Тестирование функций библиотеки IMSSLAE

suppressPackageStartupMessages(library(IMSSLAE))

# Проверка функции muFind()

Функция **muFind()** осуществляет поиск центра круга для выпуклой оболочки спектра линейного оператора, а также производит отрисовку комплексной плоскости вместе с найденной фигурой.

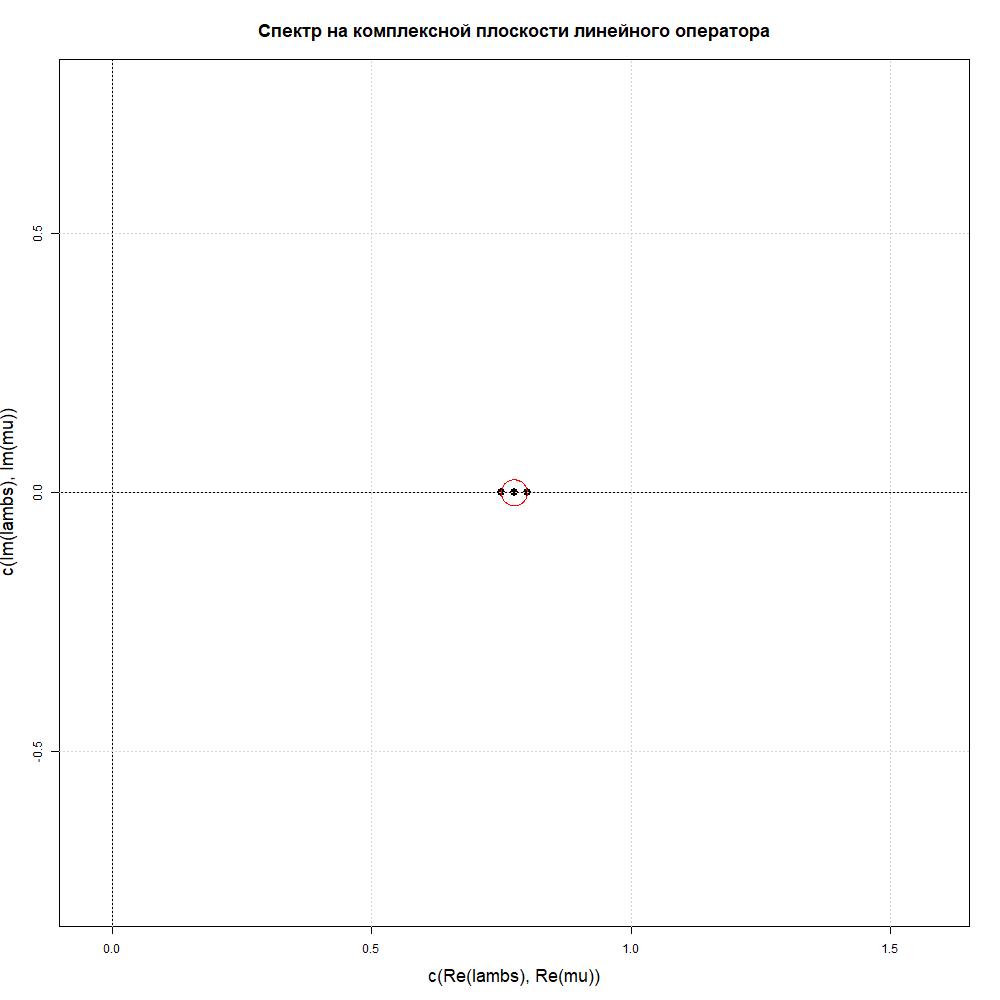
Случай случайного задания спектра:

# Зададимся случайным распределением точек спектра  
lambs <- rnorm(15, 15, 2) + 1i \* rnorm(15, 15, 2)  
# Выведем значения собственных числел  
cat(lambs,"\n")

## 16.01529+11.48653i 15.16073+15.1803i 17.761+13.0843i 9.94383+16.79527i 13.93153+16.61426i 16.65654+15.09552i 11.60252+15.69546i 16.61398+17.5128i 15.59456+13.34176i 13.28375+16.97216i 16.37167+15.98356i 13.80049+12.74095i 11.66842+11.77686i 17.06471+11.89077i 11.44304+12.5982i

# Найдем итерационный параметр  
  
print(IMSSLAE::muFind(lambs = lambs, draw = T))

## [1] 13.71295+14.646i



alt text

Случай наличия всего спектра на отрезке действительной оси:

lambs <- rnorm(50, 100, 10)  
print(IMSSLAE::muFind(lambs))

## [1] 101.6538+0i

# Численный эксперимент для конкретных задач решения СЛАУ

Рассмотрим задачу, с заданными оператором A, вектором свободных членов f, начальным приближением u. Будем также считать, что все собственные значения спектра оператора нам известны, а в программе воспользуемся базовой функцией для ее определения в пользу демострации работы алгоритмов.

# Зададимся начальными условиями  
A <- cbind(c(3, 4, 0.2),  
 c(-4, 8, 0.1),  
 c(2.9, -3, 1))  
u <- c(1, 0.5, 1.7)  
f <- c(1.8, 9.1, 1.3)  
lambs <- eigen(A)$values  
mu <- muFind(lambs = lambs, draw = T)[1]  
cat("Матрица A:\n")

## Матрица A:

print(A)

## [,1] [,2] [,3]  
## [1,] 3.0 -4.0 2.9  
## [2,] 4.0 8.0 -3.0  
## [3,] 0.2 0.1 1.0

cat("\nВектор u равен: u =( ", u, ")\nВектор f равен: f =( ", f, ")\nВектор lambs равен: l =( ", lambs,")\n")

##   
## Вектор u равен: u =( 1 0.5 1.7 )  
## Вектор f равен: f =( 1.8 9.1 1.3 )  
## Вектор lambs равен: l =( 5.498316+3.074874i 5.498316-3.074874i 1.003368+0i )

На данных начальных условиях рассмотрим задачу поиска неивестного вектора u итерационными методами

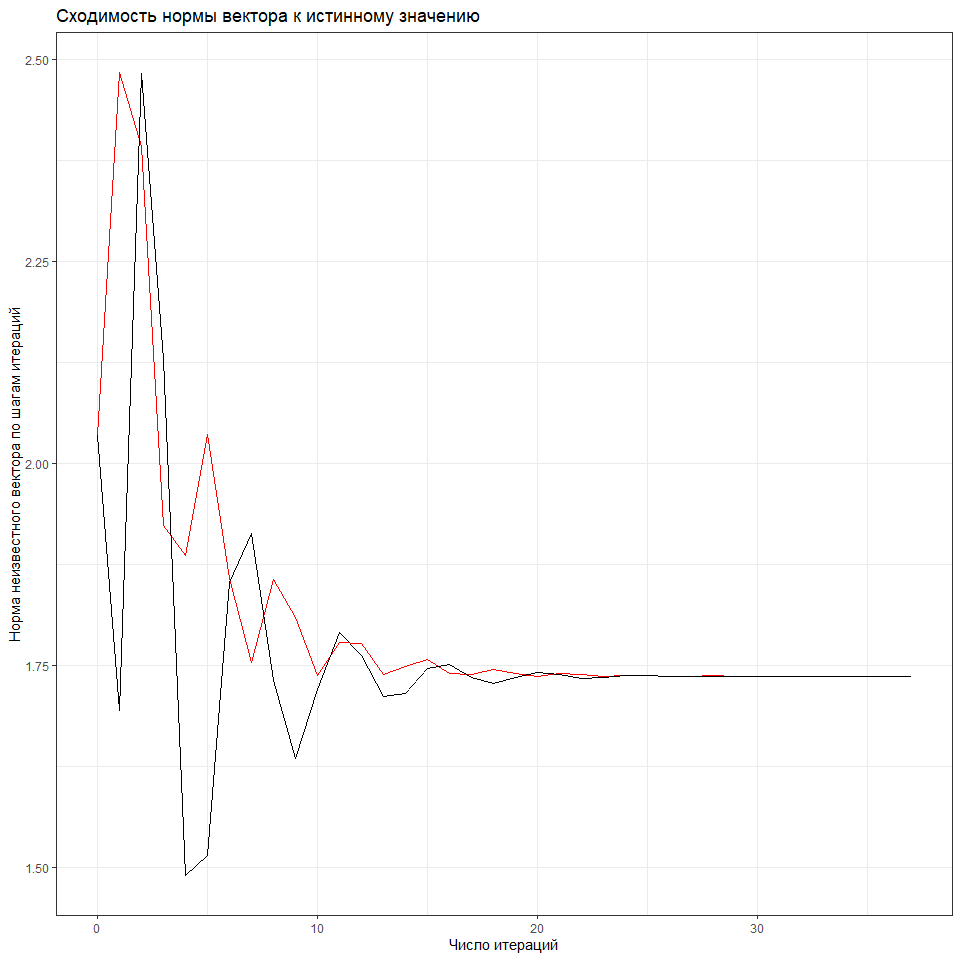
library(rbenchmark)  
library(ggplot2)  
# При помощи rbenchmark будем определять время работы алгоритмов  
r1 <- rbenchmark::benchmark({  
 result11 <- GMSI(A = A, u = u, f = f, mu = mu, eps = 10e-6)  
}, replications = 1000)  
r4 <- rbenchmark::benchmark({  
 result13 <- Jacobi(A, f, u, eps = 10e-6)  
}, replications = 1000)  
  
print(r1)

## test  
## 1 {\n result11 <- GMSI(A = A, u = u, f = f, mu = mu, eps = 1e-05)\n}  
## replications elapsed relative user.self sys.self user.child sys.child  
## 1 1000 0.45 1 0.46 0 NA NA

print(r4)

## test replications elapsed  
## 1 {\n result13 <- Jacobi(A, f, u, eps = 1e-05)\n} 1000 0.23  
## relative user.self sys.self user.child sys.child  
## 1 1 0.23 0 NA NA

resultGMSI <- GMSI.history(A, f, u, mu, eps = 10e-5)  
resultJacobi <- Jacobi.history(A, f, u, eps = 10e-5)  
  
ggplot2::ggplot() +  
 geom\_line(aes(x = 0:(resultGMSI$num.iter), y = sqrt(apply(Re(resultGMSI$var.hist^2), 2, sum))), color = "red") +  
 geom\_line(aes(x = 0:(resultJacobi$num.iter), y = sqrt(apply(Re(resultJacobi$var.hist^2), 2, sum))), color = "black") +  
 theme\_bw() +  
 labs(title = "Сходимость нормы вектора к истинному значению", x = "Число итераций", y = "Норма неизвестного вектора по шагам итераций")



print(paste(resultGMSI$systime.iter, resultJacobi$systime.iter))

## [1] "0 0"

# Зададимся начальными условиями  
A <- cbind(c(0.75, 0), c(-1, 0.8))  
f <- c(1, 1)  
u <- c(4.4, 1.57)  
lambs <- eigen(A)$values  
mu <- muFind(lambs = lambs, draw = T)[1]  
cat("Матрица A:\n")

## Матрица A:

print(A)

## [,1] [,2]  
## [1,] 0.75 -1.0  
## [2,] 0.00 0.8

cat("\nВектор u равен: u =( ", u, ")\nВектор f равен: f =( ", f, ")\nВектор lambs равен: l =( ", lambs,")\n")

##   
## Вектор u равен: u =( 4.4 1.57 )  
## Вектор f равен: f =( 1 1 )  
## Вектор lambs равен: l =( 0.8 0.75 )

r1 <- rbenchmark::benchmark({  
 result11 <- GMSI(A = A, u = u, f = f, mu = mu, eps = 10e-6)  
}, replications = 1000)  
r2 <- rbenchmark::benchmark({  
 result12 <- GMSI.mu(A = A, u = u, f = f, lambs = lambs, eps = 10e-6)  
}, replications = 1000)  
r4 <- rbenchmark::benchmark({  
 result13 <- Jacobi(A, f, u, eps = 10e-6)  
}, replications = 1000)  
r5 <- rbenchmark::benchmark({  
 result14 <- SIM(A, f, u, eps = 10e-6)  
}, replications = 1000)  
  
print(r1)

## test  
## 1 {\n result11 <- GMSI(A = A, u = u, f = f, mu = mu, eps = 1e-05)\n}  
## replications elapsed relative user.self sys.self user.child sys.child  
## 1 1000 0.06 1 0.06 0 NA NA

print(r2)

## test  
## 1 {\n result12 <- GMSI.mu(A = A, u = u, f = f, lambs = lambs, eps = 1e-05)\n}  
## replications elapsed relative user.self sys.self user.child sys.child  
## 1 1000 0.08 1 0.08 0 NA NA

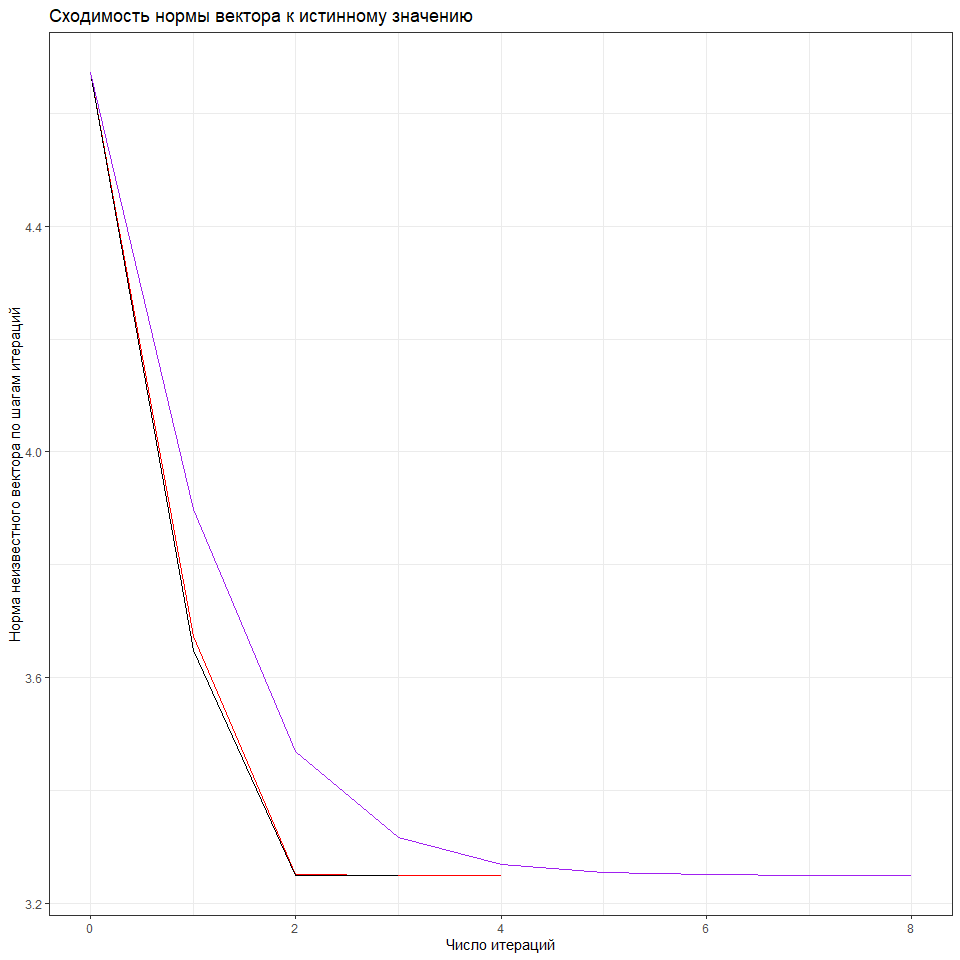
print(r4)

## test replications elapsed  
## 1 {\n result13 <- Jacobi(A, f, u, eps = 1e-05)\n} 1000 0.04  
## relative user.self sys.self user.child sys.child  
## 1 1 0.05 0 NA NA

print(r5)

## test replications elapsed relative  
## 1 {\n result14 <- SIM(A, f, u, eps = 1e-05)\n} 1000 0.11 1  
## user.self sys.self user.child sys.child  
## 1 0.11 0 NA NA

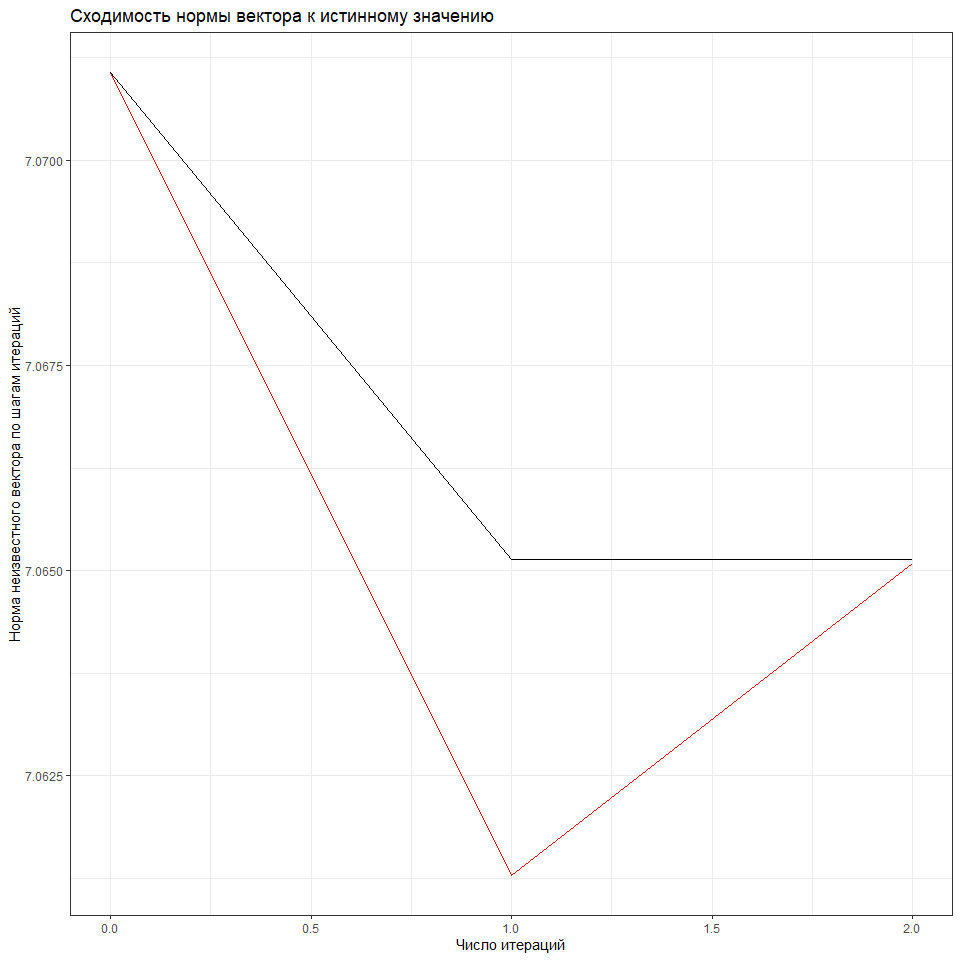
resultGMSI <- GMSI.history(A, f, u, mu, eps = 10e-5)  
resultJacobi <- Jacobi.history(A, f, u, eps = 10e-5)  
resultSIM <- SIM.history(A, f, u, eps = 10e-5)  
  
ggplot2::ggplot() +  
 geom\_line(aes(x = 0:(resultGMSI$num.iter), y = sqrt(apply(Re(resultGMSI$var.hist^2), 2, sum))), color = "red") +  
 geom\_line(aes(x = 0:(resultJacobi$num.iter), y = sqrt(apply(Re(resultJacobi$var.hist^2), 2, sum))), color = "black") +  
 geom\_line(aes(x = 0:(resultSIM$num.iter), y = sqrt(apply(Re(resultSIM$var.hist^2), 2, sum))), color = "purple") +  
 theme\_bw() +  
 labs(title = "Сходимость нормы вектора к истинному значению", x = "Число итераций", y = "Норма неизвестного вектора по шагам итераций")



print(paste(resultGMSI$systime.iter, resultJacobi$systime.iter))

## [1] "0 0"

n <- 50  
A <- diag(rnorm(n, 2, 0.05), nrow = n, ncol = n)  
f <- rnorm(n, 2, 0.05)  
u <- rep(1, n)  
  
resultGMSI <- GMSI.mu.history(A, f, u, lambs = diag(A), eps = 10e-5)  
resultJacobi <- Jacobi.history(A, f, u, eps = 10e-5)  
#resultSIM <- SIM.history(A, f, u, eps = 10e-5)  
ggplot2::ggplot() +  
 geom\_line(aes(x = 0:(resultGMSI$num.iter), y = sqrt(apply(Re(resultGMSI$var.hist^2), 2, sum))), color = "red") +  
 geom\_line(aes(x = 0:(resultJacobi$num.iter), y = sqrt(apply(Re(resultJacobi$var.hist^2), 2, sum))), color = "black") +  
 theme\_bw() +  
 labs(title = "Сходимость нормы вектора к истинному значению", x = "Число итераций", y = "Норма неизвестного вектора по шагам итераций")



print(paste(resultGMSI$systime.iter, resultJacobi$systime.iter))

## [1] "0.285501003265381 0.0019841194152832"