Vilniaus Universitetas

Tiesinių lygčių sistemos sprendimas

2 laboratorinis darbas

Skaitiniai metodai

Darbą atliko:

Dovydas Martinkus

Duomenų Mokslas 4 kursas 1 gr.

Vilnius, 2022

Turinys

1 Užduoties ataskaita

Reikalinga 0,0001 tikslumu iteraciniais ir variaciniais metodais išspręsti tiesinę lygčių sistemą $\mathrm{Ax}=\mathrm{B}$, kur:

$\mathrm{A} = \mathrm{D} + 0.1(\mathrm{k} + 3)\mathrm{E}$,

$\mathrm{k}=16$,

$$\mathrm{D} = \begin{pmatrix} 0.18 & 0.57 & 0.38 & 0.42 \\ 0.57 & 0.95 & 0.70 & 0.44 \\ 0.38 & 0.70 & 0.37 & 0.18 \\ 0.42 & 0.44 & 0.18 & 0.40 \end{pmatrix}, \ \mathrm{E} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathrm{B} = \begin{pmatrix} 1{,}2 \\ 2 \\ 3 \\ 1{,}5 \end{pmatrix}.$$

Įsistatę prieš tai aprašytas reikšmes gauname sistemos matricą:

$$\mathrm{A} = \begin{pmatrix} 2.08 & 0.57 & 0.38 & 0.42 \\ 0.57 & 2.85 & 0.70 & 0.44 \\ 0.38 & 0.70 & 2.27 & 0.18 \\ 0.42 & 0.44 & 0.18 & 2.30 \end{pmatrix}.$$

Toliau žymėjimuose naudojama:

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}, \ -L = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & 0 \end{pmatrix},$$

$$-U = \begin{pmatrix} 0 & a_{12} & \dots & a_{1,n-1} & a_{1n} \\ 0 & 0 & \dots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{n-1,n} \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

1.1 Zeidelio metodas

Žemiau pateikti Zeidelio metodu gauti iteracijų rezultatai 5 skaičių po kablelio tikslumu:

1 lentelė. Iteracijų rezultatai naudojant Zeidelio metodą

k	x_1^k	x_2^k	x_3^k	x_4^k	$\|\|Ax^k-b\|\|_\infty$
0	0,00000	0,00000	0,00000	0,00000	3,00000
1	0,57692	0,58637	1,04419	0,35293	0,88622
2	0,15421	0,35996	1,15679	0,46462	0,12796
3	0,.17313	0,31128	1,15977	0,47025	0,02425
4	0,18479	0,30734	1,15859	0,46896	0,00323
5	0,18634	0,30752	1,15838	0,46866	0,00028
6	0,18639	0,30761	1,15836	0,46864	0,00004

Konvergavimas pasiektas po 6 iteracijų. Patikrinimui gautą artinį įstatome į lygčių sistemą ir palyginame gautus rezultatus (5 skaičių po kablelio tikslumu):

2 lentelė. Gauto artinio patikrinimas jį įstatant į lygčių sistemą

	b_1	b_2	b_3	b_4
Gautas B	1,20003	1,9999	2,99999	1,5
Norimas B	1,20000	2,0000	3,00000	1,5

Pagal teoremą turime, kad iteracinė seka $x_{k+1} = Cx_k + d, k > 0$, konverguoja į $Ax = b$ sprendinį su bet kokiu pirminiu artiniu jeigu $max\{|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|\} < 1$, kur λ_n – matricos C tikrinės reikšmės.

Zeidelio metodo atveju turime iteracinę seką $x_{k+1} = (D\text{-}L)^{-1}(Ux_k + b)$, vadinasi $C = (D\text{-}L)^{-1}U$.

Apskaičiavę $max\{|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|\}$ gauname reikšmę 0,1269249, todėl turima iteracinė seka konverguoja su bet kokiu pradiniu artiniu.

1.2 Jungtinių gradientų metodas

Primename, kad turime matricą:

$$A = \begin{pmatrix} 2.08 & 0.57 & 0.38 & 0.42 \\ 0.57 & 2.85 & 0.70 & 0.44 \\ 0.38 & 0.70 & 2.27 & 0.18 \\ 0.42 & 0.44 & 0.18 & 2.30 \end{pmatrix}.$$

Matome, kad matrica A yra simetrinė, be to ji ir teigiamai apibrėžta: ji simetrinė ir jos tikrinės reikšmės (apytiksliai λ_1=3,85, λ_2=2,17, λ_3=1,79 ir λ_4=1,67) visos yra realiosios ir didesnės už 0.

Žemiau pateikti jungtinių gradientų metodu gauti iteracijų rezultatai 5 skaičių po kablelio tikslumu:

3 lentelė. Iteracijų rezultatai naudojant jungtinių gradientų metodą

k	x_1^k	x_2^k	x_3^k	x_4^k	$\|\|Ax^k-b\|\|_\infty$
0	0,00000	0,00000	0,00000	0,00000	3,00000
1	0,33579	0,55965	0,83948	0,41974	0,55873
2	0,19674	0,30824	1,16337	0,44882	0,04005
3	0,18854	0,30627	1,15819	0,46903	0,00385
4	0,18637	0,30762	1,15837	0,46864	0,00000

Konvergavimas pasiektas po 4 iteracijų. Patikrinimui gautą artinį įstatome į lygčių sistemą ir palyginame gautus rezultatus (neapvalintus):

4 lentelė. Gauto artinio patikrinimas jį įstatant į lygčių sistemą

	b_1	b_2	b_3	b_4
Gautas B	1,2	2	3	1,5
Norimas B	1,2	2	3	1,5

Matome, kad gautas tikslus lygčių sistemos Ax=B sprendinys.

Galiausiai lentelėje pateiktas Zeidelio, jungtinių gradientų ir naudojant R biblioteką *Rlinsolve* gautų artinių palyginimas. Kaip matome visais metodais gauti beveik identiški rezultatai:

5 lentelė. Skirtingais metodais gautų artinių palyginimas

Metodas	x_1	x_2	x_3	x_4
Zeidelio	0,1863910	0,3076080	1,158364	0,4686360
Jungtinių gradientų	0,1863705	0,3076182	1,158366	0,4686376
R iteracinis *lsolve.jacobi()*	0,1863781	0,3075950	1,158385	0,4686426
R variacinis *lsolve.cg()*	0,1863705	0,3076182	1,158366	0,4686376

Priedas

Žemiau pateiktas naudotas programinis kodas:

```
# Dovydas Martinkus
# Duomenų Mokslas 4k. 1gr.
# 2 uzduotis

###


# Funkciju aprasymas

zeidelio <- function(x0,D,L,U,B,eps) {
  x <- matrix(x0)
  n <- 0

  repeat {
    x_n <- solve(D-L) %*% (U %*% matrix(x[,n+1])+B)
    x <- cbind(x, x_n)


    if (norm(matrix(matrix(x[,n+2]) - matrix(x[,n+1]),
                    byrow=TRUE),
             type = "M") > eps) {
       n <- n + 1
    }
    else {
      return(t(x))
    }
  }
}



jungtiniu_gradientu <- function(x0,A,B,eps) {
  x <- matrix(x0)
  z <- A %*% x - B
  p <- z
  r <- A %*% p
  tau <- c((t(z) %*% z) / (t(r) %*% p))
  n <- 0

  repeat {


    if( n>0 ){
      r_n <- A %*% matrix(p[,n+1])
      r <- cbind(r, r_n)
      tau_n <- (t(matrix(z[,n+1])) %*% matrix(z[,n+1])) / (t(matrix(r[,n+1])) %*% matrix(p[,n+1]))
      tau <- cbind(tau, tau_n)
     }


    x_n <- matrix(x[,n+1]) - tau[n+1]*matrix(p[,n+1])
    z_n <- matrix(z[,n+1]) - tau[n+1]*matrix(r[,n+1])
```

```
    x <- cbind(x, x_n)
    z <- cbind(z, z_n)


    if ( t(z_n) %*% z_n >= eps^2 ) {
      beta_n <- c((t(z_n) %*% z_n) / ( t(matrix(z[,n+1])) %*%  matrix(z[,n+1])))
      p_n <- z_n + beta_n*matrix(p[,n+1])
      p <- cbind(p, p_n)
      n <- n + 1
    }
    else {
      return(list(a=t(x),b=t(p)))
    }
  }
}

## lentele, parodanti artiniu paklaidas pagal iteracijas
lentele <- function(xn,A,B) {

  norms <- apply(xn,1,function(x) norm(matrix(A %*% x - B,byrow=TRUE),type="M"))
  xn <- cbind(0:(nrow(xn)-1),xn,norms)
  return(xn)
}


D <- matrix(c(0.18, 0.57, 0.38, 0.42,
              0.57, 0.95, 0.70, 0.44,
              0.38, 0.70, 0.37, 0.18,
              0.42, 0.44, 0.18, 0.40),
            ncol=4,nrow=4)


E <- diag(1, 4, 4)


B <- matrix(c(1.2,2,3,1.5))


A <- D + 0.1*(16+3)*E


U <- A
U[lower.tri(U,TRUE)] <- 0
U <- U * -1


L <- A
L[upper.tri(L,TRUE)] <- 0
L <- L * -1


D <- A
D[!(lower.tri(D,TRUE) & upper.tri(D,TRUE))] <- 0
D <- D


eps <- 0.0001

x0 <- rep(0,length(B)) # bet koks pradinis artinys
```

```
# Zeidelio metodas


## konvergavimo sąlygų patikrinimas

max(abs(eigen(solve(D-L)%*%U)$values))



xn_1 <- zeidelio(x0,D,L,U,B,eps)

round(lentele(xn_1,A,B),5)

#matrix(apply(diff(xn_1),1,function(y) norm(matrix(y),type="M")))



palyginimas <- cbind(A %*% matrix(xn_1[nrow(xn_1),]),B)
colnames(palyginimas) <- c("Gautas B","Norimas B")
round(t(palyginimas),6)




# Jungtiniu gradientu metodas


## konvergavimo sąlygų patikrinimas


eigen(A)$values


result <- jungtiniu_gradientu(x0,A,B,eps)

xn_2 <- result[[1]]
p_n <- result[[2]]


round(lentele(xn_2,A,B),50)


palyginimas <- cbind(A %*% matrix(xn_2[nrow(xn_2),]),B)
colnames(palyginimas) <- c("Gautas B","Norimas B")
t(palyginimas)



## patikriname ar gauti p_n tikrai sudaro matricos A atzvilgiu jungtiniu vektoriu sistema

for(i in 1:nrow(p_n)) {
 for(j in 1:nrow(p_n)) {
   if(i != j) {
     print(all.equal(matrix(0)
                            ,t(A%*%p_n[i,]) %*% p_n[j,]))
   }
 }
}


library(Rlinsolve)
```

```
# Artiniu palyginimas

palyginimas <- cbind(matrix(xn_1[nrow(xn_1),]),matrix(xn_2[nrow(xn_2),]),lsolve.jacobi(A,B)$x,
lsolve.cg(A,B)$x)
colnames(palyginimas) <- c("Zeidelio","Jungtinių gradientų","R iteracinis metodas lsolve.jacobi","R
variacinis metodas lsove.cg")
t(palyginimas)
```