihovo rješavanje primijenite različite metode pogodne za tu strukturu. Svoj program testirajte na raznim primjerima sa raznim rubnim i inicijalnim uvjetima. Osim toga za svaki primjer testirajte diskretizacije na raznim mrežama, sa različitim odnosima između δx i δt . Uporedite metode diskretizacija na istoj mreži. Što možete zaključiti? Da li su sve tri metode jednako točne? Kada se događa da nisu?

Literatura: 25. predavanje "Numerička analiza" na doktorskom studiju 2011./12. (dostupno na http://web.math.pmf.unizg.hr/~nela/nad_11_12.html)

Zadatak 13. Iterativne metode za svojstveni problem — metoda bisekcije

Metoda bisekcije služi za nalaženje svojstvenih vrijednosti simetričnih tridijagonalnih matrica koje se nalaze u nekom zadanom poluotvorenom intervalu $[a, b)$ ili za nalaženje m -te najveće svojstvene vrijednosti. Za simetričnu matricu A definiramo

- $n(b)$ = broj svojstvenih vrijednosti od A manjih od b , tj. broj negativnih svojstvenih vrijednosti matrice $A - bI$,
- $n(a)$ = broj svojstvenih vrijednosti od A manjih od a , tj. broj negativnih svojstvenih vrijednosti matrice $A - aI$.

Iz prethodnih razmatranja odmah slijedi da je broj svojstvenih vrijednosti matrice A u intervalu $[a, b)$ jednak $n_b - n_a$.

Dalje, za simetričnu tridijagonalnu matricu T oblika

$$T = \begin{bmatrix} d_1 & e_1 & & \cdots & 0 \\ e_1 & d_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & e_{n-1} \\ 0 & \cdots & & e_{n-1} & d_n \end{bmatrix},$$

sa T_k označimo njenu vodeću $k \times k$ podmatricu, i definiramo polinome $p_k(x) = \det(T_k - xI)$, za $k = 1, \dots, n$. Razvojem determinante može se pokazati da

$$p_k(x) = (d_k - x)p_{k-1}(x) - e_{k-1}^2 p_{k-2}(x), \quad \text{za } k = 2, \dots, n, \text{ uz uvjet } p_0(x) = 1.$$

Klasičan Sturmov teorem kaže ako su svi elementi sporednih dijagonala (e_i) različiti od nule, te ako $s(x)$ označava broj promjena predznaka u nizu $\{p_0(x), p_1(x), \dots, p_n(x)\}$, tada je $s(x) = n(x)$, odnosno $s(x)$ je jednak broju svojstvenih vrijednosti od T manjih od x . Ovdje $p_k = 0$ računamo kao pozitivan broj.

Veličine $n(x)$ mogu se iskoristiti u metodi bisekcije, i to na dva načina.

- Za računanje m -te najveće svojstvene vrijednosti λ_m od T . Tada možete npr. staviti $a_0 = -\|T\|_\infty$ i $b_0 = \|T\|_\infty$. Pretpostavimo da smo u k -tom koraku iteracija izračunali interval $[a_k, b_k)$ za koji vrijedi da je $\lambda_m \in [a_k, b_k)$ i $b_k - a_k = (b_0 - a_0)/2^k$. Tada se u $k + 1$ -oj iteraciji određuje interval $[a_{k+1}, b_{k+1})$ na sljedeći način:

1. $c_k = (a_k + b_k)/2$
2. izračunajte $n(c_k)$
3. ako je
 - $n(c_k) \geq m$, tada su $a_{k+1} = a_k$ i $b_{k+1} = c_k$
 - $n(c_k) < m$, tada su $a_{k+1} = c_k$ i $b_{k+1} = b_k$

Iz ovakvog odabira intervala slijedi da je $\lambda_m \in [a_{k+1}, b_{k+1})$ i $b_{k+1} - a_{k+1} = (b_0 - a_0)/2^{k+1}$. Daljnjim iteracijama, interval se može dovoljno suziti oko svojstvene vrijednosti. Za posljednji najuži interval, kao aproksimaciju svojstvene vrijednosti uzmite polovinu intervala $\mu = c_K = (a_K + b_K)/2$. Tada vrijedi $|\mu - \lambda_m| \leq (b_0 - a_0)/2^{K+1}$, čime se dobiva i kriterij zaustavljanja: za željenu toleranciju ε metodu zaustavljamo kada je

$$\frac{b_0 - a_0}{2^{K+1}} < \varepsilon \quad \Rightarrow \quad K > \frac{\log\left(\frac{b_0 - a_0}{2\varepsilon}\right)}{\log 2}.$$

- Za računanje svojstvenih vrijednosti od T koje se nalaze u nekom zadanom polu-otvorenom intervalu $[a, b)$. Za početak izračunajte $n(a)$ i $n(b)$. Ako je $n(a) = n(b)$ tada u danom intervalu nema svojstvenih vrijednosti i metoda se zaustavlja. U suprotnom skup podataka $\{a, n(a), b, n(b)\}$ stavite na radnu listu intervala koje još treba obraditi. Sada, tako dugo dok radna lista nije prazna
 - skinite prvi dostupni skup podataka $\{a_i, n(a_i), b_i, n(b_i)\}$
 - ako je $b_i - a_i < \varepsilon$, za neku toleranciju ε tada možete vratiti bilo koji broj iz intervala, (najjednostavnije je opet uzeti $(a_i + b_i)/2$) i on predstavlja aproksimaciju za $n(b_i) - n(a_i)$ svojstvenih vrijednosti u tom intervalu
 - ako je $b_i - a_i \geq \varepsilon$
 - * $c_i = (a_i + b_i)/2$
 - * izračunajte $n(c_i)$
 - * ako je $n(c_i) > n(a_i)$ stavite skup podataka $\{a_i, n(a_i), c_i, n(c_i)\}$ na radnu listu
 - * ako je $n(b_i) > n(c_i)$ stavite skup podataka $\{c_i, n(c_i), b_i, n(b_i)\}$ na radnu listu

Za računanje svojstvenih vrijednosti općenite simetrične matrice, prvo ju trebete tridijagonalizirati pomoću Householderovih reflektora.

Napišite program koji implementira obje varijante metode, i testirajte je na raznim primjerima simetričnih matrica za koje znate svojstvene vrijednosti. Varirajte toleranciju ε i usporedite broj iteracija. Za dobivene aproksimacije svojstvenih vrijednosti, odgovarajuće svojstvene vektore izračunajte inverznim iteracijama, pri čemu za rješavanje tridijagonalnih sustava koristite neku pogodnu metodu.

Literatura: 10. predavanje "Numerička analiza" na doktorskom studiju 2011./12. (dostupno na http://web.math.pmf.unizg.hr/~nela/nad_11_12.html), te podsekcije 8.3.1, 8.5.1 i 8.5.2 iz knjige: G. H. Golub, C. F. Van Loan, "Matrix computations", Third Edition, The Johns Hopkins University Press, Baltimore and London, 1996.

Problem ažuriranja je sad sveden na nalaženje svojstvenih vrijednosti λ matrice oblika

$$\widehat{D} = D + \rho uu^T,$$

pri čemu je D dijagonalna, $\rho = b_m$, a u je poznati vektor.

Prvo pretpostavimo da je matrica $D - \lambda I$ regularna za sve svojstvene vrijednosti λ od \widehat{D} , pa karakteristični polinom matrice \widehat{D} glasi

$$\det(D + \rho uu^T - \lambda I) = \det[(D - \lambda I)(I + \rho(D - \lambda I)^{-1}uu^T)].$$

Iz pretpostavke o regularnosti $D - \lambda I$ izlazi da mora biti

$$\det(I + \rho(D - \lambda I)^{-1}uu^T) = 0$$

kad god je λ svojstvena vrijednost matrice \widehat{D} . Nadalje, matrica $I + \rho(D - \lambda I)^{-1}uu^T$ je jedinična matrica plus matrica ranga 1, za koju se lako računa determinanta. Može se pokazati da vrijedi

$$\begin{aligned} \det(I + \rho(D - \lambda I)^{-1}uu^T) &= 1 + \rho u^T (D - \lambda I)^{-1} u \\ &= 1 + \rho \sum_{i=1}^n \frac{u_i^2}{d_i - \lambda} =: f(\lambda). \end{aligned}$$

Jednadžba $f(\lambda) = 0$ obično se zove **sekularna jednadžba**, a vrijednosti λ za koje je $f(\lambda) = 0$ su svojstvene vrijednosti.

U najjednostavnijem slučaju su svi d_i različiti, a svi $u_i \neq 0$. Budući da je f monotona (f' i f'' fiksno predznaka), onda

- Newtonova metoda za nalaženje nultočaka konvergira, ali
- konvergencija može biti vrlo spora, ako su d_i vrlo maleni.

Direktnim računom može se potvrditi da ako je λ svojstvena vrijednost matrice $D + \rho uu^T$, onda je $(D - \lambda I)^{-1}u$ svojstveni vektor za tu svojstvenu vrijednost.

U prethodnoj konstrukciji algoritma smo pretpostavili da su svi d_i međusobno različiti, i svi $u_i \neq 0$. Ako to nije slučaj, tada sekularna jednadžba ima $k < n$ nultočaka. Ali tada se preostalih $n - k$ svojstvenih vrijednosti može vrlo “jeftino” dobiti.

- Ako je $d_i = d_{i+1}$ tada je d_i svojstvena vrijednost od \widehat{D} i odgovarajući svojstveni vektor je $u_{i+1}e_i - u_i e_{i+1}$.
- Ako je $u_i = 0$ tada je d_i svojstvena vrijednost od \widehat{D} i odgovarajući svojstveni vektor je e_i .

Ovdje vidimo da su to ujedno i slučajevi kada $D - \lambda I$ nije regularna.

Za računanje svojstvenih vrijednosti općenite simetrične matrice, prvo ju trebate tridijagonalizirati pomoću Householderovih reflektora.

Napišite program koji računa svojstvene vrijednosti simetrične matrice pomoću metode “podijeli pa vladaj”. Za tridijagonalnu simetričnu matricu T algoritam rekurzivno računa svojstvene vrijednosti i svojstvene vektore $T = Q\Lambda Q^T$ na sljedeći način.

- ako je dimenzija matrice $n = 1$ vratite $Q = 1$ i $\Lambda = T$

- ako je dimenzija matrice $n > 1$ tada
 1. izračunajte rastav $T = \text{diag}(T_1, T_2) + b_m v v^T$
 2. rekurzivnim pozivom izračunajte $T_1 = Q_1 \Lambda_1 Q_1^T$
 3. rekurzivnim pozivom izračunajte $T_2 = Q_2 \Lambda_2 Q_2^T$
 4. izračunajte $\widehat{D} = D + \rho u u^T$ iz $Q_1, \Lambda_1, Q_2, \Lambda_2$
 5. izračunajte svojstvene vrijednosti Λ i svojstvene vektore Q' od \widehat{D}
 6. izračunajte $Q = \text{diag}(Q_1, Q_2) \cdot Q'$
 7. vratite Q i Λ

Testirajte svoj program na raznim simetričnim matricama sa poznatim svojstvenim vrijednostima. Usporedite rezultate ove metode sa raznim metodama za rješavanje sekularne jednadžbe. Provjerite ortogonalnost izračunatih svojstvenih vektora.

Literatura: 10. predavanje "Numerička analiza" na doktorskom studiju 2011./12. (dostupno na http://web.math.pmf.unizg.hr/~nela/nad_11_12.html), i podsekcija 5.3.3 iz knjige J. Demmel, "Applied numerical linear algebra", SIAM, Philadelphia, 1997.

Zadatak 15. Iterativne metode za svojstveni problem — Jacobi-Davidsonova metoda

Jacobi–Davidsonova metoda je iterativna metoda koja računa svojstveni vektor i odgovarajuću svojstvenu vrijednost simetrične matrice, koja je najbliža nekoj zadanoj vrijednosti. Pomoću te metode moguće je izračunati i ekstremne svojstvene vrijednosti: najmanju i najveću. Glavna ideja metode je da se u svakoj iteraciji računa okomita korekcija trenutne aproksimacije zadanog svojstvenog vektora. Izborom korekcija u svakom koraku proširuje se potprostor u kojem tražimo aproksimativna rješenja primjenom Rayleigh–Ritzove metode.

Sama metoda radi na sljedeći način. Pretpostavimo da želimo izračunati svojstvenu vrijednost od A koja je najbliža τ , i da nam je dan inicijalni vektor t_0 . Inicijalizirajte $V_0 = \emptyset$, $W_0 = \emptyset$, $H_0 = \emptyset$, i $k = 0$, pri čemu je k indeks iteracije. Za svaku iteraciju $k = 1, 2, \dots$ radite kao što slijedi.

1. ako je $k > 1$, t_{k-1} ortogonalizirajte u odnosu na stupce u V_{k-1} : $t_{k-1} = (I - V_{k-1}V_{k-1}^T)t_{k-1}$
2. izračunajte $v_k = t_{k-1}/\|t_{k-1}\|_2$, $w_k = Av_k$, $V_k = [V_{k-1}, v_k]$, $W_k = [W_{k-1}, w_k]$
3. izračunajte $H_k = V_k^T AV_k$:
 - za $k = 1$ je $H_1 = [v_1^T w_1]$
 - za $k > 1$ izračunajte $H_k = \begin{bmatrix} H_{k-1} & V_{k-1}^T w_k \\ w_k^T V_{k-1} & v_k^T w_k \end{bmatrix}$
4. Rayleigh–Ritzov korak: izračunajte svojstveni vektor s_k , $\|s_k\|_2 = 1$, i odgovarajuću svojstvenu vrijednost θ_k od H_k koja je najbliža τ .
5. izračunajte $u_k = V_k s_k$, $r_k = Au_k - \theta_k u_k$
6. ako je rezidual manji od neke tolerancija ε : $\|r_k\|_2 < \varepsilon$ vratite aproksimaciju svojstvene vrijednosti $\tilde{\lambda} = \theta_k$ i aproksimaciju svojstvenog vektora $\tilde{x} = u_k$ i metoda je gotova
7. ako rezidual nije dovoljno mali (aproksimativno) riješite jednadžbu korekcije za t_k

$$(I - u_k u_k^T)(A - \theta_k I)(I - u_k u_k^T)t_k = -r_k, \quad t_k \perp u_k.$$

Obzirom da sa rastom broja iteracija, raste i dimenzija prostora V_k kao i dimenzija matrice H_k za koju računamo svojstveni par, metoda se restarta nakon svakih k_{max} iteracija. Retart možete napraviti tako da stavite $V_1 = [u_{k_{max}}]$, $W_1 = [Au_{k_{max}}]$, $H_1 = [\theta_{k_{max}}]$.

Napišite program koji implementira Jacobi–Davidsonov algoritam, pri čemu možete napraviti nekoliko varijanti ovisno o izboru metode za računanje traženog svojstvenog para “male” matrice H_k u Rayleigh–Ritzov koraku, i o izboru metode za rješavanje jednadžbe korekcije.

- Za računanje traženog svojstvenog para “male” matrice H_k možete koristiti inverzne iteracije, ili neku metodu za računanje cijele spektralne dekompozicije simetrične matrice kao što je Jacobijeva metoda, ili QR metoda (sa tridijagonalizacijom kao prvi korak).
- Za ješavanje jednadžbe korekcije možete je riješiti

– direktnom metodom:

$$t_k = \alpha_k(A - \theta_k I)^{-1}u_k - (A - \theta_k I)^{-1}r_k, \quad \alpha_k = \frac{u_k^T(A - \theta_k I)^{-1}r_k}{u_k^T(A - \theta_k I)^{-1}u_k}$$

– iterativnom metodom: GMRES ili MINRES (slična kao metoda konjugiranih gradijenata), pri tom je testirajte sa raznim tolerancijama na točnost

Testirajte svoj program na raznim simetričnim matricama sa poznatim svojstvenim vrijednostima, sa raznim metodama za rješavanje gore opisanih potproblema i sa različitim tolerancijama na točnost.

Literatura: Nela Bosner, "Fast methods for large scale singular value decomposition", doktorski rad (dostupno na mojoj web stranici)

Zadatak 16. Dekompozicija singularnih vrijednosti — Golub–Kahanov algoritam

Ova metoda za računanje dekompozicije singularnih vrijednosti zasniva se na dva osnovna koraka: Householderovoj bidiagonalizaciji i Golub–Kahanovom algoritmu za računanje dekompozicije singularnih vrijednosti bidiagonalne matrice. Za matricu $A \in \mathbb{R}^{m \times n}$, $m \geq n$, metoda izvršava sljedeće korake.

1. Householderova bidiagonalizacija se bazira na množenju matrice Householderovim reflektorima s lijeva i s desna, tako da na kraju dobijemo

$$\begin{bmatrix} B \\ 0 \end{bmatrix} = U_n \cdots U_1 A V_1 \cdots V_{n-2}, \quad U = U_1 \cdots U_n, \quad V = V_1 \cdots V_{n-2},$$

gdje su $U_k \in \mathbb{R}^{m \times m}$ i $V_k \in \mathbb{R}^{n \times n}$ Householderovi reflektori, a $B \in \mathbb{R}^{n \times n}$ je bidiagonalna matrica oblika

$$B = \begin{bmatrix} d_1 & f_1 & & & & \\ & d_2 & f_3 & & & \\ & & \ddots & \ddots & & \\ & & & d_{n-1} & f_{n-1} & \\ & & & & & d_n \end{bmatrix}.$$

Postupak se odvija u n koraka. Za $k = 1, \dots, n$ radite sljedeće

- (a) generirajte Householderov reflektor $U_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \bar{U}_k \end{bmatrix}$ takav da poništava elemente $A(k+1:m, k)$: $\bar{U}_k A(k:m, k) = [* \ 0 \ \cdots \ 0]^T$
 - (b) ažurirajte ostatak matrice A sa U_k : $A(k:m, k+1:n) = \bar{U}_k A(k:m, k+1:n)$
 - (c) za $k \leq n-2$ generirajte Householderov reflektor $V_k = \begin{bmatrix} I_k & 0 \\ 0 & \bar{V}_k \end{bmatrix}$ takav da poništava elemente $A(k, k+2:n)$: $A(k, k+1:n) \bar{V}_k = [* \ 0 \ \cdots \ 0]$
 - (d) ažurirajte ostatak matrice A sa V_k : $A(k+1:m, k+1:n) = A(k+1:m, k+1:n) \bar{V}_k$
2. Golub–Kahanov algoritam za računanje dekompozicije singularnih vrijednosti bidiagonalne matrice svodi se na implicitnu primjenu QR metode sa pomakom na tridijagonalnu matricu $T = B^T B$: definira se $T_0 = B_0^T B_0$, gdje je $B_0 = B$, a potom za $k = 0, 1, 2, \dots$

$$\begin{aligned} T_k - \lambda_k I &= QR \quad (\text{QR faktorizacija}) \\ T_{k+1} &= RQ + \lambda_k I, \end{aligned}$$

čime se dobiva nova tridijagonalna matrica T_{k+1} sa $T_{k+1} = Q^T T_k Q$, ali matrica $T_{k+1} = B_{k+1}^T B_{k+1}$ se nikada eksplicitno ne formira. Pomak λ_k se bira tako da ubrza konvergenciju. Koraci algoritma su sljedeći:

- (a) izračunajte svojstvenu vrijednost λ_k matrice

$$T_k(n-1:n, n-1:n) = \begin{bmatrix} d_{n-1}^2 + f_{n-1}^2 & d_{n-1} f_n \\ d_{n-1} f_n & d_n^2 + f_n^2 \end{bmatrix}$$

koja je bliža vrijednosti $d_n^2 + f_n^2$.

(b) izračunajte $c_{\theta_1} = \cos(\theta_1)$ and $s_{\theta_1} = \sin(\theta_1)$ takve da je

$$\begin{bmatrix} c_{\theta_1} & s_{\theta_1} \\ -s_{\theta_1} & c_{\theta_1} \end{bmatrix}^T \begin{bmatrix} d_1^2 - \lambda_k \\ d_1 f_2 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}$$

i definirajte Givensovu rotaciju $V_1 = G_{1,2}(\theta_1)$.

(c) izračunajte Givensove rotacije V_2, \dots, V_{n-1} takve da ako je $V^{(k)} = V_1 \cdots V_{n-1}$ tada je $T_{k+1} = (V^{(k)})^T T_k V^{(k)}$ opet tridijagonalna, a $V^{(k)} e_1 = V_1 e_1$. taj postupak se naziva “naganjanje kvрге” u bidijagonalnoj matrici B_k

$B_{k,1} \leftarrow B_k V_1 =$		$B_{k,2} \leftarrow U_1^T B_{k,1} =$	
$B_{k,3} \leftarrow B_{k,2} V_2 =$		$B_{k,4} \leftarrow U_2^T B_{k,3} =$	
$B_{k,5} \leftarrow B_{k,4} V_3 =$		$B_{k,6} \leftarrow U_3^T B_{k,5} =$	
$B_{k,7} \leftarrow B_{k,6} V_4 =$		$B_{k,8} \leftarrow U_4^T B_{k,7} =$	
$B_{k,9} \leftarrow B_{k,8} V_5 =$		$B_{k,10} \leftarrow U_5^T B_{k,9} =$	

Ovaj korak završava formiranjem nove bidijagonalne matrice B_{k+1} koja se dobiva iz B_k pomoću transformacija

$$B_{k+1} = (U_{n-1}^T \cdots U_1^T) B_k (V_1 \cdots V_{n-1}) = (U^{(k)})^T B_k V^{(k)}.$$

Iteracije se završavaju kada svi vandijagonalni elementi f_i , $i = 1, \dots, n - 1$ postanu dovoljno mali, što se mjeri sa

$$|f_i| \leq \varepsilon(|d_i| + |d_{i+1}|),$$

za zadanu toleranciju ε .

3. Konačna dekompozicija singularnih vrijednosti matrice A se postiže tako da, ako definiramo

$$A = U_A \begin{bmatrix} B \\ 0 \end{bmatrix} V_A^T \quad (\text{bidijagonalizacija})$$

$$B = U_B \Sigma V_B^T \quad (\text{bidijagonalni SVD})$$

vrijedi

$$U = U_A \begin{bmatrix} U_B & 0 \\ 0 & I_{m-n} \end{bmatrix}, \quad V = V_A V_B, \quad A = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T.$$

Napišite potprogram koji implementira gore opisanu metodu, i kao ulazni parametar uzima matricu A i toleranciju ε . Primijenite svoj potprogram na raznim matricama, koje imaju unaprijed poznate singularne vrijednosti. Kontrolirajte relativnu grešku singularnih vrijednosti, a naročito obratite pozornost na matrice sa velikom razlikom u redu veličine najmanje i najveće singularne vrijednosti. Usporedite svoj rezultat za $\varepsilon = n \cdot u$, gdje je u mašinski epsilon, sa rezultatom LAPACK-ovog potprograma `dgesvd_()`. Usporedite brzine izvršavanja ta dva potprograma.

Literatura: Nela Bosner, "Fast methods for large scale singular value decomposition", doktorski rad (dostupno na mojoj web stranici)

Zadatak 17. Dekompozicija singularnih vrijednosti — jednostrana Jacobijeva metoda

Jednostrana Jacobijeva metoda koristi se za računanje SVD-a, tako da se Jacobijeva metoda za simetričan svojstveni problem implicitno primijeni na matricu $A^T A$, gdje je $A \in \mathbb{R}^{m \times n}$. Jacobijeve rotacije primijenjuju se u tom slučaju samo s jedne strane matrice A . Algoritam generira niz matrica $A^{(i)}$, takav da je

$$\begin{aligned} A^{(0)} &= A & V^{(0)} &= I \\ A^{(i)} &= A^{(i-1)} R_i & V^{(i)} &= V^{(i-1)} R_i \end{aligned}$$

Niz $(A^{(i)})^T A^{(i)}$ konvergira tada prema dijagonalnoj matrici

$$\lim_{i \rightarrow \infty} (A^{(i)})^T A^{(i)} = \Sigma^2 = \text{diag}(\sigma_1^2, \dots, \sigma_n^2),$$

i

$$V = \lim_{i \rightarrow \infty} V^{(i)}.$$

U limesu imamo izračunatu sljedeću faktorizaciju

$$A^T A = V \Sigma^2 V^T,$$

gdje

$$A^{(\infty)} = \lim_{i \rightarrow \infty} A^{(i)}$$

ima ortogonalne stupce. Odavde U može biti izračunat kao

$$U = A^{(\infty)} \Sigma^{-1} \quad \text{ako je } \det \Sigma \neq 0.$$

Napišite algoritam za ovu metodu, s time da ne generirate matricu $(A^{(i-1)})^T A^{(i-1)} = H^{(i-1)} = [h_{pq}^{(i-1)}]$, već u svakoj iteraciji izračunate samo one njene elemente koji su potrebni za generiranje tekuće jacobijeve rotacije:

$$H_{p_i, q_i}^{(i-1)} = \begin{bmatrix} h_{p_i p_i}^{(i-1)} & h_{p_i q_i}^{(i-1)} \\ h_{q_i p_i}^{(i-1)} & h_{q_i q_i}^{(i-1)} \end{bmatrix}.$$

Napišite postupak odabira odgovarajućeg kuta Jacobijevoj matrici R_i i postupak ažuriranja samo onih elemenata koji se mijenjaju primjenom Jacobijeve matrice. Pivotalna strategija neka je ciklička, pri čemu se ne treba generirati Jacobijeva rotacija ukoliko je $|h_{p_i q_i}^{(i-1)}| \leq \text{tol} \sqrt{h_{p_i p_i}^{(i-1)} h_{q_i q_i}^{(i-1)}}$, odnosno taj se korak preskače. Iteracije se zaustavljaju nakon jednog praznog ciklusa, tj. kada je prethodni kriterij ispunjen za sve vandijagonalne elemente.

Napišite potprogram koji implementira jednostranu Jacobijevu metodu, i kao ulazni parametar uzima matricu A i toleranciju tol . Primijenite svoj potprogram na raznim matricama, koje imaju unaprijed poznate singularne vrijednosti. Kontrolirajte relativnu grešku singularnih vrijednosti, a naročito obratite pozornost na matrice sa velikom razlikom u redu veličine najmanje i najveće singularne vrijednosti. Usporedite svoj rezultat za $\text{tol} = n \cdot u$, gdje je u mašinski epsilon, sa rezultatom LAPACK-ovog potprograma `dgesvd_()`. Usporedite brzine izvršavanja ta dva potprograma.

Literatura: vježbe “Problemi svojstvenih vrijednosti i SVD”

Zadatak 18. Svojtveni problem — Gaussove integracijske formule.

Gaussova integracijska formula reda n na intervalu $[a, b]$ s težinskom funkcijom w ima oblik

$$\int_a^b w(x)f(x) dx = I_n(f) + E_n(f), \quad I_n(f) = \sum_{k=1}^n w_k f(x_k),$$

s tim da se čvorovi x_k i težine w_k određuju iz uvjeta da ova formula bude egzaktna za sve polinome do stupnja $2n - 1$.

Znamo da se čvorovi mogu dobiti kao nultočke odgovarajućeg ortogonalnog polinoma p_n . U praksi se čvorovi i težine dobivaju rješavanjem svojstvenog problema za simetričnu tridijagonalnu matricu T_n koja nastaje simetrizacijom tročlane rekurzije za pripadne ortogonalne polinome.

Ako pripadni monični ortogonalni polinomi p_k zadovoljavaju rekurziju

$$p_{k+1}(x) = (x - \alpha_k)p_k(x) - \beta_k p_{k-1}(x), \quad k = 0, 1, \dots,$$

uz $p_{-1}(x) = 0$, $p_0(x) = 1$, onda simetrizacijom dobivamo rekurziju za skalirane ortogonalne polinome \tilde{p}_k

$$\sqrt{\beta_{k+1}} \tilde{p}_{k+1}(x) = (x - \alpha_k) \tilde{p}_k(x) - \sqrt{\beta_k} \tilde{p}_{k-1}(x), \quad k = 0, 1, \dots,$$

uz $\tilde{p}_{-1}(x) = 0$, $\tilde{p}_0(x) = 1$. Pripadna tridijagonalna simetrična matrica je

$$T_n = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \sqrt{\beta_{n-2}} & \alpha_{n-2} & \sqrt{\beta_{n-1}} & \\ & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} & \end{bmatrix}.$$

Svojtvene vrijednosti ove matrice su čvorovi x_k , a težine w_k dobivamo iz prve komponente $v_{1,k}$ pripadnih ortonormiranih svojstvenih vektora v_k

$$w_k = \beta_0 \cdot v_{1,k}^2, \quad \text{gdje je} \quad \beta_0 = \int_a^b w(x) dx.$$

Napišite program za nalaženje parametara Gaussovih integracijskih formula reda n , s tim da se zadaje red formule i koeficijenti u tročlanoj rekurziji. Testirajte program na rekurzijama za klasične ortogonalne polinome (Legendre, Čebišev, Laguerre, Hermite). Za provjeru točnosti dobivenih parametara iskoristite uvjet egzaktnosti integracije na polinomima do odgovarajućeg stupnja.

Literatura: predavanja 12 i 13 iz “Numeričke matematike” (dostupno na web stranici prof. Singera).

Zadatak 19. Svojtveni problem — Sturm–Liouvilleova jednadžba.

Mnogi fizikalni problemi, poput vibracija žice, vode na svojtveni problem za Sturm–Liouvilleov operator s odgovarajućim rubnim uvjetima. U najjednostavnijem simetričnom slučaju, ovaj svojtveni problem ima sljedeći oblik

$$\mathcal{L}u := -(p(x)u')' + q(x)u = \lambda u, \quad x \in (a, b),$$

s homogenim rubnim uvjetima $u(a) = u(b) = 0$, gdje su p i q zadane neprekidne i pozitivne funkcije na $[a, b]$.

Može se pokazati da ovaj problem ima netrivialna rješenja $u \neq 0$ samo za neke vrijednosti parametra λ . Takve vrijednosti λ zovemo svojtvenim vrijednostima, a pripadna netrivialna rješenja u zovemo svojtvenim funkcijama ovog problema. Dodatno, može se pokazati da ovaj problem ima prebrojivo mnogo (jednostrukih) pozitivnih svojtvenih vrijednosti λ_k , $k \in \mathbb{N}$, koje možemo poredati tako da je

$$0 < \lambda_1 < \lambda_1 < \dots < \lambda_k < \dots$$

i vrijedi $\lambda_k \rightarrow \infty$, kad $k \rightarrow \infty$.

Napišite program za približno nalaženje svojtvenih vrijednosti i funkcija Sturm–Liouvilleovog rubnog problema. Iskoristite standardne metode diskretizacije i riješite kao svojtveni problem za matricu dobivenu diskretizacijom.

Za zadani $N \in \mathbb{N}$, segment $[a, b]$ podijelimo uniformnom mrežom na $N + 1$ podintervala duljine $h = (b - a)/(N + 1)$, tako da su čvorovi mreže

$$x_i = a + ih, \quad i = 0, \dots, N + 1.$$

Uvedimo još skraćenu oznaku indeksom za funkcijske vrijednosti bilo koje funkcije f na \mathbb{R} , tako da je $f_\alpha := f(a + \alpha h)$, za bilo koji realni $\alpha \in \mathbb{R}$.

Ako iskoristimo standardnu diskretizaciju derivacije centralnim razlikama u unutarnjim čvorovima mreže, dobivamo sustav jednadžbi

$$-\frac{1}{h} \left(p_{i+1/2} \frac{u_{i+1} - u_i}{h} - p_{i-1/2} \frac{u_i - u_{i-1}}{h} \right) + q_i u_i = \lambda u_i,$$

za $i = 1, \dots, N$. Nakon uvrštavanja rubnih uvjeta $u_0 = u_{N+1} = 0$, dobivamo “obični” svojtveni problem za simetričnu pozitivno definitnu tridijagonalnu matricu reda N .

Najmanje svojtvene vrijednosti te matrice i pripadni svojtveni vektori su razumno dobra aproksimacija za odgovarajuće svojtvene vrijednosti i funkcije polaznog problema. Za standardnu diskretizaciju brzina konvergencije je $O(h^2)$.

Za zadane funkcije p i q , testirajte kvalitetu dobivenih svojtvenih vrijednosti i funkcija u ovisnosti o finoći diskretizacije. Možete koristiti i bolje diskretizacije, ali dobivena matrica može imati više punih dijagonala.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

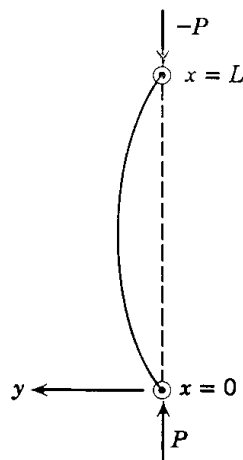
Zadatak 20. Svojstveni problem — dvostrano stisnuta greda.

Poprečni (transverzalni) otklon y elastične savitljive potporne grede (ili nosivog stupa) zadovoljava diferencijalnu jednadžbu oblika

$$\frac{d^2y}{dx^2} = \frac{M}{EI},$$

gdje je x udaljenost duž osi grede, E je Youngov modul elastičnosti, I je odgovarajući moment inercije poprečnog presjeka grede, a M je lokalni moment izvijanja (zakretni moment) pod utjecajem vanjske uzdužne (longitudinalne) sile ili opterećenja. Pretpostavljamo da greda ima konstantni presjek po x osi, i da je homogena, tako da su E i I konstantni.

Uspravna greda duljine L učvršćena je na oba kraja tako da u rubovima nema otklona, ali rubovi mogu slobodno “kliziti” duž vertikalne osi x . Takvu gredu uzdužno “stišćemo” na oba kraja simetričnom vanjskom silom (opterećenjem) P , kao na sljedećoj slici.



Za lokalni moment izvijanja onda vrijedi $M(x) = -Py(x)$, pa “ravnotežni” otklon grede zadovoljava jednadžbu

$$-\frac{d^2y}{dx^2} = \frac{P}{EI} y.$$

Za dovoljno mala opterećenja P , ovaj problem ima samo trivijalno rješenje $y = 0$. Netrivijalna rješenja ove jednadžbe postoje samo ako je faktor na desnoj strani jednak nekoj od **svojstvenih** vrijednosti ovog problema

$$\frac{P}{EI} = \lambda_n = \left(\frac{n\pi}{L}\right)^2, \quad n = 1, 2, \dots,$$

tj., za “svojstvena” opterećenja

$$P_n = \left(\frac{n\pi}{L}\right)^2 EI, \quad n = 1, 2, \dots,$$

a pripadno rješenje je onda $y_n = c \sin(n\pi x/L)$, gdje je c proizvoljna konstanta (ne može se odrediti iz ovog modela). U tom slučaju, može doći do izvijanja grede u oblik dan funkcijom y_n , a $y = 0$ je nestabilni ravnotežni položaj grede.

Najmanja vrijednost opterećenja P kod kojeg se to događa je $P_1 = (\pi/L)^2 EI$, i naziva se **kritično** ili Eulerovo opterećenje, jer tada dolazi do izvijanja, odnosno, pucanja grede.

Napišite program za približno nalaženje prvih nekoliko svojstvenih opterećenja P_n i pripadnih svojstvenih funkcija y_n ovog problema. Iskoristite standardne metode diskretizacije i riješite kao svojstveni problem za matricu dobivenu diskretizacijom.

Za zadani $N \in \mathbb{N}$, segment $[0, L]$ podijelimo uniformnom mrežom na $N + 1$ podintervala duljine $h = L/(N + 1)$, tako da su čvorovi mreže

$$x_i = ih, \quad i = 0, \dots, N + 1.$$

Uvedimo još skraćenu oznaku indeksom za funkcijske vrijednosti bilo koje funkcije f na \mathbb{R} , tako da je $f_\alpha := f(\alpha h)$, za bilo koji realni $\alpha \in \mathbb{R}$.

Ako iskoristimo standardnu diskretizaciju druge derivacije drugim centralnim razlikama u unutarnjim čvorovima mreže, dobivamo sustav jednažbi

$$-y_{i-1} + 2y_i - y_{i+1} = \frac{P}{EI} h^2 y_i,$$

za $i = 1, \dots, N$. Nakon uvrštavanja rubnih uvjeta $y_0 = y_{N+1} = 0$, dobivamo “obični” svojstveni problem za simetričnu pozitivno definitnu tridijagonalnu matricu T_N reda N .

Najmanje svojstvene vrijednosti matrice T_N i pripadni svojstveni vektori su razumno dobra aproksimacija za odgovarajuće svojstvene vrijednosti i funkcije polaznog problema. Za standardnu diskretizaciju brzina konvergencije je $O(h^2)$.

Testirajte kvalitetu dobivenih svojstvenih vrijednosti i funkcija u ovisnosti o finoći diskretizacije. Možete koristiti i bolje diskretizacije, ali dobivena matrica može imati više punih dijagonala. Za testiranje možete uzeti da je $L = E = I = 1$.

Cijeli problem možemo promatrati i kao svojstveni problem za jednažbu ravnoteže štapa, što je diferencijalna jednažba četvrtog reda. U opisanoj situaciji, s tim da greda odgovara štapu, dobivamo problem oblika

$$(EIy'')'' + Py'' = 0,$$

s rubnim uvjetima

$$y(0) = y''(0) = 0, \quad y(L) = y''(L) = 0.$$

Homogeni rubni uvjeti na druge derivacije znače da greda može slobodno rotirati u rubovima, iako je učvršćena na progib.

I ovaj problem se može diskretizirati, s tim da se traže vrijednosti parametra P za koje problem ima netrivialna (diskretizirana) rješenja za y . Uzmite da su E i I konstantni. Kako izgleda pripadni algebarski svojstveni problem za nalaženje parametara P ? Pokušajte riješiti dobiveni algebarski problem i usporediti rezultate s prethodnima.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Original: tekst i slika preuzeti su iz knjige

- B. Carnahan, H. A. Luther, J. O. Wilkes, “Applied Numerical Methods”, J. Wiley & Sons, New York, 1969., Problem 4.27, 4.28, str. 265–266.

Više o jednažbi ravnoteže štapa možete pogledati u knjizi

- I. Aganović, K. Veselić, “Jednažbe matematičke fizike”, Školska knjiga, Zagreb, 1985., Poglavlja 15. i 20.

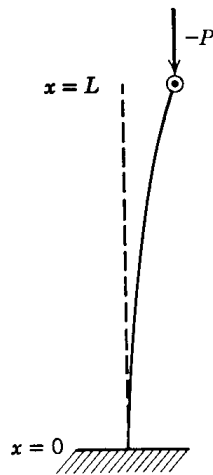
Zadatak 21. Svojstveni problem — jednostrano stisnuta greda.

Poprečni (transverzalni) otklon y elastične savitljive potporne grede (ili nosivog stupa) zadovoljava diferencijalnu jednadžbu oblika

$$\frac{d^2y}{dx^2} = \frac{M}{EI},$$

gdje je x udaljenost duž osi grede, E je Youngov modul elastičnosti, I je odgovarajući moment inercije poprečnog presjeka grede, a M je lokalni moment izvijanja (zakretni moment) pod utjecajem vanjske uzdužne (longitudinalne) sile ili opterećenja. Pretpostavljamo da greda ima konstantni presjek po x osi, i da je homogena, tako da su E i I konstantni.

Uspravna greda duljine L ukliještena je na donjem kraju tako da na dnu nema otklona i zakreta grede, a gornji rub je potpuno slobodan na progib (i zakret). Takvu gredu uzdužno “stišćemo” na gornjem kraju vanjskom silom (opterećenjem) P , kao na sljedećoj slici.



Za lokalni moment izvijanja onda vrijedi $M(x) = -P(y(x) - y(L))$, pa “ravnotežni” otklon grede zadovoljava jednadžbu

$$-\frac{d^2y}{dx^2} = \frac{P}{EI}(y - y(L)).$$

Za dovoljno mala opterećenja P , ovaj problem ima samo trivijalno rješenje $y = 0$. Netrivijalna rješenja ove jednadžbe postoje samo ako je faktor na desnoj strani jednak nekoj od **svojstvenih** vrijednosti ovog problema

$$\frac{P}{EI} = \lambda_n = \left(\frac{(2n-1)\pi}{2L} \right)^2, \quad n = 1, 2, \dots,$$

tj., za “svojstvena” opterećenja

$$P_n = \left(\frac{(2n-1)\pi}{2L} \right)^2 EI, \quad n = 1, 2, \dots,$$

a pripadno rješenje je onda

$$y_n = c \left(1 - \cos \frac{(2n-1)\pi x}{2L} \right),$$

gdje je c proizvoljna konstanta (ne može se odrediti iz ovog modela). U tom slučaju, može doći do izvijanja grede u oblik dan funkcijom y_n , a $y = 0$ je nestabilni ravnotežni položaj grede.

Najmanja vrijednost opterećenja P kod kojeg se to događa je $P_1 = (\pi/(2L))^2 EI$, i naziva se **kritično** ili Eulerovo opterećenje, jer tada dolazi do izvijanja, odnosno, pucanja grede.

Napišite program za približno nalaženje prvih nekoliko svojstvenih opterećenja P_n i pripadnih svojstvenih funkcija y_n ovog problema. Iskoristite standardne metode diskretizacije i riješite kao svojstveni problem za matricu dobivenu diskretizacijom.

Za zadani $N \in \mathbb{N}$, segment $[0, L]$ podijelimo uniformnom mrežom na $N + 1$ podintervala duljine $h = L/(N + 1)$, tako da su čvorovi mreže

$$x_i = ih, \quad i = 0, \dots, N + 1.$$

Uvedimo još skraćenu oznaku indeksom za funkcijske vrijednosti bilo koje funkcije f na \mathbb{R} , tako da je $f_\alpha := f(\alpha h)$, za bilo koji realni $\alpha \in \mathbb{R}$.

Ako iskoristimo standardnu diskretizaciju druge derivacije drugim centralnim razlikama u unutarnjim čvorovima mreže, uz $y_{N+1} = y(L)$, dobivamo sustav jednačbi

$$-y_{i-1} + 2y_i - y_{i+1} = \frac{P}{EI} h^2 (y_i - y_{N+1}),$$

za $i = 1, \dots, N$. Rubni uvjet $y'(0) = 0$ diskretiziramo tako da dodamo još jedan fiktivni čvor x_{-1} , a gornju jednačbu napišemo i za $i = 0$ (diskretizacija u rubu $x_0 = 0$). Onda je

$$y'(0) \approx \frac{y_1 - y_{-1}}{2h} = 0,$$

odakle slijedi $y_{-1} = y_1$. Kad to uvrstimo u jednačbu za $i = 0$, izlazi

$$2y_0 - 2y_1 = \frac{P}{EI} h^2 (y_0 - y_{N+1}).$$

U nastavku svođenja na algebarski problem svojstvenih vrijednosti treba biti oprezan, jer direktni put ne vodi na korektan oblik problema.

Ako odmah uvrstimo i rubni uvjet $y_0 = 0$ u prethodnu jednačbu, dobivamo

$$-2y_1 = -\frac{P}{EI} h^2 y_{N+1},$$

pa se član s nepoznatim y_{N+1} može eliminirati iz desne strane svih ostalih jednačbi za $i = 1, \dots, N$, s tim da dobivamo $2y_1$ na lijevoj strani. Kad uvrstimo $y_0 = 0$ u jednačbu za $i = 1$, dobivene jednačbe više ne sadrže y_0 . Međutim, zadnja jednačba za $i = N$ još uvijek na **lijevoj** strani sadrži nepoznati y_{N+1} , kojeg ne možemo eliminirati preko y_1 , ako želimo dobiti korektan oblik svojstvenog problema s vektorom nepoznanica y_1, \dots, y_N . To se može izbjeći tako da u točki x_N napravimo drugačiju diskretizaciju jednačbe, koristeći vrijednosti funkcije u **prednjim** čvorovima $x_N, x_{N-1}, x_{N-2}, \dots$. Možete probati, ako želite!

Umjesto toga, koristimo "trik" koji odgovara homogenizaciji rubnog uvjeta u točki $x = L$, tako da uvedemo supstituciju $z_i = y_i - y_{N+1}$, za $i = 0, \dots, N + 1$. Jednačba za $i = 0$ postaje

$$2z_0 - 2z_1 = \frac{P}{EI} h^2 z_0,$$

a ostale jednadžbe imaju oblik

$$-z_{i-1} + 2z_i - z_{i+1} = \frac{P}{EI} h^2 z_i, \quad i = 1, \dots, N.$$

Uvrstimo $z_{N+1} = 0$ u zadnju jednadžbu, i dobivamo svojstveni problem za tridijagonalnu matricu T_{N+1} , reda $N + 1$, s nepoznatim vrijednostima z_0, \dots, z_N u vektoru. Kad nađemo pripadne svojstvene vektore, onda iskoristimo rubni uvjet $y_0 = 0$, pa je $z_0 = -y_{N+1}$, a zatim izračunamo sve ostale komponente $y_i = z_i - z_0$. Uočite da T_{N+1} nije simetrična matrica, ali se može simetrizirati dijagonalnim sličnostima, na isti način kao kod ortogonalnih polinoma.

Najmanje svojstvene vrijednosti matrice T_{N+1} i pripadni svojstveni vektori su razumno dobra aproksimacija za odgovarajuće svojstvene vrijednosti i funkcije polaznog problema. Za standardnu diskretizaciju brzina konvergencije je $O(h^2)$.

Testirajte kvalitetu dobivenih svojstvenih vrijednosti i funkcija u ovisnosti o finoći diskretizacije. Možete koristiti i bolje diskretizacije, ali dobivena matrica može imati više punih dijagonala. Za testiranje možete uzeti da je $L = E = I = 1$.

Cijeli problem možemo promatrati i kao svojstveni problem za jednadžbu ravnoteže štapa, što je diferencijalna jednadžba četvrtog reda. U opisanoj situaciji, s tim da greda odgovara štapu, dobivamo problem oblika

$$(EIy'')'' + Py'' = 0,$$

s rubnim uvjetima

$$y(0) = y'(0) = 0, \quad y''(L) = 0, \quad EIy'''(L) + Py'(L) = 0.$$

Rubni uvjeti za $x = 0$ znače da je greda učvršćena na progib i rotaciju, a rubni uvjeti za $x = L$ znače da je greda slobodna na progib i rotaciju.

I ovaj problem se može diskretizirati, s tim da se traže vrijednosti parametra P za koje problem ima netrivialna (diskretizirana) rješenja za y . Uzmite da su E i I konstantni. Kako izgleda pripadni algebarski svojstveni problem za nalaženje parametara P ? Pokušajte riješiti dobiveni algebarski problem i usporediti rezultate s prethodnima.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Original: tekst i slika preuzeti su iz knjige

- B. Carnahan, H. A. Luther, J. O. Wilkes, “Applied Numerical Methods”, J. Wiley & Sons, New York, 1969., Problem 4.29, str. 266.

Više o jednadžbi ravnoteže štapa možete pogledati u knjizi

- I. Aganović, K. Veselić, “Jednadžbe matematičke fizike”, Školska knjiga, Zagreb, 1985., Poglavlja 15. i 20.

Zadatak 22. Svojtstveni problem — transverzalne oscilacije učvršćenog štapa.

Poprečni (transverzalni) otklon u štapa, sasvim općenito, ovisi o vremenu t i položaju x na centralnoj liniji nedeformiranog štapa. Funkcija u zadovoljava parcijalnu diferencijalnu jednadžbu oblika

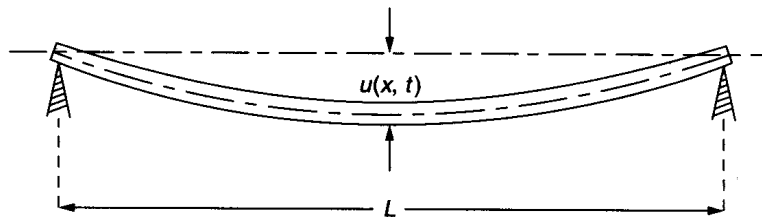
$$\frac{\partial^2}{\partial x^2} \left(EI \frac{\partial^2 u}{\partial x^2} \right) = -\rho_x \frac{\partial^2 u}{\partial t^2}.$$

gdje je E Youngov modul elastičnosti, I je odgovarajući moment inercije poprečnog presjeka štapa, a ρ_x je linijska gustoća štapa (masa po jedinici duljine). Sve tri veličine, općenito, mogu ovisiti o položaju x . Ovoj jednadžbi treba dodati odgovarajuće rubne i početne uvjete, koji onda određuju otklon štapa.

Iz prakse znamo da hodanje preko daske (ili mosta) uzrokuje vertikalne vibracije. Zato je bitno pronaći tzv. “vlastite” frekvencije prirodnih oscilacija štapa, tako da se izbjegne pojava rezonancije. Njih nalazimo tako da uvrstimo rješenje oblika $u(x, t) = y(x)e^{i\omega t}$ i tražimo sve kružne frekvencije ω za koje dobivena jednadžba za funkciju y ima netrivialno rješenje. Ta jednadžba 4. reda ima oblik svojtstvenog problema za jednadžbu štapa

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 y}{dx^2} \right) = \rho_x \omega^2 y.$$

Horizontalni štap duljine L (u x smjeru) učvršćen je na oba kraja tako da u rubovima nema otklona (progiba), ali svaki rub može slobodno rotirati oko točke učvršćenja, kao na sljedećoj slici.



Rubni uvjeti svojtstvenog problema su

$$y(0) = y''(0) = 0, \quad y(L) = y''(L) = 0.$$

Za početak, pretpostavljamo da štap ima konstantni presjek S po x osi, i da je homogen, tako da su E , I i ρ_x konstantni. Jednadžbu tada možemo napisati u obliku

$$\frac{d^4 y}{dx^4} = \frac{\rho_x}{EI} \omega^2 y.$$

Za dovoljno male frekvencije ω , ovaj problem ima samo trivijalno rješenje $y = 0$. Netrivijalna rješenja ove jednadžbe postoje samo ako je faktor na desnoj strani jednak nekoj od **svojtstvenih** vrijednosti ovog problema

$$\frac{\rho_x}{EI} \omega^2 = \lambda_n = \left(\frac{n\pi}{L} \right)^4, \quad n = 1, 2, \dots,$$

tj., za “svojtstvene” ili vlastite frekvencije

$$\omega_n = \left(\frac{n\pi}{L} \right)^2 \cdot \sqrt{\frac{EI}{\rho_x}}, \quad n = 1, 2, \dots,$$

a pripadno rješenje je onda $y_n = c \sin(n\pi x/L)$, gdje je c proizvoljna konstanta. Ako na štap djelujemo vanjskom silom koja izaziva oscilacije nekom od ovih frekvencija, dolazi do rezonancije (sve većih oscilacija i pucanja). U praksi treba naći samo nekoliko najmanjih vlastitih frekvencija ω_n (ostale su prevelike da bi se mogle prirodno izazvati).

Najmanja vrijednost frekvencije ω kod koje se to događa je $\omega_1 = (\pi/L)^2 \sqrt{EI/\rho_x}$ i naziva se **kritična** ili osnovna frekvencija štapa.

Napišite program za približno nalaženje prvih nekoliko vlastitih frekvencija ω_n i pripadnih svojstvenih funkcija y_n ovog problema. Iskoristite standardne metode diskretizacije i riješite kao svojstveni problem za matricu dobivenu diskretizacijom.

Za zadani $N \in \mathbb{N}$, segment $[0, L]$ podijelimo uniformnom mrežom na $N+1$ podintervala duljine $h = L/(N+1)$, tako da su čvorovi mreže

$$x_i = ih, \quad i = 0, \dots, N+1.$$

Uvedimo još skraćenu oznaku indeksom za funkcijske vrijednosti bilo koje funkcije f na \mathbb{R} , tako da je $f_\alpha := f(\alpha h)$, za bilo koji realni $\alpha \in \mathbb{R}$.

Ako iskoristimo standardnu diskretizaciju četvrte derivacije četvrtim centralnim razlikama u unutarnjim čvorovima mreže (s greškom $O(h^2)$), dobivamo sustav jednačbi

$$y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2} = \frac{\rho_x}{EI} \omega^2 h^4 y_i, \quad i = 1, \dots, N.$$

Ove jednačbe, osim rubnih, sadrže još i vrijednosti funkcije u dva fiktivna vanjska čvora x_{-1} i x_{N+2} . Njih eliminiramo odgovarajućom diskretizacijom rubnih uvjeta koji sadrže derivacije funkcije y .

Iz rubnih uvjeta $y(0) = y(L) = 0$ odmah dobivamo $y_0 = y_{N+1} = 0$. Rubne uvjete $y''(0) = 0$ i $y''(L) = 0$ diskretiziramo drugom centralnom razlikom u rubnim čvorovima mreže (greška je opet $O(h^2)$)

$$y''(0) \approx \frac{y_{-1} - 2y_0 + y_1}{h^2} = 0, \quad y''(L) \approx \frac{y_N - 2y_{N+1} + y_{N+2}}{h^2} = 0,$$

odakle slijedi $y_{-1} = -y_1$ i $y_{N+2} = -y_N$. Nakon uvrštavanja ovih “rubnih” vrijednosti u jednačbe sustava, dobivamo “obični” svojstveni problem za simetričnu peterodijagonalnu matricu T_N , reda N ,

$$T_N = \begin{bmatrix} 5 & -4 & 1 & & & & & \\ -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & 1 & -4 & 6 & -4 & 1 & \\ & & & 1 & -4 & 6 & -4 & \\ & & & & 1 & -4 & 5 & \end{bmatrix}.$$

Najmanje svojstvene vrijednosti matrice T_N i pripadni svojstveni vektori su razumno dobra aproksimacija za odgovarajuće svojstvene vrijednosti i funkcije polaznog problema. Za standardnu diskretizaciju brzina konvergencije je $O(h^2)$.

Testirajte kvalitetu dobivenih svojstvenih vrijednosti i funkcija u ovisnosti o finoći diskretizacije. Možete koristiti i bolje diskretizacije, ali dobivena matrica može imati više punih dijagonala. Za testiranje možete uzeti da je $L = E = I = \rho_x = 1$.

Pretpostavimo sad da štap ima pravokutan presjek okomito na centralnu liniju štapa, ali površina tog presjeka $S(x)$ više **nije** konstantna duž štapa. Štap ima konstantnu širinu b okomito na ravninu otklona (tj. okomito na papir na slici), i varijabilnu debljinu d u ravnini otklona (okomito na centralnu liniju, u ravnini papira na slici). Za površinu presjeka i moment inercije I tog presjeka onda vrijedi

$$S(x) = b \cdot d(x), \quad I(x) = \frac{1}{12} b \cdot (d(x))^3.$$

Dodatno, uzmimo da debljina d varira linearno duž štapa, tako da je

$$d(x) = d_0 + (d_L - d_0) \frac{x}{L}, \quad x \in [0, L],$$

gdje su d_0 i d_L zadane konstante. I dalje pretpostavljamo da je štap homogen, od materijala konstantne (volumne) gustoće ρ , tako da je i E konstantan. Linijska gustoća štapa je onda

$$\rho_x = S(x)\rho.$$

Jednadžbu za vlastite frekvencije ω tada možemo napisati u obliku

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 y}{dx^2} \right) = S \rho \omega^2 y,$$

s tim da su I i S funkcije od x .

Napravite diskretizaciju ovog problema tako da obje druge derivacije diskretizirate drugom centralnom razlikom u odgovarajućim čvorovima mreže. Kako izgleda pripadni algebarski svojstveni problem za nalaženje vlastitih frekvencija ω ? Pokušajte riješiti dobiveni algebarski problem. Za testiranje možete uzeti $E = L = \rho = 1$, $b = 0.1$, a za d_0 i d_L uzmite prvo iste, a zatim različite vrijednosti, nekoliko puta **manje** od b . Odnos veličina L , b i d odgovara ne preširokoj dugačkoj dasci varijabilne debljine (“najmanje” dimenzije).

Ako želite probati, razumne stvarne vrijednosti za štap od lake metalne legure (poput aluminija) su: $E = 6.9 \cdot 10^{10} \text{ N/m}^2$, $\rho = 2.7 \cdot 10^3 \text{ kg/m}^3$. Dimenzije štapa izaberite po želji, samo da je L bitno veći od b , a smisleno je i $b > d_0, d_L$.

Literatura: predavanja “Iterativne metode” i “Dodaci” za školsku godinu 2008./09. (dostupno na web stranici prof. Singera).

Original: tekst je preuzet iz knjige

- B. Carnahan, H. A. Luther, J. O. Wilkes, “Applied Numerical Methods”, J. Wiley & Sons, New York, 1969., Problem 4.31, str. 266.

Slika je preuzeta iz knjige

- A. Jeffrey, “Applied Partial Differential Equations”, Academic Press, Dan Diego, 2003., str. 263.

Više o jednadžbi štapa možete pogledati u knjizi

- I. Aganović, K. Veselić, “Jednadžbe matematičke fizike”, Školska knjiga, Zagreb, 1985., Poglavlja 15., 16. i 20.

Zadatak 23. Iterativne metode za linearne sustave — metoda kvazi–minimalnog reziduala (QMR)

Metoda kvazi–minimalnog reziduala (QMR) je metoda slična BCG metodi, bazirana na dvostranom Lanczosovom algoritmu, koja se primijenjuje za rješavanje nehermitskih linearnih sustava, ali koja je u stanju prevladati probleme BCG-a. U QMR metodi ponovo se uzima da je aproksimacija x_k oblika

$$x_k = x_0 + V_k y_k,$$

samo što se y_k bira tako da minimizira kvazi–rezidual, vrijednost usko povezanu sa euklidskom normom reziduala. Budući da je $r_k = r_0 - AV_k y_k$, kao i kod BCG metode dobiva se da je

$$r_k = V_{k+1}(\beta \xi_1 - T_{k+1,k} y_k),$$

tako da norma reziduala r_k zadovoljava

$$\|r_k\|_2 \leq \|V_{k+1}\|_2 \|\beta \xi_1 - T_{k+1,k} y_k\|_2.$$

Budući da stupci od V_{k+1} nisu ortogonalni, rješavanje problema najmanjih kvadrata $\min_{y \in \mathbb{C}^k} \|r_0 - V_{k+1} T_{k+1,k} y\|_2$ zahtijevalo bi previše operacija. Stoga ćemo umjesto toga minimizirati drugi faktor u prethodnoj nejednakosti. Svi stupci matrice V_{k+1} imaju normu jedan, zato prvi faktor zadovoljava

$$\|V_{k+1}\|_2 \leq \|V_{k+1}\|_F \leq \sqrt{k+1}.$$

Dakle, y_k u QMR metodi rješava problem najmanjih kvadrata

$$\min_{y \in \mathbb{C}^k} \|\beta \xi_1 - T_{k+1,k} y\|_2,$$

koji uvijek ima rješenje. Naime, matrica $T_{k+1,k}$ je $(k+1) \times k$ tridijagonalna matrica koja na donjoj sporednoj dijagonali, ukoliko se dvostrani Lanczosov algoritam nije zbog nekog razloga zaustavio, ima elemente $\gamma_j > 0$ za $j = 1, \dots, k$, pa je ona punog ranga. Što više rješnje je jedinstveno. Rješavanje prethodno definiranog problema najmanjih kvadrata provodi se dalje kao i kod GMRES metode.

Algoritam 3 (Metoda kvazi–minimalnog reziduala (QMR))

x_0 i \hat{r}_0 zadani;

$$r_0 = b - Ax_0;$$

$$\beta = \|r_0\|_2;$$

$$v_1 = \frac{r_0}{\beta};$$

$$w_1 = \frac{\hat{r}_0}{\|\hat{r}_0\|_2};$$

$$l = [1, 0, \dots, 0]^T;$$

while (!kriterij_zaustavljanja){

Izračunaj v_{k+1} , w_{k+1} , $\alpha_k = T(k, k)$, $\beta_k = T(k, k+1)$, i $\gamma_k = T(k+1, k)$, koristeći dvostrani Lanczosov algoritam;

Primijeni Givensove rotacije F_{k-2} i F_{k-1} na zadnji stupac od T , odnosno:

$$\text{ako je } k > 2 \text{ tada } \begin{bmatrix} T(k-2, k) \\ T(k-1, k) \end{bmatrix} := \begin{bmatrix} c_{k-2} & s_{k-2} \\ -\bar{s}_{k-2} & c_{k-2} \end{bmatrix} \begin{bmatrix} 0 \\ T(k-1, k) \end{bmatrix};$$

$$\text{ako je } k > 1 \text{ tada } \begin{bmatrix} T(k-1, k) \\ T(k, k) \end{bmatrix} := \begin{bmatrix} c_{k-1} & s_{k-1} \\ -\bar{s}_{k-1} & c_{k-1} \end{bmatrix} \begin{bmatrix} T(k-1, k) \\ T(k, k) \end{bmatrix};$$

Izračunaj k -tu Givensovu rotaciju F_k kako bi se poništio $(k+1, k)$ element od T :

$$c_k = \frac{|T(k, k)|}{\sqrt{|T(k, k)|^2 + |T(k+1, k)|^2}};$$

$$\text{ako je } c_k \neq 0 \text{ tada } s_k = c_k \frac{T(k+1, k)}{T(k, k)}, \text{ ako je } c_k = 0 \text{ tada } s_k = 1;$$

Primijeni k -tu rotaciju na l i na zadnji stupac od T :

$$\begin{bmatrix} l(k) \\ l(k+1) \end{bmatrix} := \begin{bmatrix} c_k & s_k \\ -\bar{s}_k & c_k \end{bmatrix} \begin{bmatrix} l(k) \\ 0 \end{bmatrix};$$

$$T(k, k) := c_k T(k, k) + s_k T(k+1, k);$$

$$T(k+1, k) = 0 \quad (*);$$

Izračunaj $p_{k-1} = (1/T(k, k))[v_k - T(k-1, k)p_{k-2} - T(k-2, k)p_{k-3}]$, gdje su p_{-1}, p_{-2} jednaki nuli;

$$x_k = x_{k-1} + \beta l(k) p_{k-1};$$

$$r_k = r_{k-1} - \beta l(k) A p_{k-1};$$

}

(*) $T(k+1, k) = 0$ treba zapravo izvesti u sljedećem, $(k+1)$ -om koraku, jer je originalna vrijednost $T(k+1, k)$ potrebna za izvođenje $(k+1)$ -og koraka dvostranog Lanczosovog algoritma.

U kojim slučajevima bi moglo doći do sloma ove metode? Koji uvjet nam garantira da će metoda završiti za $m \leq n$ iteracija pri čemu je n dimenzija sustava?

Napišite potprograme koji implementiraju dvostrani Lanczosov algoritam i QMR metodu i primijenite ih na raznim matricama. Najprije izvršite potprogram za QMR metodu, koji treba završiti za $m \leq n$ iteracija bilo zbog sloma dvostranog Lanczosovog algoritma, bilo zbog konvergencije. Zatim provjerite da je $W_m^T A V_m = T_m$ i $W_m^T V_m = I$. Usporedite normu reziduala i brzinu konvergencije QMR metode s GMRES metodom. Posebno pronađite barem jedan primjer u kojem dolazi do sloma metode, i barem jedan primjer u kojem je konvergencija dobra kao i kod GMRES metode. Prokomentirajte sve dobivene rezultate i brzine konvergencije.

Literatura: Nela Bosner, "Iterativne metode za rješavanje linearnih sustava", magistarski rad (dostupno na mojoj web stranici)

Zadatak 24. Iterativne metode za linearne sustave — metoda kvadriranih konjugiranih gradijenata (CGS)

Metode BCG i QMR zahtijevaju množenje vektora sa matricom A ali i sa matricom A^* . To zahtijeva dodatni posao, a naročito kada je kompliciranije množiti sa A^* nego sa A . Zbog toga je poželjno imati iterativnu metodu koja će zahtijevati samo množenje sa matricom A , a neće biti puno zahtjevnija od BCG metode. Takva metoda je *metoda kvadriranih konjugiranih gradijenata* (Conjugate gradient squared method) ili CGS metoda.

Krenut ćemo od BCG algoritma, odakle možemo primijetiti da vrijedi

$$\begin{aligned} r_k &= \phi_k(A)r_0, & \hat{r}_k &= \bar{\phi}_k(A^*)\hat{r}_0, \\ p_k &= \psi_k(A)r_0, & \hat{p}_k &= \bar{\psi}_k(A^*)\hat{r}_0 \end{aligned}$$

za određene polinome k -tog stupnja ϕ_k i ψ_k . Ako algoritam dobro konvergira, tada je $\|\phi_k(A)r_0\|_2$ mala, pa bi mogli očekivati da je $\|\phi_k^2(A)r_0\|_2$ još manja. Ako još pokušamo izračunati $\phi_k^2(A)r_0$ sa otprilike jednako mnogo operacija kao i $\phi_k(A)r_0$ tada bi to najvjerojatnije bio brzo konvergirajući algoritam. To je bila osnovna ideja razvoja CGS metode.

Ako definiramo polinome

$$\Phi_k(z) = \phi_k^2(z), \quad \Theta_k(z) = \phi_k(z)\psi_{k-1}(z), \quad \Psi_k(z) = \psi_k^2(z),$$

iz rekurzija za BCG metodu možemo izvesti rekurzije

$$\begin{aligned} \Phi_k(z) &= \Phi_k(z) - 2\alpha_{k-1}z(\Phi_{k-1}(z) + \beta_{k-2}\Theta_{k-1}(z)) + \alpha_{k-1}^2z^2\Psi_{k-1}(z), \\ \Theta_k(z) &= \Phi_{k-1}(z) + \beta_{k-2}\Theta_{k-1}(z) - \alpha_{k-1}z\Psi_{k-1}(z), \\ \Psi_k(z) &= \Phi_k(z) + 2\beta_{k-1}\Theta_k(z) + \beta_{k-1}^2\Psi_{k-1}(z). \end{aligned}$$

Ove rekurzije su osnova algoritma. Ako sada definiramo nove vrijednosti

$$\begin{aligned} r_k &= \Phi_k(A)r_0, \\ q_k &= \Theta_k(A)r_0, \\ p_k &= \Psi_k(A)r_0, \end{aligned}$$

tada gornje rekurzije za polinome prelaze u

$$\begin{aligned} r_k &= r_{k-1} - \alpha_{k-1}A(2r_{k-1} + 2\beta_{k-2}q_{k-1} - \alpha_{k-1}Ap_{k-1}), \\ q_k &= r_{k-1} + \beta_{k-2}q_{k-1} - \alpha_{k-1}Ap_{k-1}, \\ p_k &= r_k + 2\beta_{k-1}q_k + \beta_{k-1}^2p_{k-1}. \end{aligned}$$

Ako uvedemo još jedan pomoćni vektor

$$u_{k-1} = r_{k-1} + \beta_{k-2}q_{k-1},$$

tada možemo izvesti konačan oblik CGS metode.

Algoritam 4 (Metoda kvadriranih konjugiranih gradijenata (CGS))

x_0 i \hat{r}_0 zadani, takvi da je $\hat{r}_0^T r_0 \neq 0$;

$$r_0 = b - Ax_0;$$

$$p_0 = r_0$$

$$u_0 = r_0;$$

```

while (!kriterij_zaustavljanja) {
     $\alpha_{k-1} = \frac{\hat{r}_0^T r_{k-1}}{\hat{r}_0^T A p_{k-1}};$ 
     $q_k = u_{k-1} - \alpha_{k-1} A p_{k-1};$ 
     $x_k = x_{k-1} + \alpha_{k-1} (u_{k-1} + q_k);$ 
     $r_k = r_{k-1} - \alpha_{k-1} A (u_{k-1} + q_k);$ 

     $\beta_{k-1} = \frac{\hat{r}_0^T r_k}{\hat{r}_0^T r_{k-1}};$ 
     $u_k = r_k + \beta_{k-1} q_k;$ 
     $p_k = u_k + \beta_{k-1} (q_k + \beta_{k-1} p_{k-1});$ 
}

```

U kojim slučajevima bi moglo doći do sloma ove metode? Koji uvjet nam garantira da će metoda završiti za $m \leq n$ iteracija pri čemu je n dimenzija sustava?

Napišite potprograme koji implementiraju dvostrani Lanczosov algoritam i CGS metodu i primijenite ih na raznim matricama. Najprije izvršite potprogram za CGS metodu, koji treba završiti za $m \leq n$ iteracija bilo zbog sloma, bilo zbog konvergencije. Zatim izračunajte T_m , V_m i W_m pomoću dvostranog Lanczosovog algoritma. Provjerite da je $W_m^T A V_m = T_m$. Usporedite brzinu konvergencije CGS metode s brzinom konvergencije BCG metode (za BCG možete upotrijebiti gotovu rutinu). Kako se ponaša norma reziduala tokom iteracija? Posebno pronađite barem jedan primjer u kojem dolazi do sloma metode, i barem jedan primjer u kojem je konvergencija dobra kao i kod GMRES metode. Prokomentirajte sve dobivene rezultate i brzine konvergencije.

Literatura: Nela Bosner, “Iterativne metode za rješavanje linearnih sustava”, magistarski rad (dostupno na mojoj web stranici)

Zadatak 25. Iterativne metode za linearne sustave — metoda stabiliziranih bikonjugiranih gradijenata (BICGSTAB)

CGS metoda je bazirana na kvadriranju rezidualnog polinoma BCG metode, i u slučaju neregularne konvergencije, ona može dovesti do značajnih grešaka zaokruživanja, pa čak može proizvesti i vrijednosti koje su izvan raspoloživog raspona u skupu strojnih brojeva. Naime, kada je norma BCG reziduala mala, norma CGS reziduala je obično još puno manja, ali ako je norma BCG reziduala velika, norma CGS reziduala je još veća, pa su oscilacije kod CGS puno veće nego kod BCG metode. *Metoda stabiliziranih bikonjugiranih gradijenata* (BICGSTAB) je varijacija CGS metode koja je bila razvijena s ciljem da popravi ovaj problem.

U CGS metodi definirali smo rekurzije tako da rezidual r_k zadovoljava jednakost $r_k = \phi_k(A)^2 r_0$, kod kojeg je $\phi_k(A)r_0$ rezidual BCG metode. Naravno, mi možemo konstruirati druge iterativne metode, za koje su aproksimacije x_k generirane tako da je

$$r_k = \chi_k(A)\phi_k(A)r_0,$$

gdje je ϕ_k ponovo BCG rezidualni polinom, a χ_k je izabran tako da pokuša držati normu reziduala malom u svakom koraku, ali s druge strane, da zadrži globalnu brzu konvergenciju. Jedna mogućnost izbora polinoma χ_k je polinom oblika

$$\chi_k = (1 - \omega_k z)(1 - \omega_{k-1} z) \cdots (1 - \omega_1 z),$$

pri čemu koeficijenti ω_j mogu biti izabrani tako da u svakom koraku minimiziraju

$$\|r_k\|_2 = \|(I - \omega_k A)\chi_{k-1}(A)\phi_k(A)r_0\|_2.$$

Ovo je glavna ideja BICGSTAB metode, koja se može smatrati kombinacijom metoda BCG i MINRES.

Ponovo, neka $\phi(A)r_0$ označava BCG rezidual u k -tom koraku, a $\psi_k(A)r_0$ neka označava BCG vektor smjera p_k , i neka oba polinoma zadovoljavaju rekurzije BCG metode. U BICGSTAB algoritmu želimo naći rekurzije za

$$r_k = \chi_k(A)\phi_k(A)r_0$$

i

$$p_k = \chi_k(A)\psi_k(A)r_0.$$

Nadalje, u svakoj iteraciji trebamo još odrediti parametar ω_k . Najjednostavniji izbor bi bio da ω_k minimizira euklidsku normu vektora $(I - \omega_k A)\chi_{k-1}(A)\phi_k(A)r_0$. Rekurziju za r_k tada možemo drugačije napisati kao

$$r_k = (I - \omega_k A)r_{k-1/2},$$

pri čemu je

$$r_{k-1/2} = \chi_{k-1}(A)(\phi_{k-1}(A) - \alpha_{k-1}A\psi_{k-1}(A))r_0 = r_{k-1} - \alpha_{k-1}Ap_{k-1}.$$

Nakon minimiziranja funkcije $f(\omega_k) = \|(I - \omega_k A)r_{k-1/2}\|_2^2$, kao kod MINRES, dobivamo optimalni parametar ω_k dan sa

$$\omega_k = \frac{(Ar_{k-1/2})^T r_{k-1/2}}{(Ar_{k-1/2})^T Ar_{k-1/2}}.$$

Uzimajući sve ovo u obzir dobivamo efikasan algoritam za BICGSTAB metodu.

Algoritam 5 (Metoda stabiliziranih bikonjugiranih gradijenata (BICGSTAB)) x_0 i \hat{r}_0 zadani, takvi da je $\hat{r}_0^T r_0 \neq 0$;

$$r_0 = b - Ax_0;$$

$$p_0 = r_0$$

while (!kriterij_zaustavljanja){

$$\alpha_{k-1} = \frac{\hat{r}_0^T r_{k-1}}{\hat{r}_0^T Ap_{k-1}};$$

$$x_{k-1/2} = x_{k-1} + \alpha_{k-1} p_{k-1};$$

$$r_{k-1/2} = r_{k-1} - \alpha_{k-1} Ap_{k-1};$$

$$\omega_k = \frac{(Ar_{k-1/2})^T r_{k-1/2}}{(Ar_{k-1/2})^T Ar_{k-1/2}};$$

$$x_k = x_{k-1/2} + \omega_k r_{k-1/2};$$

$$r_k = r_{k-1/2} - \omega_k Ar_{k-1/2};$$

$$\beta_{k-1} = \frac{\alpha_{k-1} \hat{r}_0^T r_k}{\omega_k \hat{r}_0^T r_{k-1}};$$

$$p_k = r_k + \beta_{k-1}(p_{k-1} - \omega_k Ap_{k-1});$$

}

U kojim slučajevima bi moglo doći do sloma ove metode? Koji uvjet nam garantira da će metoda završiti za $m \leq n$ iteracija pri čemu je n dimenzija sustava?

Napišite potprograme koji implementiraju dvostrani Lanczosov algoritam i BICGSTAB metodu i primijenite ih na raznim matricama. Najprije izvršite potprogram za BICGSTAB metodu, koji treba završiti za $m \leq n$ iteracija bilo zbog sloma, bilo zbog konvergencije. Zatim izračunajte T_m , V_m i W_m pomoću dvostranog Lanczosovog algoritma. Provjerite da je $W_m^T AV_m = T_m$. Usporedite brzinu konvergencije BICGSTAB metode s brzinom konvergencije CGS metode (za CGS možete upotrijebiti gotovu rutinu). Kako se ponaša norma reziduala tokom iteracija? Posebno pronađite barem jedan primjer u kojem dolazi do sloma metode, i barem jedan primjer u kojem je konvergencija dobra kao i kod GMRES metode. Prokomentirajte sve dobivene rezultate i brzine konvergencije.

Literatura: Nela Bosner, "Iterativne metode za rješavanje linearnih sustava", magistrski rad (dostupno na mojoj web stranici)

Zadatak 26. Iterativne metode za linearne sustave — CGNR i CGNE metode

Postoji još jedan način na koji možemo pristupiti rješavanju nesimetričnog sustava, a to je simetrizacija problema. On se svodi na transformaciju nesimetričnog problema na simetrični, rješavanjem jedne od *normalnih jednadžbi*

$$A^T A x = A^T b \quad \text{ili} \quad A A^T \hat{x} = b, \quad x = A^T \hat{x}.$$

Rješavanje se može ostvariti bez eksplicitnog računanja hermitskih pozitivno definitnih matrica $A^T A$ ili $A A^T$, a u drugom slučaju nije niti potrebno generirati aproksimacije za \hat{x} , već se direktno računaju aproksimacije za x . Ako upotrijebimo CG metodu za rješavanje bilo kojeg od danih sustava tada odgovarajuće algoritme nazivamo CGNR za prvi, i CGNE za drugi sustav.

Algoritam 6 (CGNR i CGNE metode)

x_0 zadan;

$$r_0 = b - A x_0;$$

$$p_0 = A^T r_0$$

while (!kriterij_zaustavljanja){

$$\alpha_{k-1} = \frac{(A^T r_{k-1})^T A^T r_{k-1}}{(A^T p_{k-1})^T A^T p_{k-1}} \text{ za CGNR, } \alpha_{k-1} = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T p_{k-1}} \text{ za CGNE};$$

$$x_k = x_{k-1} + \alpha_{k-1} p_k - 1;$$

$$r_k = r_{k-1} - \alpha_{k-1} A p_{k-1};$$

$$\beta_k = \frac{(A^T r_k)^T A^T r_k}{(A^T r_{k-1})^T A^T r_{k-1}} \text{ za CGNR, } \beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \text{ za CGNE};$$

$$p_k = A^T r_k + \beta_k p_{k-1};$$

}

Koju normu greške ili reziduala minimiziraju metode CGNR i CGNE u svakoj iteraciji, i na kom prostoru? Što možete reći o točnosti rezultata u odnosu na CG ili GMRES metodu, obzirom na uvjetovanost sustava? Napišite potprograme koji implementiraju obje metode i primijenite ih na raznim matricama. Usporedite međusobno njihove brzine konvergencije i usporedite sa konvergencijom GMRES metode. Posebno pronađite barem jedan primjer u kojem jedna od danih metoda konvergira puno lošije od GMRES-a, i barem jedan primjer u kojem je konvergencija dobra kao i kod GMRES metode. Prokomentirajte sve dobivene rezultate i brzine konvergencije.

Literatura: Nela Bosner, “Iterativne metode za rješavanje linearnih sustava”, magistarski rad (dostupno na mojoj web stranici)

Zadatak 27. Iterativne metode za linearne sustave — metoda dekompozicije domene (nepreklapajuće domene)

Simulacijski se problemi često sastoje od kompliciranih struktura, kao na primjer trupovi aviona ili automobila. Zbog ograničenosti računalne memorije i vremena, ovakvi problemi se često ne mogu riješiti od jednom, već se problem razbija na više djelova koji se posebno rješavaju. Najbrže metode za metode, funkcioniraju najbolje na prilično regularnim problemima kao npr. rješavanje Poissonove jednadžbe definirane na pravilnim domenama, poput pravokutnika. S druge strane domena problema kojeg želimo riješiti može biti nepravilnog oblika. Tada se traži mogućnost razbijanja domene na manje dijelove na kojima problem nije kompliciran za rješavanje, i na kojima se može primijeniti jednostavnija metoda, recimo. Ako se rješenja potproblema mogu iskombinirati na pametan način, pri čemu bi se dobilo rješenje kompletnog problema, tada ćemo dobiti bržu i paralelizabilnu metodu za dobivanje rješenja, od primjene standardne iterativne metode na cijeli, veliki problem.

Metode sa nepreklapajućim poddomenama nazivaju se još *substrukturalne* ili *metode sa Schurovim komplementom*. Neka je \mathcal{L} diferencijalni operator definiran na domeni Ω , i pretpostavimo da želimo riješiti rubni problem

$$\begin{aligned} \mathcal{L}u &= f, & \text{na } \Omega, \\ u &= u_\Gamma & \text{na } \Gamma = \partial\Omega. \end{aligned}$$

Domena Ω je otvoreni skup u ravnini ili 3-dimenzionalnom prostoru, a $\partial\Omega$ označava rub od Ω . Ovdje smo uzeli Dirichletove rubne uvjete, ali mogu se isto tako staviti Neumannovi ili Robinovi rubni uvjeti.

Pretpostavimo da je naš problem bio diskretiziran centralnim diferencijama, i da je na domeni problema Ω definirana mreža. Pretpostavimo da je domena podijeljena na s poddomena Ω_i , $i = 1, \dots, s$, odnosno da vrijedi

$$\Omega = \bigcup_{i=1}^s \Omega_i,$$

pri čemu se poddomene ne preklapaju, što znači da susjedne domene dijele samo rub. Nazovimo rub između Ω_i i Ω_j , $i \neq j$ granicom Γ_{ij} između i -te i j -te poddomene. Te se granice mogu ili ne moraju poklapati sa bridovima mreže, ali općenito možemo pretpostaviti da svaka granica Γ_{ij} sadrži neke točke mreže. Čvorove mreže možemo sada poredati tako da grupiramo, redom, čvorove koji se nalaze u unutrašnjosti poddomena, najprije poddomene Ω_1 , zatim Ω_2 , sve do Ω_s . Na kraju poredamo čvorove granica. Kao rezultat matrica pridružena tom problemu imat će oblik

$$\begin{bmatrix} B_1 & & & E_1 \\ & B_2 & & E_2 \\ & & \ddots & \vdots \\ & & & B_s & E_s \\ F_1 & F_2 & \cdots & F_s & C \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_s \\ y \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_s \\ g \end{bmatrix},$$

gdje x_i predstavlja podvektor nepoznanica koje se odnose na točke u unutrašnjosti poddomene Ω_i , a y predstavlja vektor svih nepoznanica koje se odnose na točke koje pripadaju graničnom području. Primijetimo da su blokovi na pozicijama (i, j) , za $i, j = 1, \dots, s$,

$i \neq j$ jednaki nuli, zato što niti jedna točka iz unutrašnjosti jedne domene nije direktan susjed niti jedne točke iz unutrašnjosti bilo koje druge domene. Jedino može biti susjedna nekoj točki iz graničnog područja.

Dobiveni sustav možemo napisati u jednostavnijoj formi, koju će namo koristiti u daljnjoj analizi.

$$A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \quad \text{sa} \quad A = \begin{bmatrix} B & E \\ F & C \end{bmatrix}.$$

Pretpostavimo da je B regularna matrica, što je za očekivati, jer su B_i regularne, kao matrice lokalnih problema na Ω_i . Tada iz prve jednadžbe možemo izraziti x kao

$$x = B^{-1}(f - Ey). \quad (1)$$

Uvrštavajući to u drugu jednadžbu, dobivamo *reducirani sustav* na nepoznanicama granice

$$(C - FB^{-1}E)y = g - FB^{-1}f. \quad (2)$$

Matrica

$$S = C - FB^{-1}E \quad (3)$$

zove se matrica *Schurovog komplementa* sustava, po nepoznanici y . Ako bi se ta matrica mogla izračunati, i ako se može riješiti sustav (2), dobili bi vrijednosti za sve varijable graničnog područja. Ostale nepoznanice mogu se tada izračunati preko (1). Zbog posebne strukture matrice B , koja je zapravo blok-dijagonalna, rješavanje linearnog sustava s njom je ekvivalentno rješavanju s nezavisnih i manjih sustava. Ovdje do izražaja može doći paralelno računanje.

Metoda za rješavanje sustava bazirana na ovom pristupu sasatoji se od četiri koraka:

1. Izračunaj desnu stranu reduciranog sustava (2).
2. Izračunaj matricu Schurovog komplementa (3).
3. Riješi reducirani sustav (2).
4. Pomoću (1) izračunaj ostale nepoznanice.

Jedno rješavanje sustava sa matricom B može se uštedjeti preformuliranjem algoritma u pogodniji oblik. Definirajmo

$$E' = B^{-1}E, \quad \text{i} \quad f' = B^{-1}f.$$

Matrica E' i vektor f' su potrebni u prvom i drugom koraku algoritma. Četvrti korak, tada možemo napisati kao

$$x = B^{-1}f - B^{-1}Ey = f' - E'y,$$

što nam daje sljedeći algoritam.

Algoritam 7 (Algoritam blok-Gaussovih eliminacija)

Riješi $BE' = E$, i $Bf' = f$ po E' i f' ;

Izračunaj $g' = g - Ff'$;

Izračunaj $S = C - FE'$;

Riješi $Sy = g'$;

Izračunaj $x = f' - E'y$;

Testirajte ovu metodu za Poissonovu jednadžbu na različitim nepravilnim domenama poput slova L, T, H ili nekih drugih koje se mogu dobiti sastavljanjem pravokutnika. Isprobajte i razne mreže na tim domenama. Rješavanje sustava sa matricom B implementirajte kao rješavanje s sustava sa matricama B_i , i pritom upotrijebite neku pogodnu iterativnu metodu. Usporedite vrijeme trajanja i rezultat vaše metode dekompozicije domene sa istom iterativnom metodom ali koju ste ovaj puta primijenili na cijeli sustav sa matricom A .

Literatura: Nela Bosner, “Iterativne metode za rješavanje linearnih sustava”, magistarski rad (dostupno na mojoj web stranici)

Zadatak 28. Iterativne metode za svojstveni problem — QR algoritam

Ovo je standardna metoda za računanje svih svojstvenih vrijednosti kompleksne matrice preko Schurove forme, a zasniva se na dva osnovna koraka: svođenju na Hessenbergovu formu pomoću Householderovih reflektora i implicitnih QR iteracija sa pomakom za računanje Schurove forme Hessenbergove matrice. Za matricu $A \in \mathbb{C}^{n \times n}$, metoda izvršava sljedeće korake.

1. Householderova redukcija na Hessenbergovu formu bazira se na transformacijama sličnosti Householderovim reflektorima, tako da na kraju dobijemo

$$H = U_{n-2} \cdots U_1 A U_1^* \cdots U_{n-2}^*, \quad U = U_1^* \cdots U_{n-2}^*$$

gdje su $U_k \in \mathbb{C}^{n \times n}$ Householderovi reflektori, a $H \in \mathbb{C}^{n \times n}$ je Hessenbergova matrica oblika

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,n-1} & h_{1,n} \\ h_{21} & h_{22} & \cdots & h_{2,n-1} & h_{2,n} \\ & \ddots & \ddots & & \\ & & h_{n-1,n-2} & h_{n-1,n-1} & h_{n-1,n} \\ & & & h_{n,n-1} & h_{nn} \end{bmatrix}.$$

Postupak se odvija u $n - 2$ koraka. Za $k = 1, \dots, n - 2$ radite sljedeće

- (a) generirajte Householderov reflektor $U_k = \begin{bmatrix} I_k & 0 \\ 0 & \bar{U}_k \end{bmatrix}$ takav da poništava elemente $A(k + 2 : n, k)$: $\bar{U}_k A(k + 1 : n, k) = [* 0 \cdots 0]^T$,
 - (b) ažurirajte ostatak matrice A sa U_k s lijeva: $A(k + 1 : n, k + 1 : n) = \bar{U}_k A(k + 1 : n, k + 1 : n)$,
 - (c) ažurirajte ostatak matrice A sa U_k^* s desna: $A(1 : n, k + 1 : n) = A(1 : n, k + 1 : n) \bar{U}_k^*$.
2. QR iteracije sa pomakom primijenjene na Hessenbergovu matricu baziraju se na transformacijama sličnosti Givensovim rotacijama: definira se $H_0 = H$, a potom se za $k = 0, 1, 2, \dots$ i za neki izbor pomaka λ_k rade implicitne iteracije

$$H_k - \lambda_k I = QR \quad (\text{QR faktorizacija})$$

$$H_{k+1} = RQ + \lambda_k I,$$

čime se dobiva nova Hessenbergova matrica $H_{k+1} = Q^* H_k Q$. Pomak λ_k se bira tako da ubrza konvergenciju, a često se bira kao $\lambda_k = h_{nn}$. Koraci algoritma su sljedeći:

- (a) odaberite $\lambda_k = h_{nn} = H_k(n, n)$,
- (b) izračunajte $c_{\theta_1} \in \mathbb{R}$ and $s_{\theta_1} \in \mathbb{C}$ takve da je

$$\begin{bmatrix} c_{\theta_1} & s_{\theta_1} \\ -\bar{s}_{\theta_1} & c_{\theta_1} \end{bmatrix} \begin{bmatrix} h_{11} - \lambda_k \\ h_{21} \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix},$$

i definirajte Givensovu rotaciju $U_1 = G_{1,2}(\theta_1)$. U kompleksnom slučaju Givensova rotacija $G_{1,2}$ koja poništava drugu komponentu vektora $[x_1 \ x_2]^T$ definira se sa

$$c_{\theta} = \frac{|x_1|}{\sqrt{|x_1|^2 + |x_2|^2}}, \quad s_{\theta} = \text{sign}(x_1) \frac{|\bar{x}_2|}{\sqrt{|x_1|^2 + |x_2|^2}}.$$

- (c) izračunajte Givensove rotacije U_2, \dots, U_{n-1} takve da, ako je $U^{(k)} = U_1^* \dots U_{n-1}^*$ tada je $H_{k+1} = (U^{(k)})^* H_k U^{(k)}$ opet Hessenbergova, a $U^{(k)} e_1 = U_1^* e_1$. Taj postupak naziva se “naganjanje kvрге” u Hessenbergovoj matrici H_k

$$\begin{array}{cc}
 H_{k,1a} \leftarrow U_1 H_k = \begin{array}{|c|} \hline \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \times & \times & \times & \times & \times & \\ & \times & \times & \times & \times & \\ & & \times & \times & \times & \\ & & & \times & \times & \\ & & & & \times & \times \\ \hline \end{array} & H_{k,1} \leftarrow H_{k,1a} U_1^* = \begin{array}{|c|} \hline \bullet & \bullet & \times & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ & & \times & \times & \times & \times \\ & & & \times & \times & \times \\ & & & & \times & \times \\ & & & & & \times & \times \\ \hline \end{array} \\
 \\
 H_{k,2a} \leftarrow U_2 H_{k,1a} = \begin{array}{|c|} \hline \times & \times & \times & \times & \times & \times \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \times & \times & \times & \times \\ & & & \times & \times & \times \\ & & & & \times & \times \\ \hline \end{array} & H_{k,2} \leftarrow H_{k,2a} U_2^* = \begin{array}{|c|} \hline \times & \bullet & \bullet & \times & \times & \times \\ \times & \bullet & \bullet & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ & & & \times & \times & \times \\ & & & & \times & \times \\ \hline \end{array} \\
 \\
 H_{k,3a} \leftarrow U_3 H_{k,2a} = \begin{array}{|c|} \hline \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \times & \times & \times & \times \\ & & & \times & \times & \times \\ \hline \end{array} & H_{k,3} \leftarrow H_{k,3a} U_3^* = \begin{array}{|c|} \hline \times & \times & \bullet & \bullet & \times & \times \\ \times & \times & \bullet & \bullet & \times & \times \\ \times & \bullet & \bullet & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ & & & \times & \times & \times \\ \hline \end{array} \\
 \\
 H_{k,4a} \leftarrow U_4 H_{k,3a} = \begin{array}{|c|} \hline \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \times & \times & \times & \times \\ \hline \end{array} & H_{k,4} \leftarrow H_{k,4a} U_4^* = \begin{array}{|c|} \hline \times & \times & \times & \bullet & \bullet & \times \\ \times & \times & \times & \bullet & \bullet & \times \\ \times & \times & \bullet & \bullet & \times & \times \\ \times & \bullet & \bullet & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ \bullet & \bullet & \times & \times & \times & \times \\ \hline \end{array} \\
 \\
 H_{k,5a} \leftarrow U_5 H_{k,4a} = \begin{array}{|c|} \hline \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \bullet & \bullet \\ \hline \end{array} & H_{k,5} \leftarrow H_{k,5a} U_5^* = \begin{array}{|c|} \hline \times & \times & \times & \times & \bullet & \bullet \\ \times & \times & \times & \times & \bullet & \bullet \\ \times & \times & \times & \times & \bullet & \bullet \\ \times & \times & \times & \times & \bullet & \bullet \\ \times & \times & \times & \times & \bullet & \bullet \\ \times & \times & \times & \times & \bullet & \bullet \\ \hline \end{array}
 \end{array}$$

Iteracije se završavaju kada svi vandijagonalni elementi $h_{i+1,i}$, $i = 1, \dots, n-1$ postanu dovoljno mali, što se mjeri sa

$$|h_{i+1,i}| \leq \varepsilon(|h_{ii}| + |h_{i+1,i+1}|),$$

za zadanu toleranciju ε .

3. Konačna Schurova dekompozicija matrice A se postiže tako da, ako definiramo

$$\begin{aligned}
 A &= U_A H U_A^* && \text{(redukcija na Hessenbergovu formu)} \\
 H &= U_H T U_H^* && \text{(Schurova dekompozicija Hessenbergove matrice)}
 \end{aligned}$$

vrijedi

$$U = U_A U_H, \quad A = U T U^*.$$

Zbog povoljnih svojstava strogo Hessenbergovih matrica, koje imaju sve netrivialne ispoddijagonalne elemente, ova metoda se provodi na takvim matricama. Ako je neki $h_{j+1,j} = 0$ (ili na računalu dovoljno mali u gornjem smislu) onda se problem razbija na probleme manjih dimenzije. Zato se konačna QR metoda provodi na sljedeći način.

- Izvedite redukciju matrice A na Hessenbergovu formu $H = U_A^* A U_A$.
- Postavite dimenziju trokutastog dijela matrice H na $q = 0$.
- Dok nije $q = n$, radite sljedeće
 - Postavite sve ispoddijagonalne elemente na nulu koji zadovoljavaju uvjet $|h_{i+1,i}| \leq \varepsilon(|h_{ii}| + |h_{i+1,i+1}|)$.
 - Nađite najveći nenegativan q i najmanji nenegativan p takve da je

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ 0 & H_{22} & H_{23} \\ 0 & 0 & H_{33} \\ p & n-p-q & q \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$

gdje je H_{33} gornje trokutasta, a H_{22} je strogo Hessenbergova (p ili q mogu biti 0).

- Ako je $q < n$ tada izvršite QR iteracije sa pomakom na $H_{22} = U_{22}^* H_{22} U_{22}$. Ažurirajte ostatak matrice

$$H_{12} = H_{12} U_{22}, \quad H_{23} = U_{22}^* H_{23}.$$

Napišite potprogram koji implementira gore opisanu metodu, i kao ulazni parametar uzima matricu A i toleranciju ε . Primijenite svoj potprogram na raznim matricama, koje imaju unaprijed poznate svojstvene vrijednosti. Kontrolirajte relativnu grešku svojstvenih vrijednosti, a naročito obratite pozornost na matrice sa nakupinama bliskih svojstvenih vrijednosti, i na matrice sa velikom razlikom u redu veličine modula svojstvenih vrijednosti. Usporedite svoj rezultat za $\varepsilon = n \cdot u$, gdje je u mašinski epsilon, sa rezultatom LAPACK-ovog potprograma `zgeev_()`. Usporedite brzine izvršavanja ta dva potprograma.

Literatura: 9. predavanje "Numerička analiza" na doktorskom studiju 2011./12. (dostupno na http://web.math.pmf.unizg.hr/~nela/nad_11_12.html), te podsekcije 7.4.3, 7.5.1, 7.5.2, 7.5.3 i 7.5.6 iz knjige: G. H. Golub, C. F. Van Loan, "Matrix computations", The Johns Hopkins University Press, Baltimore and London, 1989.

Zadatak 29. Iterativne metode za svojstveni problem — Simetričan QR algoritam

Ovo je standardna metoda za računanje svih svojstvenih vrijednosti simetrične realne matrice, koja je u ovom slučaju dijagonalna. Metoda se zasniva na dva osnovna koraka: svođenju na tridijagonalnu formu pomoću Householderovih reflektora i implicitnih QR iteracija sa pomakom za svođenje tridijagonalne matrice na dijagonalnu formu. Za matricu $A \in \mathbb{R}^{n \times n}$, $A^T = A$, metoda izvršava sljedeće korake.

1. Householderova redukcija na tridijagonalnu formu bazira se na transformacijama sličnosti Householderovim reflektorima, tako da na kraju dobijemo

$$T = U_{n-2} \cdots U_1 A U_1 \cdots U_{n-2}, \quad U = U_1 \cdots U_{n-2}$$

gdje su $U_k \in \mathbb{C}^{n \times n}$ Householderovi reflektori, a $T \in \mathbb{R}^{n \times n}$ je simetrična tridijagonalna matrica oblika

$$T = \begin{bmatrix} a_1 & b_1 & & \cdots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & & b_{n-1} & a_n \end{bmatrix}.$$

Postupak se odvija u $n - 2$ koraka. Za $k = 1, \dots, n - 2$ radite sljedeće

- (a) generirajte Householderov reflektor $U_k = \begin{bmatrix} I_k & 0 \\ 0 & \bar{U}_k \end{bmatrix}$ takav da poništava elemente $A(k + 2 : n, k)$: $\bar{U}_k A(k + 1 : n, k) = \begin{bmatrix} * & 0 & \cdots & 0 \end{bmatrix}^T$, i analogno ažurirajte k -ti redak,
- (b) ažurirajte ostatak matrice A sa U_k : $A(k + 1 : n, k + 1 : n) = \bar{U}_k A(k + 1 : n, k + 1 : n) \bar{U}_k$.

Ovdje je važno da sva ažuriranja budu simetrična, i dovoljno je da se ažurira samo jedan trokut matrice. Pretpostavimo da reflektor \bar{U}_k ima oblik $\bar{U}_k = I - \gamma_k v_k v_k^T$, $v \in \mathbb{R}^{n-k}$, tada ako definiramo

$$p_k = \gamma_k A(k + 1 : n, k + 1 : n) v_k \quad \text{i} \quad w_k = p_k - \frac{\gamma_k p_k^T v_k}{2} v_k,$$

ažuriranje možemo zapisati kao

$$\bar{U}_k A(k + 1 : n, k + 1 : n) \bar{U}_k = A(k + 1 : n, k + 1 : n) - v_k w_k^T - w_k v_k^T.$$

2. QR iteracije sa pomakom primijenjene na tridijagonalnu simetričnu matricu baziraju se na transformacijama sličnosti Givensovim rotacijama: definira se $T_0 = T$, a potom se za $k = 0, 1, 2, \dots$ i za neki izbor pomaka λ_k rade implicitne iteracije

$$\begin{aligned} T_k - \lambda_k I &= QR \quad (\text{QR faktorizacija}) \\ T_{k+1} &= RQ + \lambda_k I, \end{aligned}$$

čime se dobiva nova tridijagonalna matrica $T_{k+1} = Q^T T_k Q$. Pomak λ_k se bira tako da ubrza konvergenciju. Čest odabir je Wilkinsonov pomak koji se bira kao svojstvena

vrijednost matrice $T_k(n-1:n, n-1:n) = \begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$ bliža elementu $T_k(n, n) = a_n$.
 On se može izračunati kao

$$\lambda_k = a_n + d - \text{sign}(d) \sqrt{d^2 + b_{n-1}^2}, \quad d = \frac{a_{n-1} - a_n}{2}.$$

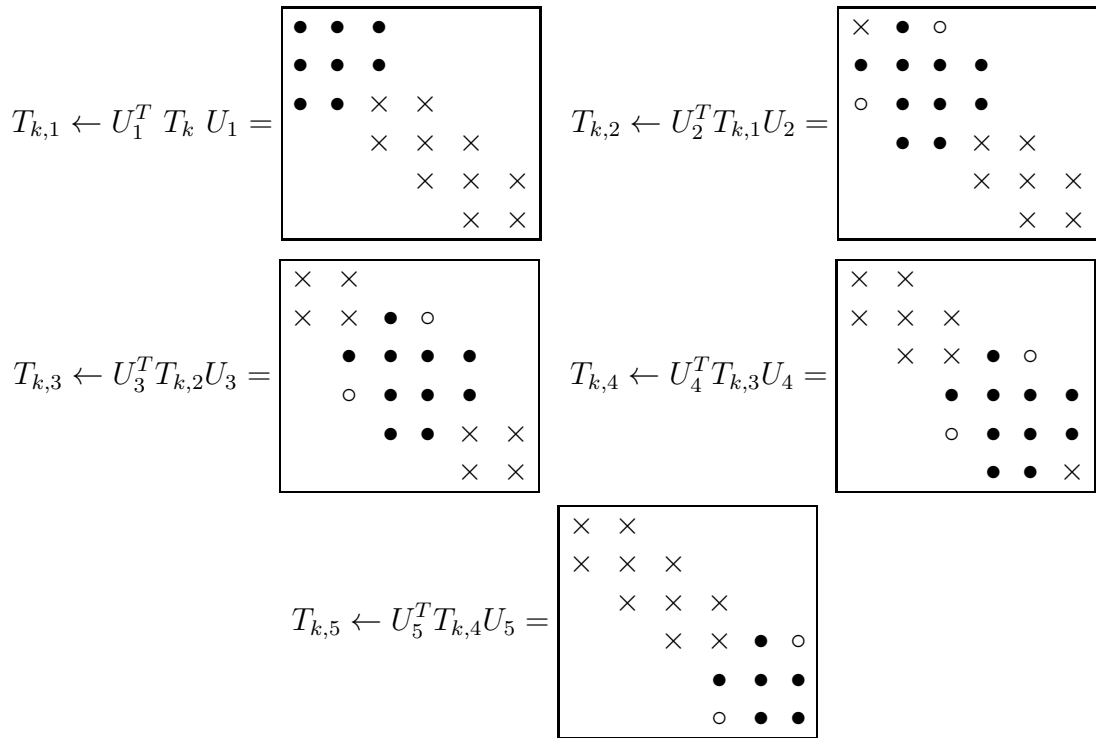
Koraci algoritma su sljedeći:

- (a) odaberite Wilkinsonov pomak λ_k ,
- (b) izračunajte $c_{\theta_1} = \cos(\theta_1)$ and $s_{\theta_1} \sin(\theta_1)$ takve da je

$$\begin{bmatrix} c_{\theta_1} & s_{\theta_1} \\ -s_{\theta_1} & c_{\theta_1} \end{bmatrix}^T \begin{bmatrix} a_1 - \lambda_k \\ b_1 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix},$$

i definirajte Givensovu rotaciju $U_1 = G_{1,2}(\theta_1)$.

- (c) izračunajte Givensove rotacije U_2, \dots, U_{n-1} takve da, ako je $U^{(k)} = U_1 \dots U_{n-1}$ tada je $T_{k+1} = (U^{(k)})^T T_k U^{(k)}$ opet tridijagonalna, a $U^{(k)} e_1 = U_1 e_1$. Taj postupak naziva se “naganjanje kvрге” u tridijagonalnoj matrici T_k



Iteracije se završavaju kada svi vandijagonalni elementi $t_{i+1,i} = t_{i,i+1}$, $i = 1, \dots, n-1$ postanu dovoljno mali, što se mjeri sa

$$|t_{i+1,i}| = |t_{i,i+1}| \leq \varepsilon(|t_{ii}| + |t_{i+1,i+1}|),$$

za zadanu toleranciju ε .

- 3. Konačna Schurova dekompozicija (dijagonalizacija) matrice A se postiže tako da, ako definiramo

$$\begin{aligned} A &= U_A T U_A^* && \text{(redukcija na tridijalnu formu)} \\ T &= U_T D U_T^* && \text{(Schurova dekompozicija tridijagonalne matrice)} \end{aligned}$$

vrijedi

$$U = U_A U_T, \quad A = U D U^*.$$

Zbog povoljnih svojstava strogo tridijagonalnih matrica, koje imaju sve netrivialne vandijagonalne elemente, ova metoda se provodi na takvim matricama. Ako je neki $t_{j+1,j} = t_{j,j+1} = 0$ (ili na računalu dovoljno mali u gornjem smislu) onda se problem razbija na probleme manjih dimenzije. Zato se konačna simetrična QR metoda provodi na sljedeći način.

- Izvedite redukciju matrice A na tridijagonalnu formu $T = U_A^T A U_A$.
- Postavite dimenziju dijagonalnog dijela matrice T na $q = 0$.
- Dok nije $q = n$, radite sljedeće
 - Postavite sve vandijagonalne elemente na nulu koji zadovoljavaju uvjet $|t_{i+1,i}| = |t_{i,i+1}| \leq \varepsilon(|t_{ii}| + |t_{i+1,i+1}|)$.
 - Nađite najveći nenegativan q i najmanji nenegativan p takve da je

$$T = \begin{bmatrix} T_{11} & 0 & 0 \\ 0 & T_{22} & 0 \\ 0 & 0 & T_{33} \\ p & n-p-q & q \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$

gdje je T_{33} dijagonalna, a T_{22} je strogo tridijagonalna (p ili q mogu biti 0).

- Ako je $q < n$ tada izvršite simetrične QR iteracije sa pomakom na $T_{22} = U_{22}^T T_{22} U_{22}$.

Napišite potprogram koji implementira gore opisanu metodu, i kao ulazni parametar uzima matricu simetričnu A i toleranciju ε . Primijenite svoj potprogram na raznim matricama, koje imaju unaprijed poznate svojstvene vrijednosti. Kontrolirajte relativnu grešku svojstvenih vrijednosti, a naročito obratite pozornost na matrice sa nakupinama bliskih svojstvenih vrijednosti, i na matrice sa velikom razlikom u redu veličine svojstvenih vrijednosti. Usporedite svoj rezultat za $\varepsilon = n \cdot u$, gdje je u mašinski epsilon, sa rezultatom LAPACK-ovog potprograma `dsyev_()`. Usporedite brzine izvršavanja ta dva potprograma.

Literatura: 9. predavanje "Numerička analiza" na doktorskom studiju 2011./12. (dostupno na http://web.math.pmf.unizg.hr/~nela/nad_11_12.html), te sekciju 8.2 iz knjige: G. H. Golub, C. F. Van Loan, "Matrix computations", The Johns Hopkins University Press, Baltimore and London, 1989.

Zadatak 30. Iterativne metode za svojstveni problem — Arnoldijev algoritam s restartom

Pretpostavimo da je $A \in \mathbb{R}^{n \times n}$ (ili $\mathbb{C}^{n \times n}$) velika, rijetko popunjena, nesimetrična matrica, i pretpostavimo da nas zanima samo nekoliko određenih svojstvenih vrijednosti. Arnoldijeva metoda generira stupac po stupac ortogonalne matrice Q takve da je $Q^T A Q = H$ svodi na Hessenbergovu matricu. Stupci matrice $Q = [q_1 \ q_2 \ \cdots \ q_n]$ ujedno predstavljaju i ortonormiranu bazu za Krylovljeve prostore $\mathcal{K}_k(A, q_1)$. Parcijalno generirana $k \times k$ Hessenbergova matrica $H_k = [h_{ij}] = H(1 : k, 1 : k)$ malih dimenzija može nam poslužiti za dobivanje aproksimacija svojstvenih vrijednosti matrice A .

Algoritam 8 (Arnoldijev algoritam)

```

 $r_0 = q_1;$ 
 $h_{10} = 1;$ 
 $k = 0;$ 
while ( $h_{k+1,k} \neq 0$ ) {
     $q_{k+1} = r_k / h_{k+1,k};$ 
     $k = k + 1;$ 
     $r_k = A q_k;$ 
    for  $i = 1 : k$  {
         $h_{ik} = q_i^T r_k;$ 
         $r_k = r_k - h_{ik} q_i;$ 
    }
     $h_{k+1,k} = \|r_k\|_2;$ 
}

```

Nakon k -tog koraka algoritma, rezultat algoritma možemo zapisati u matričnom obliku kao Arnoldijevu faktorizaciju

$$A Q_k = Q_k H_k + r_k e_k^T, \quad Q_k^T r_k = 0,$$

gdje je $Q_k = [q_1 \ \cdots \ q_k]$ i $e_k = [0 \ \cdots \ 0 \ 1]^T \in \mathbb{R}^k$. Odavde se vidi da, ako je $r_k = 0$, tada stupci od Q_k razapinju invarijantni potprostor od A i $\sigma(H_k) \subseteq \sigma(A)$. Ako to nije slučaj, cilj je izvući informacije o spektru matrice A iz Hessenberove matrice H_k i ortonormirane matrice Q_k .

Ako je $y \in \mathbb{R}^k$ jedinični svojstveni vektor od H_k i $H_k y = \lambda y$, tada iz matričnog oblika Arnoldijeve iteracije slijedi

$$(A - \lambda I)x = (e_k^T y) r_k,$$

pri čemu je $x = Q_k y$ Ritzov vektor (aproksimacija svojstvenog vektora) za Ritzovu vrijednost λ (aproksimacija svojstvene vrijednosti). Veličina $|e_k^T y| \|r_k\|_2$ može se uzeti kao ograda na grešku aproksimacije svojstvenog vektora.

Kod ove metode postoje neki numerički problemi koji se javljaju kad se ona izvodi na računalu. To su: gubitak ortogonalnosti stupaca matrice Q_k i porast zahtjeva za memorijom i brojem operacija sa porastom broja iteracija.

Zato se Arnoldijeva metoda najčešće implementira sa restartanjem nakon fiksnog broja iteracija. Pretpostavimo da smo izvršili m iteracija metode, i da želimo restartati metodu sa vektorom q_+ iz prostora Krylovljevog prostora razapetog stupcima matrice Q_m . Tada je q_+ oblika

$$q_+ = p(A)q_1, \quad \text{za neki polinom stupnja } m - 1.$$

Polinom p bira se tako da priguši komponente u smjeru svojstvenih vektora koji nas ne interesiraju. Restart će se izvršiti nakon svakih m iteracija, za $m > j$, gdje je j broj traženih svojstvenih vrijednosti. Broj iteracija m bira se tako da bude dovoljno mali kako ne bi došlo do gubitka otogonalnosti i prevelikih zahtjeva za memorijom. Vektor q_+ kojim se restarta određuje se implicitno koristeći QR iteracije s pomakom. Krećemo od Arnoldijeve faktorizacije nakon m iteracija

$$AQ_m = Q_m H_m + r_m e_m^T,$$

i izračunamo spektar male matrice H_m nekom direktnom metodom (QR metoda), i partitioniramo ga na

$$\sigma(H_m) = \{\tilde{\lambda}_1, \dots, \tilde{\lambda}_j\} \cup \{\tilde{\lambda}_{j+1}, \dots, \tilde{\lambda}_m\}.$$

Ako $\tilde{\lambda}_1, \dots, \tilde{\lambda}_j$ aproksimiraju dio spektra matrice A koji nas zanima, tada ćemo za pomake uzeti $\mu_i = \tilde{\lambda}_{j+i}$ za $i = 1, \dots, p$, $p = m - j$. U tom slučaju može se pokazati da je q_+ implicitno određen sa polinomom

$$p(\lambda) = (\lambda - \mu_1)(\lambda - \mu_2) \cdots (\lambda - \mu_p).$$

Znači, nakon m iteracija Arnoldijeve metode izvrše se p koraka QR iteracija sa pomakom na Hessenbergovoj matrici H_m (vidi korak 2. u zadatku 28)

Algoritam 9 (QR iteracije s pomakom)

$$H^{(1)} = H_m;$$

for $i = 1 : p$ {

$$H^{(i)} - \mu_i I = V_i R_i;$$

$$H^{(i+1)} = R_i V_i + \mu_i I;$$

}

$$H_+ = H^{(p+1)}$$

Definiramo $V = V_1 \cdots V_p$, i tada se može pokazati da vrijedi da je $H_+ = V^T H_m V$, a obzirom da su sve V_i Hessenbergove matrice V je p -Hessenbergova matrica ($v_{ij} = 0$ za $i > j + p$). Zbog toga Arnoldijevu faktorizaciju možemo transformirati u oblik

$$AQ_+ = Q_+ H_+ + r_m e_m^T V,$$

gdje je $Q_+ = Q_m V$. Ako pogledamo samo prvih j stupaca ove faktorizacije, koristeći strukturu matrice V , dobivamo

$$AQ_+(\cdot, 1:j) = Q_+(\cdot, 1:j)H_+(1:j, 1:j) + v_{mj}r_m e_j^T.$$

Dobiveni izraz upravo predstavlja Arnoldijevu faktorizaciju nakon j iteracija sa početnim vektorom $q_+ = Q_+(\cdot, 1)$, za kojeg se može pokazati da je upravo oblika $q_+ = p(A)q_1$, i koji ima bolje svojstva obzirom na željeni dio spektra. Dalje se Arnoldijeve iteracije nastavljaju do m -te, ali sa matricama $Q_+(\cdot, 1:j)$ i $H_+(1:j, 1:j)$, a nakon toga se eventualno ponovo ponavlja postupak ukoliko nije došlo do konvergencije. Za Ritzov par (λ, y) smatramo da je izkonvergirao ako zadovoljava kriterij zaustavljanja

$$\|r_m\|_2 |e_m^T y| \leq \max\{\epsilon_M \|H_m\|_2, \text{tol} \cdot |\lambda|\},$$

pri čemu je ϵ_M je mašinska preciznost, a tol je ograda na povratnu grešku, tj. garantira da je Ritzov par egzaktni svojstveni par za perturbiranu matricu

$$(A + E)y = \lambda y, \quad \|E\|_2 \leq \text{tol}\|A\|_2.$$

Napišite potprogram koji implementira gore opisanu Arnoldijevu metodu sa implicitnim restartom, i kao ulazni parametar uzima matricu A i toleranciju tol . Primijenite svoj potprogram na raznim matricama, koje imaju unaprijed poznate svojstvene vrijednosti, ali tako da tražite nekoliko najvećih ili najmanjih po modulu svojstvenih vrijednosti. Kontrolirajte relativnu grešku svojstvenih vrijednosti. Usporedite izvršavanje ove metode sa samom Arnoldijevom metodom bez restarta, i to tako da usporedite točnost izračunatih aproksimacija svojstvenih vrijednosti, broj iteracija potrebnih za iste ulazne parametre, broj realnih ili kompleksnih brojeva koje morate spremiti za svaki od algoritama (potrošnja memorije) i vrijeme izvršavanja.

Literatura: Podsekcije 9.4.1 i 9.4.2 iz knjige G. H. Golub, C. F. Van Loan, "Matrix computations", Third Edition, The Johns Hopkins University Press, Baltimore and London, 1996., i predavanja iz predmeta Numerička analiza 1.

Zadatak 31. Iterativne metode za svojstveni problem — Lanczosov algoritam s reortogonalizacijom

Pretpostavimo da je $A \in \mathbb{R}^{n \times n}$ velika, rijetko popunjena, simetrična matrica, i pretpostavimo da nas zanima samo nekoliko određenih svojstvenih vrijednosti. Lanczosova metoda generira stupac po stupac ortogonalne matrice Q takve da je $Q^T A Q = T$ svodi na tridijagonalnu matricu, gdje je

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

Stupci matrice $Q = [q_1 \ q_2 \ \cdots \ q_n]$ ujedno predstavljaju i ortonormiranu bazu za Krylovljeve prostore $\mathcal{K}_k(A, q_1)$. Parcijalno generirana $k \times k$ tridijagonalna matrica $T_k = T(1:k, 1:k)$ malih dimenzija može nam poslužiti za dobivanje aproksimacija svojstvenih vrijednosti matrice A .

Algoritam 10 (Lanczosov algoritam)

```

 $r_0 = q_1;$ 
 $\beta_0 = 1; q_0 = 0;$ 
 $k = 0;$ 
while ( $\beta_k \neq 0$ ) {
     $q_{k+1} = r_k / \beta_k;$ 
     $k = k + 1;$ 
     $\alpha_k = q_k^T A q_k;$ 
     $r_k = A q_k - \alpha_k q_k - \beta_{k-1} q_{k-1};$ 
     $\beta_k = \|r_k\|_2;$ 
}

```

Nakon k -tog koraka algoritma, rezultat algoritma možemo zapisati u matričnom obliku kao Lanczosovu faktorizaciju

$$AQ = Q_k T_k + r_k e_k^T, \quad Q_k^T r_k = 0,$$

gdje je $Q_k = [q_1 \ \cdots \ q_k]$ i $e_k = [0 \ \cdots \ 0 \ 1]^T \in \mathbb{R}^k$. Odavde se vidi da, ako je $r_k = 0$, tada stupci od Q_k razapinju invarijantni potprostor od A i $\sigma(T_k) \subseteq \sigma(A)$. Ako to nije slučaj, cilj je izvući informacije o spektru matrice A iz tridijagonalne matrice T_k i ortonormirane matrice Q_k . Analiza ove metode pokazuje da će ekstremne svojstvene vrijednosti (namannje i najveće) izkonvergirati, i biti dobro aproksimirane, i prije nego što se tridijagonalizacija izvrši do kraja.

Ako je $y \in \mathbb{R}^k$ jedinični svojstveni vektor od T_k i $T_k y = \lambda y$, tada iz matričnog oblika Lanczosove iteracije slijedi

$$(A - \lambda I)x = (e_k^T y)r_k,$$

pri čemu je $x = Q_k y$ Ritzov vektor (aproksimacija svojstvenog vektora) za Ritzovu vrijednost λ (aproksimacija svojstvene vrijednosti). Veličina $\beta_k |e_k^T y|$ može se uzeti kao ograda na grešku aproksimacije svojstvenog vektora i kao kriterij zaustavljanja iteracija.

Kod ove metode postoje neki numerički problemi koji se javljaju kad se ona izvodi na računalu. To su je prije svega gubitak ortogonalnosti stupaca matrice Q_k , koji može dovesti do loših aproksimacija svojstvenih vrijednosti.

Zbog toga se vrši reortogonalizacija stupaca matrice Q_k , i to se može napraviti na nekoliko načina.

- Potpuna reortogonalizacija izračunati vektor r_k iz Lanczosovog algoritma još jednom ortogonalizira u odnosu na vektore q_1, \dots, q_k :

$$r_k = Aq_k - \alpha_k q_k - \beta_{k-1} q_{k-1}, \quad r_k = r_k - Q_k(Q_k^T r_k);$$

- Parcijalna reortogonalizacija nastoji reortogonalizirati vektore q_i samo onda kada je potrebno i kada se gubitak ortogonalnosti zaista dogodi. Temelji se na rezultatu koji pokazuje da ako su vektori q_1, \dots, q_k semiortogonalni, što znači da je

$$W_k = Q_k^T Q_k = I_k + E, \quad \|E\|_F < \sqrt{\epsilon_M},$$

gdje je ϵ_M mašinska preciznost, tada je tridijagonalna matrica T_k projekcija matrice A na potprostor $\mathcal{R}(Q_k)$ razapet sa q_1, \dots, q_k , i vrijedi

$$T_k = P_k^T A P_k + F, \quad \|F\|_F = \mathcal{O}(\epsilon_M) \|A\|_F,$$

pri čemu stupci od P_k čine ortonormiranu bazu za $\mathcal{R}(Q_k)$. Zbog toga je za točne aproksimacije svojstvenih vrijednosti bitno da Q_k bude semiortogonalna matrica, a to se postiže praćenjem gubitka ortogonalnosti vektora q_i u toku Lanczosovih iteracija.

Izračunati Lanczosovi vektori zadovoljavaju

$$\beta_j q_{j+1} = Aq_j - \alpha_j q_j - \beta_{j-1} q_{j-1} + g_j,$$

gdje g_j predstavlja greške zaokruživanja počinjene u j -toj iteraciji. Neka je $W = [w_{ik}]$. Množenjem prethodne jednakosti sa q_k^T dobivamo

$$\beta_j w_{j+1,k} = q_k^T Aq_j - \alpha_j w_{jk} - \beta_{j-1} w_{j-1,k} + q_k^T g_j.$$

Zamjena indeksa j i k u prethodnoj jednakosti daje

$$\beta_k w_{j,k+1} = q_j^T Aq_k - \alpha_k w_{jk} - \beta_{k-1} w_{j,k-1} + q_j^T g_k.$$

Oduzimanjem zadnjih dviju jednakosti dobivamo

$$\beta_j w_{j+1,k} = \beta_k w_{j,k+1} + (\alpha_k - \alpha_j) w_{jk} - \beta_{k-1} w_{j,k-1} - \beta_{j-1} w_{j-1,k} - q_j^T g_k + q_k^T g_j.$$

Ako nam je poznata matrica W_j , prethodnu jednakost možemo iskoristiti da izračunamo $j+1$ -vi redak od W_{j+1} . Međutim, elementi $w_{j+1,j}$ i $w_{j+1,j+1}$ ne mogu se definirati na taj način. Njima možemo dodijeliti vrijednosti pomoću sljedećih zaključaka

- Postavit ćemo $w_{j+1,j+1} = 1$, jer se q_{j+1} eksplicitno normalizira.
- Postavit ćemo $w_{j+1,j} = \mathcal{O}(\epsilon_M)$ zato što se q_{j+1} eksplicitno ortogonalizira u odnosu na q_j u Lanczosovoj iteraciji.

Zato konačno računanje elementa $w_{j+1,k}$ računamo na sljedeći način

$$\begin{aligned}\tilde{w} &= \beta_k w_{j,k+1} + (\alpha_k - \alpha_j) w_{jk} - \beta_{k-1} w_{j,k-1} - \beta_{j-1} w_{j-1,k}, \\ w_{j+1,k} &= (\tilde{w} + \text{sign}(\tilde{w}) \cdot 2\epsilon_M \|A\|_F) / \beta_j,\end{aligned}$$

gdje $2\epsilon_M \|A\|_F$ predstavlja procjenu za $q_j^T g_k + q_k^T g_j$. Čim je $w_{j+1,k} > \sqrt{\epsilon_M}$, vektori q_j i q_{j+1} moraju se ortogonalizirati u odnosu na sve prijašnje Lanczosove vektore q_1, \dots, q_{j-1} . Tada se elementi zadnja dva retka matrice W_j postavljaju na brojeve reda veličine $\mathcal{O}(\epsilon_M)$. Primijetite da morate čuvati samo zadnja dva retka matrice W_j .

Napišite potprograme koji implementiraju gore opisane Lanczosove metode sa dvije vrste reortogonalizacija, i kao ulazni parametar uzimaju simetričnu matricu A . Svojtvene vrijednosti matrice T_k računajte simetričnim QR iteracijama za tridijagonalne matrice, opisane u Zadatku 29. Primijenite svoje potprograme na raznim matricama, koje imaju unaprijed poznate svojstvene vrijednosti, ali tako da tražite nekoliko najvećih ili najmanjih svojstvenih vrijednosti. Kontrolirajte relativnu grešku svojstvenih vrijednosti. Usporedite izvršavanje osnovne Lanczosove metode i dvije verzije metode sa reortogonalizacijom, i to tako da usporedite točnost izračunatih aproksimacija svojstvenih vrijednosti, ortogonalnost Lanczosovih vektora, broj iteracija potrebnih za iste ulazne parametre, broj brojeva koje morate spremiti za svaki od algoritama (potrošnja memorije) i vrijeme izvršavanja.

Literatura: Sekcija 9.2 iz knjige G. H. Golub, C. F. Van Loan, "Matrix computations", Third Edition, The Johns Hopkins University Press, Baltimore and London, 1996., i predavanja iz predmeta Numerička analiza 1.

Zadatak 32. Iterativne metode za svojstveni problem — Lanczosov algoritam s restartom

Pretpostavimo da je $A \in \mathbb{R}^{n \times n}$ velika, rijetko popunjena, simetrična matrica, i pretpostavimo da nas zanima samo nekoliko određenih svojstvenih vrijednosti. Lanczosova metoda generira stupac po stupac ortogonalne matrice Q takve da je $Q^T A Q = T$ svodi na tridijagonalnu matricu, gdje je

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & & \beta_{b-1} & \alpha_n \end{bmatrix}.$$

Stupci matrice $Q = [q_1 \ q_2 \ \cdots \ q_n]$ ujedno predstavljaju i ortonormiranu bazu za Krylovljeve prostore $\mathcal{K}_k(A, q_1)$. Parcijalno generirana $k \times k$ tridijagonalna matrica $T_k = T(1:k, 1:k)$ malih dimenzija može nam poslužiti za dobivanje aproksimacija svojstvenih vrijednosti matrice A .

Algoritam 11 (Lanczosov algoritam)

$r_0 = q_1;$

$\beta_0 = 1; q_0 = 0;$

$k = 0;$

while ($\beta_k \neq 0$) {

$q_{k+1} = r_k / \beta_k;$

$k = k + 1;$

$\alpha_k = q_k^T A q_k;$

$r_k = A q_k - \alpha_k q_k - \beta_{k-1} q_{k-1};$

$\beta_k = \|r_k\|_2;$

}

Nakon k -tog koraka algoritma, rezultat algoritma možemo zapisati u matričnom obliku kao Lanczosovu faktorizaciju

$$A Q = Q_k T_k + r_k e_k^T, \quad Q_k^T r_k = 0,$$

gdje je $Q_k = [q_1 \ \cdots \ q_k]$ i $e_k = [0 \ \cdots \ 0 \ 1]^T \in \mathbb{R}^k$. Odavde se vidi da, ako je $r_k = 0$, tada stupci od Q_k razapinju invarijantni potprostor od A i $\sigma(T_k) \subseteq \sigma(A)$. Ako to nije slučaj, cilj je izvući informacije o spektru matrice A iz tridijagonalne matrice T_k i ortonormirane matrice Q_k . Analiza ove metode pokazuje da će ekstremne svojstvene vrijednosti (namannje i najveće) izkonvergirati, i biti dobro aproksimirane, i prije nego što se tridijagonalizacija izvrši do kraja.

Kod ove metode postoje neki numerički problemi koji se javljaju kad se ona izvodi na računalu. To su je prije svega gubitak ortogonalnosti stupaca matrice Q_k , koji može dovesti do loših aproksimacija svojstvenih vrijednosti kad broj iteracija raste. A s porastom iteracija raste i dimenzija od T_k čiju spektralnu dekompoziciju računamo. Zato je praktično ograničiti broj iteracija na maksimalan broj m , a nakon toga se radi restart sa novim i povoljnijim početnim vektorom q_1 .

Neka je

$$T_k Y_k = Y_k \Lambda_k, \quad Y_k = [y_1 \ \cdots \ y_k], \quad \Lambda_k = \text{diag}(\lambda_k, \dots, \lambda_k)$$

svojstvene vrijednosti, ali tako da tražite nekoliko najvećih ili najmanjih svojstvenih vrijednosti. Kontrolirajte relativnu grešku svojstvenih vrijednosti. Usporedite izvršavanje osnovne Lanczosove metode i verzije metode sa restartom, i to tako da usporedite točnost izračunatih aproksimacija svojstvenih vrijednosti, ortogonalnost Lanczosovih vektora, broj iteracija potrebnih za iste ulazne parametre, broj brojeva koje morate spremi za svaki od algoritama (potrošnja memorije) i vrijeme izvršavanja.

Literatura: Sekcija 9.2 i podsekcija 9.4.2 iz knjige G. H. Golub, C. F. Van Loan, "Matrix computations", Third Edition, The Johns Hopkins University Press, Baltimore and London, 1996., i predavanja iz predmeta Numerička analiza 1.