Vilniaus Universitetas

2 laboratorinis

Skaitiniai metodai

Darbą atliko:

Dovydas Martinkus

Duomenų Mokslas 4 kursas 1 gr.

Vilnius, 2022

**Turinys**

[1 Užduoties ataskaita 3](#_Toc114515806)

[1.1 Pusiaukirtos metodas 3](#_Toc114515807)

[1.2 Niutono metodas 5](#_Toc114515808)

[Priedas 8](#_Toc114515809)

# Užduoties ataskaita

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

A = ,

B = .

Šiam tikslui buvo naudojami Zeidelio ir jungtinių gradientų metodai.

## Zeidelio metodas

1 lentelė Iteracijų rezultatai naudojant Zeidelio metodą

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *n* | *x1* | *x2* | *x3* | *x4* | *||xn-x||∞* |
| 0 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 7.70000 |
| 1 | 0.57692 | 0.58637 | 1.04419 | 0.35293 | 1.82900 |
| 2 | 0.15421 | 0.35996 | 1.15679 | 0.46462 | 0.18742 |
| 3 | 0.17313 | 0.31128 | 1.15977 | 0.47025 | 0.02983 |
| 4 | 0.18479 | 0.30734 | 1.15859 | 0.46896 | 0.00486 |
| 5 | 0.18634 | 0.30752 | 1.15838 | 0.46866 | 0.00044 |
| 6 | 0.18639 | 0.30761 | 1.15836 | 0.46864 | 0.00006 |

Konvergavimas pasiektas po 6 iteracijų. Patikrinimui gautą artinį įstatome į lygčių sistemą ir palyginame gautus rezultatus:

2 lentelė Gauto artinio patikrinimas jį įstatant į lygčių sistemą

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | b1 | b2 | b3 | b4 |
| Gautas B | 1.200035 | 1.99998 | 2.999995 | 1.5 |
| Norimas B | 1.200000 | 2.00000 | 3.000000 | 1.5 |

Jei *||C|| < 1*, tai iteracinė seka *xk+1 = Cxk + d, k > 0,* konverguoja į *Ax = b* sprendinį kiekvienam pradiniam artiniui x0.

Zeidelio metodo atveju turime iteracinę seką *xk+1= D−1[(L + U)xk + b] = D−1[(D − A)xk + b],* vadinasi

*C = D−1(L + U).*

Apskaičiavę *||C||* reikšmę gauname *||C|| = 0.5527425 < 1,* todėl sudaryta iteracinė seka konverguoja su bet kokiu pradiniu artiniu.

## Jungtinių gradientų metodas

3 Lentelė Iteracijų rezultatai naudojant jungtinių gradientų metodą

|  |  |  |  |
| --- | --- | --- | --- |
| n | xn | f(xn) | |xn-xn+1| |
| 0 | 0 | -3 | 0 |
| 1 | 3 | 243 | 3 |
| 2 | 2.4014778 | 79.2731746 | -0.5985222 |
| 3 | 1.9276308 | 25.5421894 | -0.473847 |
| 4 | 1.5629215 | 7.8887151 | -0.3647093 |
| 5 | 1.3070809 | 2.1222369 | -0.2558406 |
| 6 | 1.1709893 | 0.3727224 | -0.1360916 |
| 7 | 1.1351546 | 0.0199975 | -0.0358347 |
| 8 | 1.1330049 | 6.75e-5 | -0.0021498 |
| 9 | 1.1329976 | 0 | -7.3e-6 |

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

4 lentelė Skirtingais būdais gautų artinių palyginimas

|  |  |  |
| --- | --- | --- |
| Pusiaukirtos metodas | Niutono metodas | uniroot() |
| 1.133087 | 1.132998 | 1.133026 |

# Priedas

Žemiau pateiktas naudotas programinis kodas:

# Dovydas Martinkus

# Duomenų Mokslas 4k. 1gr.

func <- function(x) {

x^5 + x - 3

}

derivative <- function(x) {

5\*x^4 + 1

}

####

intervalas <- function(an, bn, cn, func) {

if (func(an) \* func(cn) < 0) {

return(c(an, cn))

} else {

return(c(cn, bn))

}

}

pusiaukirtos <- function(a0, b0, func, eps) {

n <- 0

c0 <- mean(c(a0, b0))

a <- a0

b <- b0

c <- c0

repeat {

if ((abs(a[n+1] - b[n+1]) / 2 > eps)) {

if (func(c[n+1]) == 0) {

return(data.frame(a,b,c))

}

naujas\_intervalas <- intervalas(a[n+1], b[n+1], c[n+1], func)

a <- c(a,naujas\_intervalas[1])

b <- c(b,naujas\_intervalas[2])

c\_n <- mean(c(a[n+2],b[n+2]))

c <- c(c, c\_n)

n <- n + 1

} else {

return(data.frame(a,b,c))

}

}

}

pusiaukirtos\_lentele <- function(x) {

cbind(n=seq(0,lengths(x)[1]-1),

x,

y=func(x$c),

abs(x$a-x$b)/2)

}

####

niutono <- function(x0, func, deriv, eps) {

x <- x0

n <- 0

repeat {

x\_n <- x[n+1] - func(x[n+1]) / deriv(x[n+1])

x <- c(x, x\_n)

if ((abs(x[n+2] - x[n+1]) > eps)) {

n <- n + 1

} else {

return(x)

}

}

}

niutono\_lentele <- function(x) {

data.frame(n=seq(0,length(x)-1),

x\_n =x,

y = func(x),

`abs(x\_n-x\_n+1)`=c(0,diff(x)),

check.names = FALSE)

}

library(ggplot2)

library(ggrepel)

library(latex2exp)

#####

eps <- 0.0001

# pradinis funkcijos grafikas

ggplot(data.frame(x = seq(-3, 3, 0.1)), aes(x)) +

geom\_function(fun = func, colour = "black") +

geom\_hline(yintercept = 0, color = "red") +

theme\_minimal(base\_size = 16) +

labs(title=TeX("x^5+x-3"),subtitle = "Pradinis funkcijos grafikas")

# išvestinės grafikas

ggplot(data.frame(x = seq(-3, 3, 0.1)), aes(x)) +

geom\_function(fun = derivative, colour = "black") +

geom\_hline(yintercept = 0, color = "red") +

theme\_minimal(base\_size = 16) +

labs(title=TeX("5x^4+1"),subtitle = "Išvestinės grafikas")

####

res <- pusiaukirtos(-1,2,func)

func(res$c[length(res$c)])

ggplot(data.frame(x = seq(-2, 2, 0.1)), aes(x)) +

geom\_function(fun = func, colour = "black") +

geom\_hline(yintercept = 0, color = "red") +

theme\_minimal(base\_size = 16) +

geom\_point(data=pusiaukirtos\_lentele(res)[1:8,],

aes(x=c,y=y)) +

geom\_text\_repel(data=pusiaukirtos\_lentele(res)[1:8,],aes(x=c,y=y,label=paste0("x",n))) +

labs(title=TeX("x^5+x-3"),subtitle = "Pusiaukirtos metodas")

pusiaukirtos\_lentele(res)

####

xn <- niutono(0,func,derivative,eps)

func(xn[length(xn)])

ggplot(data.frame(x = seq(-3, 3, 0.1)), aes(x)) +

geom\_function(fun = func, colour = "black") +

geom\_hline(yintercept = 0, color = "red") +

theme\_minimal(base\_size = 16) +

geom\_point(data=niutono\_lentele(xn)[1:8,],

aes(x=x\_n,y=y)) +

geom\_text\_repel(data=niutono\_lentele(xn)[1:8,],aes(x=x\_n,y=y,label=paste0("x",n))) +

labs(title=TeX("x^5+x-3"),subtitle = "Niutono metodas")

niutono\_lentele(xn)

####

palyginimas <- c(tail(res$c,1),tail(xn,1),uniroot(func,c(-3,3))$root)

names(palyginimas) <- c("Pusiaukirtos","Niutono","Uniroot()")

palyginimas