Iterativne numerične metode v linearni algebri 2022/2023

1. domača naloga

Rešitve oddajte do **ponedeljka, 15. maja 2023**, do 23. ure. Navodila:

- Programe in poročilo stisnite v ZIP datoteko z imenom `ime-priimek-vpisna-1.zip`, ki jo oddate preko spletne učilnice (`http://ucilnica.fmf.uni-lj.si`).
- V poročilo ni potrebno stisniti testnih datotek z matrikami, ki so na voljo na spletni učilnici. Če pa imate kakšne druge testne podatke, jih priložite.
- V poročilu za vsako nalogo opišite postopek reševanja, zapišite rešitev in komentirajte rezultat. Če poročilo skenirate, mora biti oddano v pdf obliki.
- Rešitvi priložite izjavo, da ste naloge reševali samostojno.
- Naloge rešujte v Matlabu (ali v drugem jeziku po predhodnem dogovoru z asistentom). Programi naj bodo smiselno poimenovani in razporejeni v mapah, ki so poimenovane nal1, nal2, ... K vsaki nalogi spada glavna skripta, ki izpiše rešitve naloge (npr. nal1.m, ..., nal4.m za Matlab datoteke). Preverite, da se skripte res izvedejo v ukazni vrstici (npr. klic nal1 se mora izvesti brez napak), v nasprotnem boste izgubili polovico točk pri konkretni nalogi.
- Z vprašanji o nalogah ali Matlabu se lahko obrnete na asistenta. Če menite, da je vprašanje zanimivo tudi za ostale, uporabite forum. Vprašanja so dobrodošla.

Vsaka naloga je vredna 5 točk, ena naloga je za bonus, skupno lahko torej zberete do 20 točk (15 točk = 100%).

Naloga 1

Eden izmed možnih pristopov, kako rešiti robni problem na bolj zapleteni domeni Ω, je preko t.i. *kaznovalne metode* (angl. penalty method). Radi bi poiskali numerično rešitev parcialne diferencialne enačbe s homogenimi robnimi pogoji,

$$\begin{cases} \Delta u(x,y) = 1, & (x,y) \in \Omega \\ u(x,y) = 0, & (x,y) \in \partial\Omega \end{cases}, \tag{1}$$

kjer je u iskana funkcija in $\Omega \subset [-1,1]^2$ netrivialna domena. Pri kaznovalni metodi problem enostavno preoblikujemo na reševanje modificiranega problema

$$\begin{cases} \Delta u(x,y) + k(x,y)u(x,y) = 1, & (x,y) \in [-1,1]^2 \\ u(x,y) = 0, & (x,y) \in \partial[-1,1]^2 \end{cases}, \tag{2}$$

kjer smo vpeljali kaznovalno funkcijo

$$k(x,y) = \begin{cases} k, & (x,y) \notin \Omega \\ 0, & (x,y) \in \Omega \end{cases}$$

za izbrani $k \gg 0$.

Poiščite numerični približek za rešitev PDE (1) preko formulacije (2) za domeno $\Omega = [-1,1]^2 \backslash \{x^2 + y^2 < 1/10\}$. Območje $[-1,1]^2$ razdelite na enakomerno $n \times n$ kvadratno mrežo, nato parcialne odvode aproksimirajte s 5-točkovno shemo, dobljeno iz končnih diferenc.

- Zapišite razpršen linearen sistem $A\boldsymbol{x} = \boldsymbol{b}$ in ga rešite z vgrajeno metodo za reševanje linearnih sistemov.
- Obravnajte numerične rešitve v odvisnosti od n in k.
- Smiselno razbijte matriko A na $A = M + N$, kjer je reševanje sistema z matriko M hitrejše kot z A, in poizkusite rešiti sistem $A\boldsymbol{x} = \boldsymbol{b}$ s kakšno od običajnih iterativnih metod.

Naloga 2

Na predavanjih smo obravnavali metodo D-Lanczos, ki jo najdete tudi v knjigi [1] na strani 156. V originalnem algoritmu se izvaja LU algoritem brez pivotiranja, zaradi česar se lahko algoritem kritično zaustavi. Možna rešitev je uporaba QR razcepa.

Razvijte varianto D-Lanczos z Givensovimi rotacijami in jo implementirajte (v Matlabu). Na manjših primerih najprej preverite, da metoda pravilno deluje. Potem uporabite metodo na primerih `ncvxbqp` (1304), `spmsrtls` (1309) in `wing` (2532) iz spletne učilnice s toleranco 10^{-10} in z maksimalnim številom korakov 5000. Za desno stran vzemite $b = Ae$, kjer je e vektor samih enic. Narišite grafe konvergenc.

Nasveti:

- Pazite, da v spominu hranite le tiste vektorje, ki jih potrebujete.
- Pri eksaktnem računanju bi se morali približki ujemati s približki, ki jih dobimo z metodo FOM. Tako lahko z obema metodama naredite le nekaj korakov in preverite, če se rešitvi ujemata (do vpliva zaokrožitvenih napak), da testirate vašo metodo.

Naloga 3

Iz zbirke SSMC naložite naslednje matrike (na voljo so tudi v spletni učilnici): `c-63` (1224), `gridgena` (1311) in `RFdevice` (1877). Za vsako izmed matrik poskusite rešiti linearni sistem $Ax = b$ z iterativnimi metodami, ki so na voljo v Matlabu: `gmres, minres, pcg, bicg, qmr, symmlq, bicgstab`. Če desna stran ni podana zraven problema, vzemite $b = Ae$, kjer je e vektor samih enic. Za začetni približek izberite $x_0 = 0$. Na SSMC si oglejte, kakšne so lastnosti teh treh matrik, da ne boste npr. uporabljali metode konjugiranih gradientov za matriko, ki ni s.p.d.

- Ugotovite, katere metode so primernejše za katero matriko in to obrazložite.
- Pri metodah, kjer je možno uporabiti ponoven zagon (kot npr. GMRES), ugotovite optimalno izbiro maksimalne velikosti podprostorov in raziščite vpliv izbire velikosti na konvergenco.
- Če je konvergenca počasna, poskusite poiskati ustrezno predpogojevanje, ki izboljša konvergenco. Poskusite npr. z nepopolnim LU razcepom ali nepopolnim razcepom Choleskega.
- Ali lahko z iterativnimi metodami dobite, če uporaben približek (npr. da bo izračunani x točen na vsaj tri decimalke) hitreje kot Matlab z direktno metodo `x=A\b`.

Naloga 4

Naj bodo $c_1c_2c_3c_4$ zadnje 4 števke vaše vpisne številke in $V = 4 * c_1c_2 + c_3c_4$.

Metodo konjugiranih gradientov lahko uporabimo tudi za iskanje približka rešitve nedoločenega sistema $Ax = b$ (matrika A je kvadratna in singularna) preko regularizacije. Za reševanje takega sistema si lahko pomagate z Matlabovo funkcijo `pcg`.

Algoritem preizkusite na realnem problemu, ki nastane pri računanju približka Mahalanobis razdalje pri ocenjevanju kvalitete rudarjenja podatkov. Radi bi izračunali

$$X_{\text{test}}^{\top}(X_{\text{train}}X_{\text{train}}^{\top})^{+}X_{\text{test}}.$$

Obe matriki X_{***} imata veliko večje število vrstic kot stolpcev in sta razpršeni. V Matlab ju naložite z ukazom `load mahalanobis`, kjer datoteko `mahalanobis` dobite na spletni učilnici.

Namesto računanja psevdoinverza singularne matrike $X_{\text{train}}X_{\text{train}}^{\top}$ raje uporabimo regularizacijo. Tako bi namesto z $(X_{\text{train}}X_{\text{train}}^{\top})^{+}$ radi implicitno računali z inverzom regularizirane matrike $A = (1-\alpha)X_{\text{train}}X_{\text{train}}^{\top} + \alpha I$ ter izračunali

$$X_{\text{test}}^{\top}A^{-1}X_{\text{test}}.$$

Množenje z matriko A je potrebno implementirati implicitno, saj bi matrika $X_{\text{train}}X_{\text{train}}^{\top}$ lahko bila prevelika, da bi jo shranili v pomnilnik.

Narišite graf števila korakov potrebnih za konvergenco metode konjugiranih gradientov za sistem $Ax = b$ v odvisnosti od $\alpha \in (0, 1]$, kjer je b V-ti stolpec matrike X_{test}. Poiščite najmanjši α pri katerem dobite zadovoljivo konvergenco. Pazite: ko povečujemo α, se regularizirana rešitev oddaljuje od prave; možen kriterij za kvaliteto je, da izračunamo ostanek za originalno matriko ($\alpha = 0$) in vektor x, izračunan s pomočjo regularizacije. Smiselno je torej najti α iz desnega roba intervala $(0, 1]$, da bo konvergenca hitra, napaka pa ne prevelika. Ko določite optimalen α, izračunajte celoten produkt $X_{\text{test}}^{T}A^{-1}X_{\text{test}}$.

Veliko uspeha pri reševanju!

Literatura

[1] Y.Saad, Iterative methods for sparse linear systems. Na voljo na `http://www-users.cs.umn.edu/~saad/PS/all_pdf.zip`

[2] T. Davis, Suite Sparse Matrix Collection. Na voljo na `https://sparse.tamu.edu/`