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 3](#_Toc117178654)

[1.1 Zeidelio metodas 4](#_Toc117178655)

[1.2 Jungtinių gradientų metodas 5](#_Toc117178656)

[Priedas 7](#_Toc117178657)

# Užduoties ataskaita

Reikalinga 0,0001 tikslumu iteraciniais ir variaciniais metodais išspręsti tiesinę lygčių sistemą Ax=B, kur:

A = D + 0.1(k+ 3)E,

k=16,

D = , E = , B = .

Įsistatę prieš tai aprašytas reikšmes gauname sistemos matricą:

A = .

Toliau žymėjimuose naudojama:

, ,

.

## Zeidelio metodas

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

1 lentelė. Iteracijų rezultatai naudojant Zeidelio metodą

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *k* | *x1k* | *x2k* | *x3k* | *x4k* | *||Axk-b||∞* |
| 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | b1 | b2 | b3 | b4 |
| Gautas B | 1,20003 | 1,9999 | 2,99999 | 1,5 |
| Norimas B | 1,20000 | 2,0000 | 3,00000 | 1,5 |

Pagal teoremą jei ||C|| < 1, tai iteracinė seka xk+1 = Cxk + d, k > 0,konverguoja į Ax = bsprendinį kiekvienam pradiniam artiniui x0.

Zeidelio metodo atveju turime iteracinę seką xk+1= (D-L)-1(Uxk + b), vadinasi C = (D-L)-1 U.

Apskaičiavę ||C|| reikšmę gauname||C||=0,5527425 < 1, todėl sudaryta iteracinė seka konverguoja su bet kokiu pradiniu artiniu.

Pagal kitą teoremą turime, kad iteracinė seka konverguoja su bet kokiu pirminiu artiniu jeigu , kur – matricos C tikrinės reikšmės.

Apskaičiavę gauname reikšmę 0,1269249, todėl ir pagal šią teoremą matome, kad iteracinė seka konverguoja su bet kokiu pradiniu artiniu.

## Jungtinių gradientų metodas

Primename, kad turime matricą:

A = .

Matome, kad matrica A yra simetrinė, be to ji ir teigiamai apibrėžta: ji simetrinė ir jos tikrinės reikšmės (apytiksliai λ1=3,85, λ2=2,17, λ3=1,79 ir λ4=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* | *x1k* | *x2k* | *x3k* | *x4k* | *||Axk-b||∞* |
| 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | b1 | b2 | b3 | b4 |
| Gautas B | 1,2 | 2 | 3 | 1,5 |
| Norimas B | 1,2 | 2 | 3 | 1,5 |

Matome, kad gautas tikslus lygčių sistemos Ax=Bsprendinys.

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 | x1 | x2 | x3 | x4 |
| 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))

norm(solve(D-L)%\*%U)

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)