Iterativne metode v numerični linearni algebri
2013/2014

1. domača naloga

Rešitve stisnite v ZIP datoteko z imenom `ime-priimek-vpisna-1.zip` in jih oddajte preko spletne učilnice (`http://ucilnica.fmf.uni-lj.si`) najkasneje do ponedeljka, 23. decembra 2013, do 12. ure. Priložite poročilo, v katerem za vsako nalogo opišete postopek reševanja, zapišete rešitev, in komentirajte rezultat. Če poročilo skenirate, mora biti obvezno oddano v pdf obliki. Rešitvi priložite programe, s katerimi ste naloge rešili in izjavo, da ste naloge reševali samostojno. Naloge naj bodo rešene v Matlabu (uporabite lahko tudi Octave ali Scilab). Priložite programe, s katerimi ste naloge rešili. Programi naj bodo smiselno poimenovani in razporejeni v mapah, ki naj bo bodo poimenovane nal1, nal2, ... K vsaki nalogi spada glavna skripta, ki izpiše rešitve naloge (nal1.m, nal2.m, nal3.m, nal4.m). Prosim, da preverite, če se skripte izvedejo v ukazni vrstici (recimo klic nal1 se mora izvesti brez napak), v nasprotnem boste izgubili polovico točk pri konkretni nalogi. Naloge morajo biti rešene v Matlabu (uporabite lahko tudi Octave ali Scilab), razen če je v nalogi drugače navedeno. Zraven priložite še izjavo, da ste naloge reševali samostojno.

Če imate kakšno vprašanje o nalogah ali Matlabu, se obrnite na asistenta ali profesorja. Če menite, da je vprašanje zanimivo tudi za ostale, uporabite forum. Vsa vprašanja so več kot dobrodošla.

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

Vsaka naloga od 1. do 4. je vredna eno točko, teoretična naloga je vredna 3. točke. Dodatno rešene naloge lahko prinesejo dodatne točke. Skupaj je potrebno za 100%zbrati 4. točke.

1. Naj bo $\sigma > 0$. Numerično reši diferencialno enčbo

$$\begin{aligned} &-u'' + \sigma u' = f, \qquad\qquad 0 < x < 1. \\ &u(0) = \alpha, \quad u(1) = \beta \end{aligned}$$

   Diskretiziraj jo z uporabo simetričnih diferenc:

$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} + \sigma \frac{u_{i+1} - u_i}{h} = f_i.$$

   Velikost h je odvisna od števila točk diskretizacije n, $h = \frac{1}{n+1}$.

   (a) Sistem zapiši v matrični obliki:

$$A\mathbf{x} = \mathbf{f}.$$

Če ne gre drugače, ga lahko zapišeš tudi samo za konkretne podatke. Drugače so α, β, vektor vrednosti funkcije f in σ podatki, ki jih dobite.

(b*) Matrika A ni simetrična, poišči potreben in zadosten pogoj, da obstaja matrika D, da velja

$$\tilde{A} = DAD^{-1} = \tilde{A}^\top.$$

Namig, primerjaj levo in desno stran po komponentah. Kaj velja za d_n/d_1, ko gre $h \to 0$.

(b) Za primer $\sigma = 40$, $n = 100$, $f(x) = e^x$, $\alpha = 1/39$, $\beta = e/39$, uporabi metodo konjugiranih gradientov, SOR, Gauss-Seidlovo in SSOR metodo ter primerjaj natančnost in hitrost konvergence.

(c) Za reševanje sistema uporabi še metodo GMRES. Primerjaj natančnost s tisto v točki (b). Z uporabo merjenja časa oceni še časovno zahtevnost ene iteracije.

Točka z zvezdico je dodatna in ni obvezna.

2. Metodo konjugiranih gradientov lahko uporabimo tudi za reševanje nedoločenega sistema $Ax = b$. Za reševanje tega sistem lahko modificirate kar Matlabovo funkcijo `pcg`.

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

$$X_{\text{test}}^T (X_{\text{train}} X_{\text{train}}^T)^\dagger X_{\text{test}}.$$

   Obe matriki X imata veliko večje število vrstic kot stolpcev in sta razpršeni. Ker je sistem z matriko $X_{\text{train}} X_{\text{train}}^T$ poddoločen, namesto psevdoinverza uporabimo regularizacijo. Tako namesto z $(X_{\text{train}} X_{\text{train}}^T)^\dagger$ računamo z regularizirano matriko $A = (1-\alpha) X_{\text{train}} X_{\text{train}}^T + \alpha I$. Množenje z matriko A je potrebno implementirati implicitno, saj je matrika $X_{\text{train}} X_{\text{train}}^T$ prevelika, da bi jo shranili v pomnilnik. Nariši 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}. Poišči najmanjši α pri katerem dobiš zadovoljivo konvergenco in nato za ta α izračunaj celoten produkt $X_{\text{test}}^T A^{-1} X_{\text{test}}$. Pazi. Ko povečuješ α se rešitev oddaljuje od prave, možen kriterij za kvaliteto je, da izračunate ostanek za originalno matriko in vektor izračunan s pomočjo regularizacije.

Matriki naložite v Matlab z ukazom `load mahalanobis`, kjer datoteko `mahalanobis` dobite na učilnici.

3. Pri reševanju predoločenega sistema $Ax = b$ si lahko pomagamo z normalnim sistemom $A^TAx = A^Tb$. V splošnem to ni najboljša možnost, saj se občutljivost sistema kvadrira. Druga možnost je, da uporabimo prevedbo na reševanje večjega sistema

$$\begin{array}{cc} & \begin{array}{cc} m & n \end{array} \\ \begin{array}{c} m \\ n \end{array} & \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \end{array} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

katerega rešitev ustreza rešitvi predoločenega sistema $Ax = b$ po metodi najmanjših kvadratov. Dobljena matrika je simetrična. Množenje z njo implementirajte implicitno. Nato pa uporabite Lanczosov algoritem ali MINRES za reševanje sistema. Kateri postopek vrača boljše rezultate? Primerjate svojo metodo še z metodama

- LSQR, http://www.stanford.edu/group/SOL/software/lsqr.html,
- LSMR, http://www.stanford.edu/group/SOL/software/lsmr.html,

ki uporabljata Golub-Kahanovo bidiagonalizacijo. Matlabovi funkciji sta že na voljo na zgornjih straneh. Kaj opazite, oziroma kje dobite hitrejšo konvergenco? V zbirki matrik, ki jih lahko dobite z `UFGet`, si izberete primerno razpršeno pravokotno matriko, za b si izberete vektor samih enic. Matrike, ki so primerne so:

$$5, 7, 8, 9, 10, 11, 12, 169, 170, 260, 261, 903, 1860, 1881, 1884 - 1890, 2217$$

in 2218. Za začetek pozkusite z matriko majhnih dimenzij, nato pa preizkusite vaš algoritem še za matriko, ki ima vsaj milijon neničelnih elementov. Matrike lahko dobite tudi v grafičnem vmesniku `UFGui`, če si izberete pri `kind - least squares` in `(all groups)`.

4. Z `UFget` naložite naslednje matrike (na voljo so tudi v spletni učilnici): 1224, 1311 in 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, potem vzemite vektor $b = Ae$, kjer je e vektor samih enic, za začetni približek pa $x_0 = 0$.

   - 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.

5. Naj bo A matrika dimenzije 4×4:

$$A = \begin{bmatrix} 1 & \epsilon & & \\ & -1 & 1/\epsilon & \\ & & 1 & \epsilon \\ & & & -1 \end{bmatrix}$$

Vemo, da za ostanek GMRES metode v najslabšem primeru velja:

$$||r_k^{(w)}||_2 = \max_{||r_0||_2=1} \min_{p_k} ||p_k(A) r_0||_2,$$

kjer je p_k poljuben polinom stopnje $\leq k$ za katerega velja $p_k(0) = 1$, polinom je odvisen od r_0. Za konkreten primer matrike A pokaži, da ne velja:

$$\max_{||r_0||_2=1} \min_{p_k} ||p_k(A) r_0||_2 = ||p_*(A)||_2 = \min_{p_k} ||p_k(A)||_2 \equiv \min_{p_k} \max_{||r_0||_2=1} ||p_k(A) r_0||_2.$$

(a) Pokaži, da je polinom, ki minimizira $\min_{p_k} ||p_k(A)||_2$ enoličen.

(b) Pokaži, da je polinom stopnje ≤ 3, ki ima vrednost 1 v izhodišču, neodvisen od ϵ in enak

$$p_*(z) = 1 - \frac{3}{5} z^2 \quad \text{in} \quad ||p_*(A)|| = \frac{4}{5}.$$

Nasvet. Pokaži, da mora biti polinom sod. Prej preveri še, da je A^T ortogonalno podobna matriki $-A$ za prehodno matriko

$$Q = \begin{bmatrix} & & & -1 \\ & & 1 & \\ & -1 & & \\ 1 & & & \end{bmatrix}.$$

Izkaže se, da lahko singularne vrednosti $p(A) = 1 + \gamma z^2$ potem izračunaš analitično.

(c) Pokaži, da se izraza razlikujeta. Uporabi kar matriko A in pokaži, da za vsak vektor b velja:

$$||p_b(A) b||_2 < ||p_*(A)||_2 ||b||_2,$$

kjer je p_b polinom stopnje ≤ 3, ki ima vrednost 1 v izhodišču in da največji ostanek pri vektorju b.

(d) Pokaži, da za poljuben vektor b, kjer je $||b|| = 1$ obstaja polinom p_b stopnje ≤ 3 z $p_b(0) = 1$, da velja

$$||p_b(A)b|| \leq 4\sqrt{\epsilon} + 5\epsilon.$$

Nasvet. Najprej pokaži, da velja:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \quad Ab = \begin{bmatrix} b_1 \\ -b_2 + b_3/\epsilon \\ b_3 \\ -b_4 \end{bmatrix} + \epsilon \begin{bmatrix} b_2 \\ 0 \\ b_4 \\ 0 \end{bmatrix},$$

$$A^2 b = b + \begin{bmatrix} b_3 \\ b_4 \\ 0 \\ 0 \end{bmatrix}, \quad A^3 b = Ab + \begin{bmatrix} b_3 \\ -b_4 \\ 0 \\ 0 \end{bmatrix} + \epsilon \begin{bmatrix} b_4 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Potem določi polinom, glede na to, če velja $|b_3| \geq \sqrt{\epsilon}$ ali $|b_3| < \sqrt{\epsilon}$. Nasvet: pomagaj si z Mathematico. To pokaže, da sta idealni primer in najslabši primer poljubno blizu.

(e) Ali se kaj bistveno spremeni, če izraz $1/\epsilon$ v matriki A zamenjaš z c/ϵ? Kakšne so nove ocene in ustrezni polinomi?