Отчет

Метод вращения якоби

Даниил Игнатьев

Выполнил:  
Игнатьев Даниил  
Студент группы 3630102/80004

Оглавление

[Формулировка задачи 2](#_Toc30471805)

[Постановка задачи. 2](#_Toc30471806)

[Алгоритм метода и условия его применимости 3](#_Toc30471807)

[Предварительный анализ задачи и проверка условий применимости 5](#_Toc30471808)

[Тестовый пример 5](#_Toc30471809)

[Контрольные тесты 7](#_Toc30471810)

[Модульная структура программы 7](#_Toc30471811)

[Численный анализ решения задачи 7](#_Toc30471812)

[Вывод 9](#_Toc30471813)

[Приложение 11](#_Toc30471814)

# Формулировка задачи

Решить алгебраическую проблему собственных значений итерационным методом вращения Якоби. Исследовать работу метода Якоби при разной требуемой точности и приближениях.

# 

# Постановка задачи.

Дана симметричная матрица А размерности N.  
Задача: найти собственные числа и собственные векторы матрицы, преобразовав ее к диагональному виду методом вращения Якоби.   
H(i) – i-тая матрица вращения, каждый столбец которой соответствует собственному вектору числа .  
Следовательно, получится диагональная матрица   
A’ = (H(k))T \*…\* (H(0))T\*A\*H(0) \*…\*H(k) .  
 – матрица с.ч.   
-матрица с.в. K-тый столбец соответствует k-тому с.ч.

## 

# Алгоритм метода и условия его применимости

**Определение.** Ненулевой вектор http://mathprofi.ru/k/sobstvennye_znachenija_i_sobstvennye_vektory_clip_image018_0000.gif, который при умножении на некоторую квадратную матрицу http://mathprofi.ru/k/sobstvennye_znachenija_i_sobstvennye_vektory_clip_image016_0000.gif превращается в самого же себя с числовым коэффициентом http://mathprofi.ru/k/sobstvennye_znachenija_i_sobstvennye_vektory_clip_image025.gif, называется **собственным вектором** матрицы http://mathprofi.ru/k/sobstvennye_znachenija_i_sobstvennye_vektory_clip_image016_0001.gif. Число http://mathprofi.ru/k/sobstvennye_znachenija_i_sobstvennye_vektory_clip_image025_0000.gif называют **собственным значением** или **собственным числом** данной матрицы.

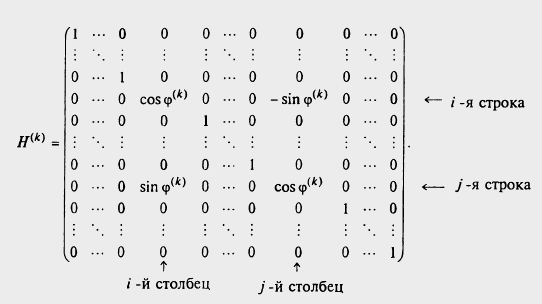
Метод вращения Якоби применяется только для симметричных матриц. Метод основывается на использовании ортогональных матриц.

Итерационная формула, для которой А(0) = А: А(k+1) = (H(k))-1\*A(k)\*H(k) = (H(k))T\*A(k)\*H(k) k = 0, 1, 2,…; где H – ортогональная матрица плоских вращений (на k-ом шаге обнуляется максимальный по модулю элемент матрицы H(k) предыдущего шага(а значит и симметричный ему элемент)). Это определяет способ фиксирования пары индексов i и j и угол поворота ϕ.

На каждом шаге таких преобразований пересчитываются только 2 строчки матрицы предыдущего шага.

Но полученный на некотором этапе преобразований нулевые элементы на следующем этапе станут ненулевыми, но предельное поведение всё равно будет: A(k)->D при k->∞, где D = diag(**λ**i) I = 1,2,…,n.

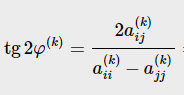
На k-й итерации для элемента а(k)ij, ij, определяется ортогональная матрица H(k), приводящая этот элемент а(k+1)ij к нулю. При этом на каждом шаге итерации в качестве а(k+1)ij выбирается наибольший по модулю.

Матрица H(k) зависит от угла ϕ(k) и имеет вид:

В данной ортогональной матрице элементы на главной диагонали единичные кроме

hii(k)=cosϕ, hjj(k)=cosϕ, hij(k)=-sinϕ, hji(k)=sinϕ.

А угол ϕ вычисляется по формуле:

  
Обозначим эту дробь переменной x (k)

Алгоритм

1. Положить A(0) = A и задать ɛ.
2. Выделить в матрице А максимальный по модулю элемент |aij(k)| i<j ().  
   Если max(|aij(k)|) ɛ – завершить алгоритм.
3. Найти синус и косинус поворота

Примем за х =

а

1. Cоставить матрицу H(k).
2. Вычислить приближение A(k+1) = (H(k))T\*A(k)\*H(k).  
   При умножении будет меняться всего 2 столбца и 2 строки матрицы А.  
   Поэтому, можно обойтись всего несколькими операциями.   
   обозначим cos за c и sin за s.  
   А\_newii =c2Aii - 2scAij + s2AjjА\_newjj =c2Aii + 2scAij + s2Ajj

А\_newij  = А\_newji = (c2 - s2)Aij + sc(Aii - Ajj)

А\_newik = А\_newki =cAik – sAjk

А\_newjk = А\_newkj =cAik + sAjk

Остальные эл-ты остаются неизменными

1. Если |aij(k)| ɛ для всех ij, то процесс завершить.

Найти собственные значения **λ**i(A(k))=a(k)ii i = 1,2,…

Собственные векторы Xi находятся как i-ые столбцы матрицы, получившейся в результате перемножения:

V = H(0)\*H(1)\* H(2)\* H(3)\*… \*H(k-1)=(Х1, Х2,…,Хn).

Перейти к пункту 2.

Условия применимости

Матрица А должна быть симметрична: АТ = А.

# Предварительный анализ задачи и проверка условий применимости

Построение матрицы:  
Возьмем матрицу D размера n с числами от 1 до n на главной диагонали. Они и будут точными собственными значениями матрицы. Затем построим на основе случайного вектора размера n ортогональную матрицу по формуле

H = .

Получим ортогональным преобразованием новую матрицу А, собственные числа которой будут такими же, что и у матрицы D.

Матрица будет симметричной.

# Тестовый пример

A(0) = A, ɛ = 0.001.

a(0)13= 2 > ɛ.

X = 2\*2/(5-3) = 2

=а0.85065

= 0.52537

Матрица вращения H(0)=

A(1)= (H(0))T\* A(0)\* H(0)A(1)=

a(1)12=1.376>ɛ

x = 2\*1.376/(-) = 1.2307

= 0.90294

= 0.42975

Матрица вращения H(1)=

A(2)= (H(1))T\* A(1)\* H(1)A(2)=

a(1)13=>ɛ

x = 2\*1./(-) = 0.54476

= 0.984253

= 0.176766

Матрица вращения H(2)=

A(3)= (H(2))T\* A(2)\* H(2)

A(3)=

# Контрольные тесты

Для исследования метода были рассмотрены симметрические матрицы размерности 10. Для каждой матрицы были вычислены собственные значения с допустимыми погрешностями от 0.1 до (0.1)^6.  
  
В первом и втором пункте исследования (пункты определяются графиками) матрица никак не изменялась. Требуемая точность менялась на порядок для каждого теста.  
  
В третьем и четвертом пунктах в одно из собственных значений вносилась погрешность равная 10^-i, (где i – номер теста). По новым с. ч. строилась матрица, для которой впоследствии будем искать собственные значения методом Якоби. То есть, вносим погрешность и смотрим, как от этого изменятся погрешность вычислений (данные, полученные с помощью с.ч. без погрешности – данные, полученные с помощью с.ч. с погрешностью) и итерации.

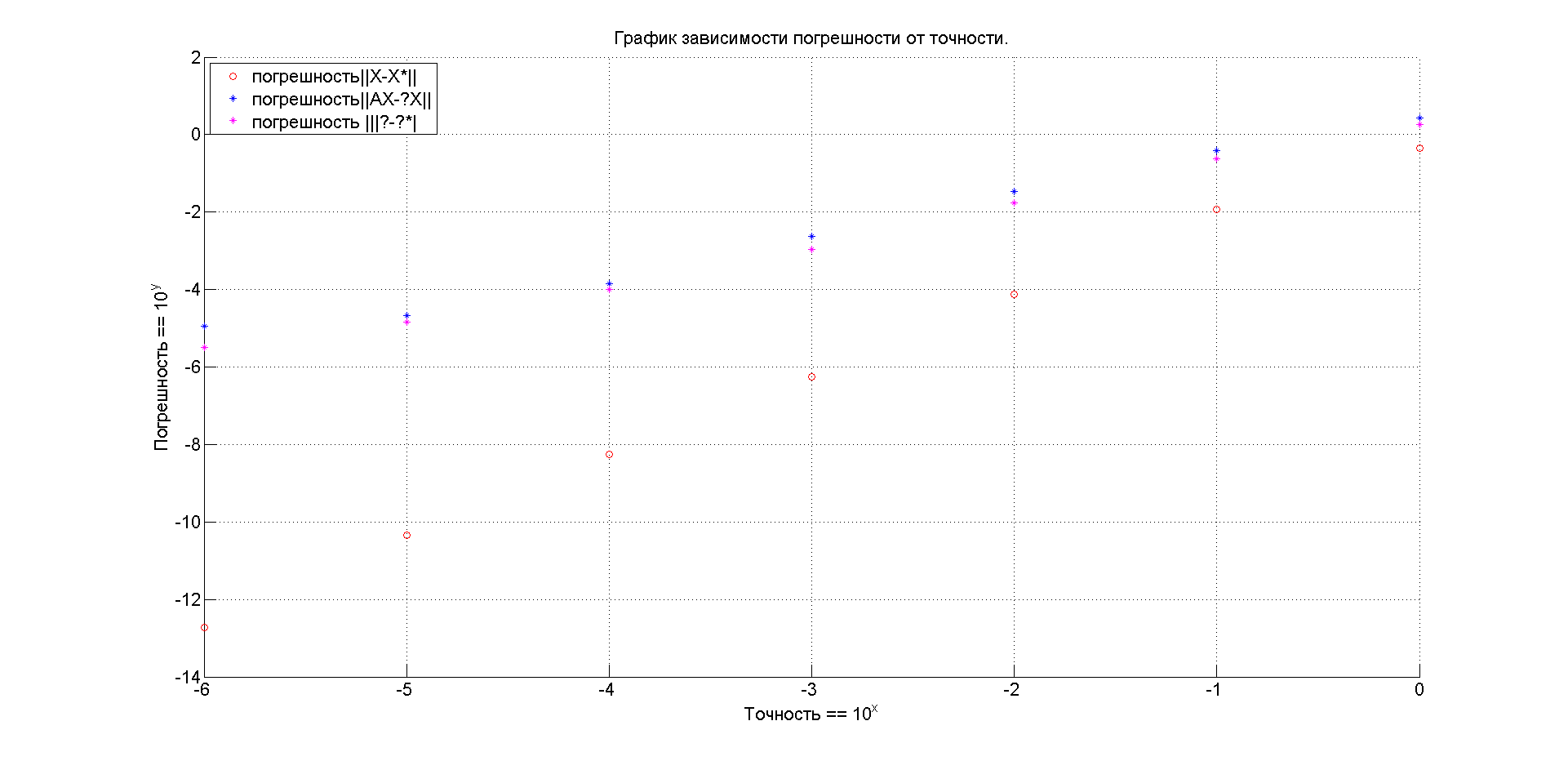
В пятом пункте рассмотрим влияние отделимости с.ч. на итерации.

# Модульная структура программы

Программа состоит из 2 модулей:

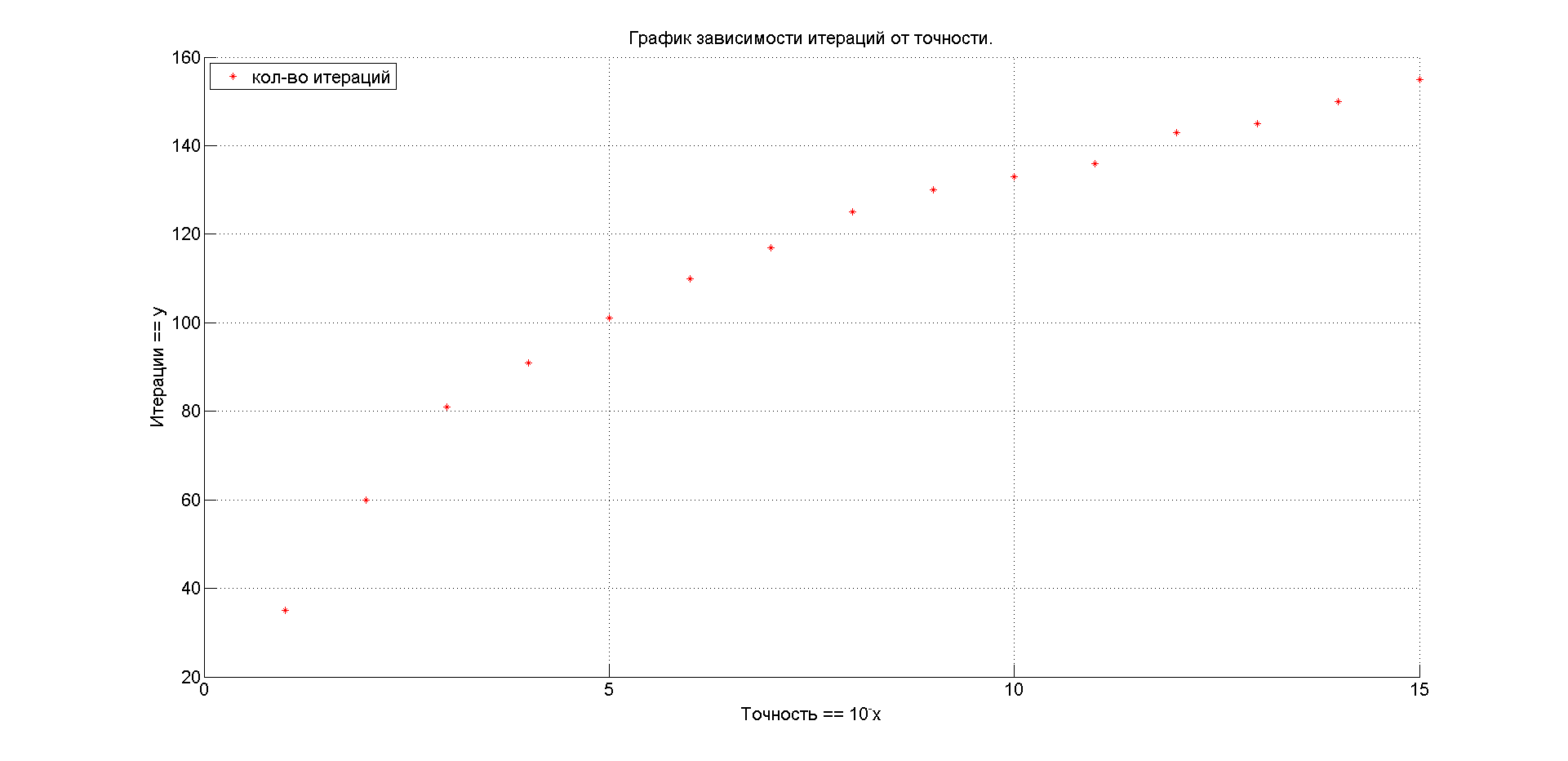
1. void method\_Jakobi(int n, double eps) – выполняет алгоритм Якоби, пока не будет выполнено условие необходимой точности. Условие проверяется в этой же функции в момент поиска максимального эл-та матрицы. Для упрощения чтения кода указатель на матрицу записан в глобальной переменной. Передавать его как входной параметр не нужно.
2. int main(void)-читает из файла размер матрицы, матрицу, требуемый порядок точности, считает собственные числа с помощью метода Якоби и кол-во итераций, а затем возвращает их в файл, который после считает матлаб.  
   Параметр точности записывается в глобальную переменную.

# Численный анализ решения задачи

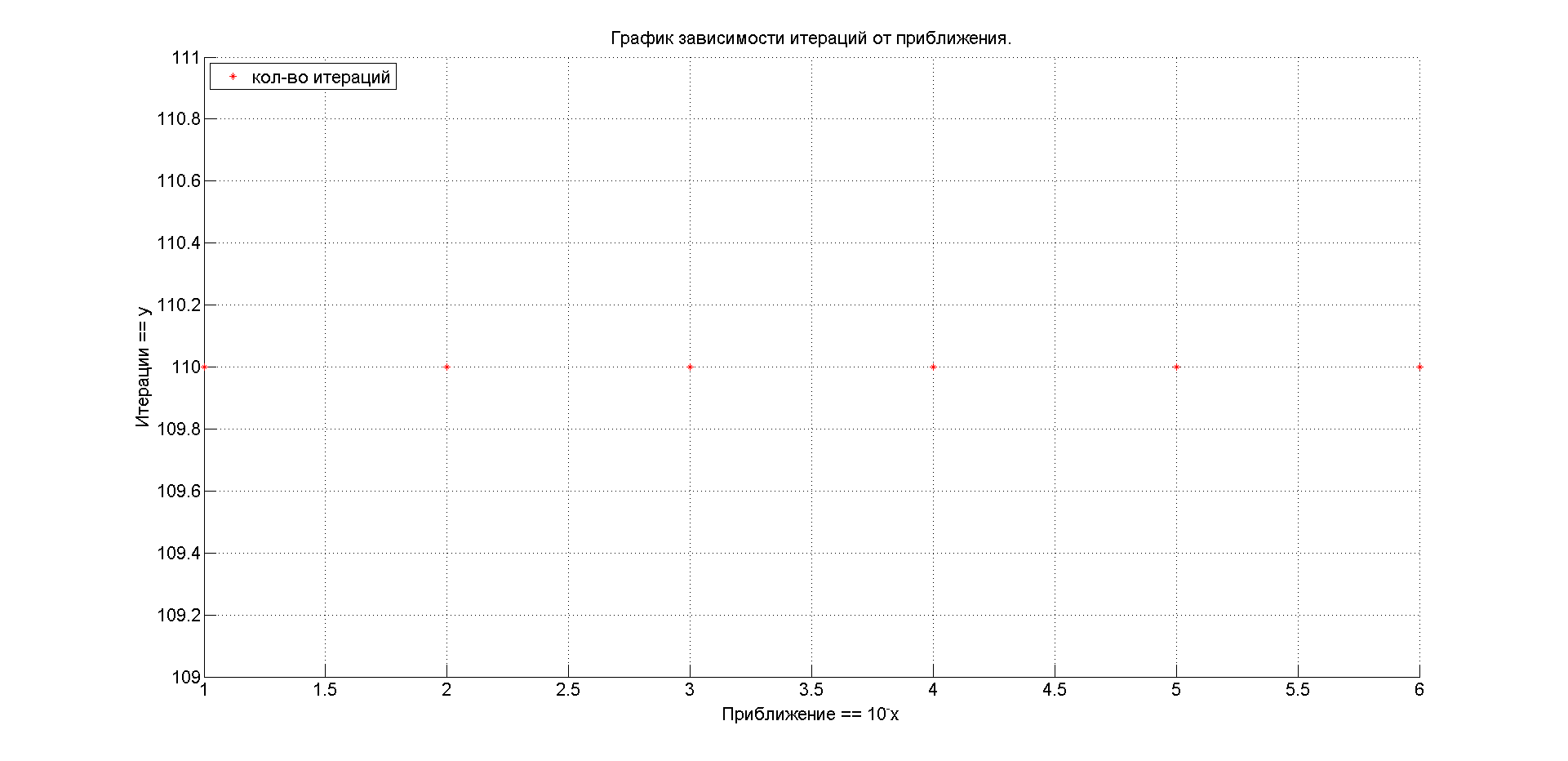


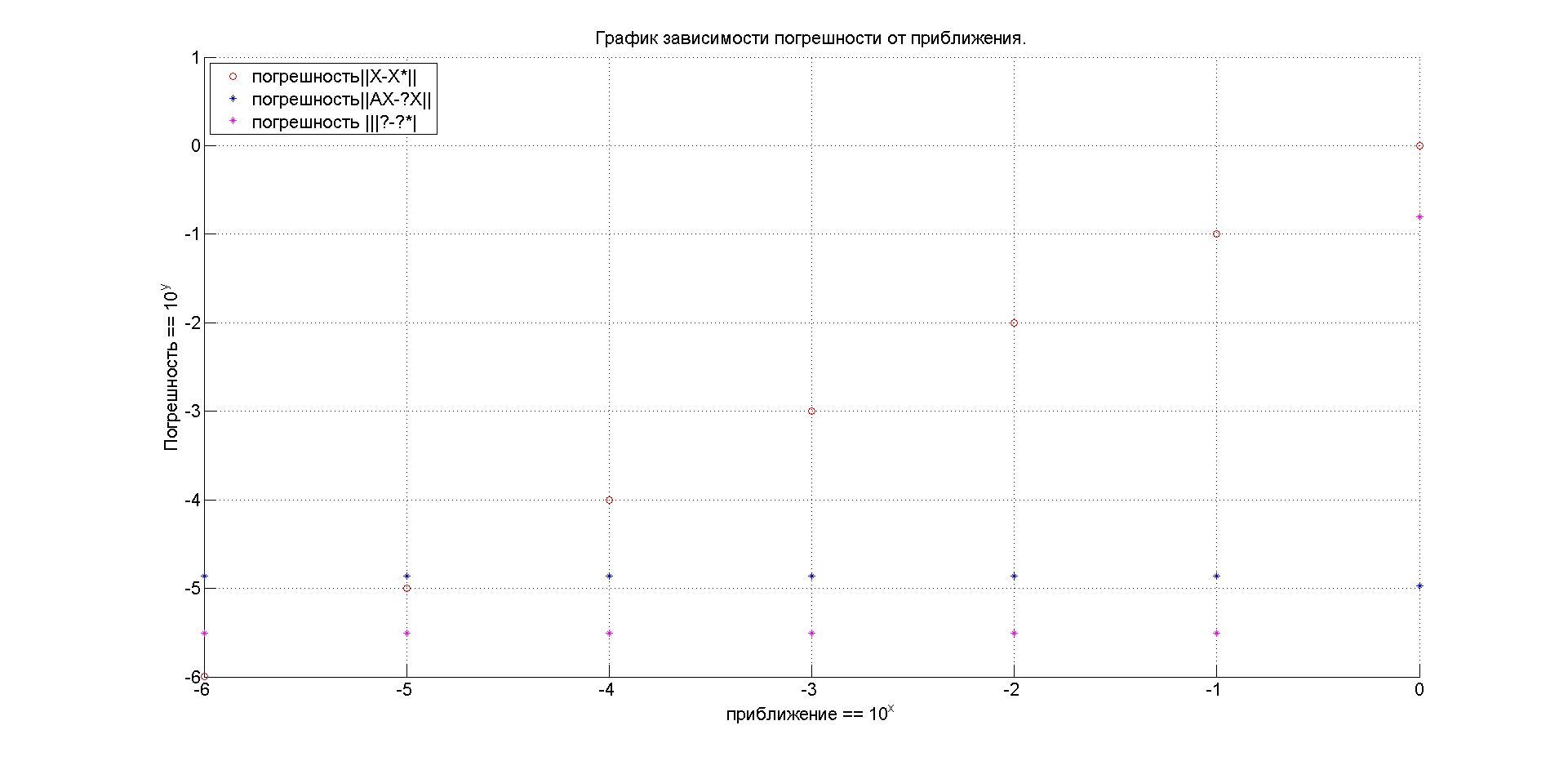
Примечание: “?” – это λ

Стоить заметить, что порядок погрешности вычисленных собст. векторов превосходит в 2 раза требуемую точность.   
Скорость сходимости является квадратичной.  
Вычисления проведены до максимально возможной точности.  
(В приложении представлен график вычисления до 10^-14)



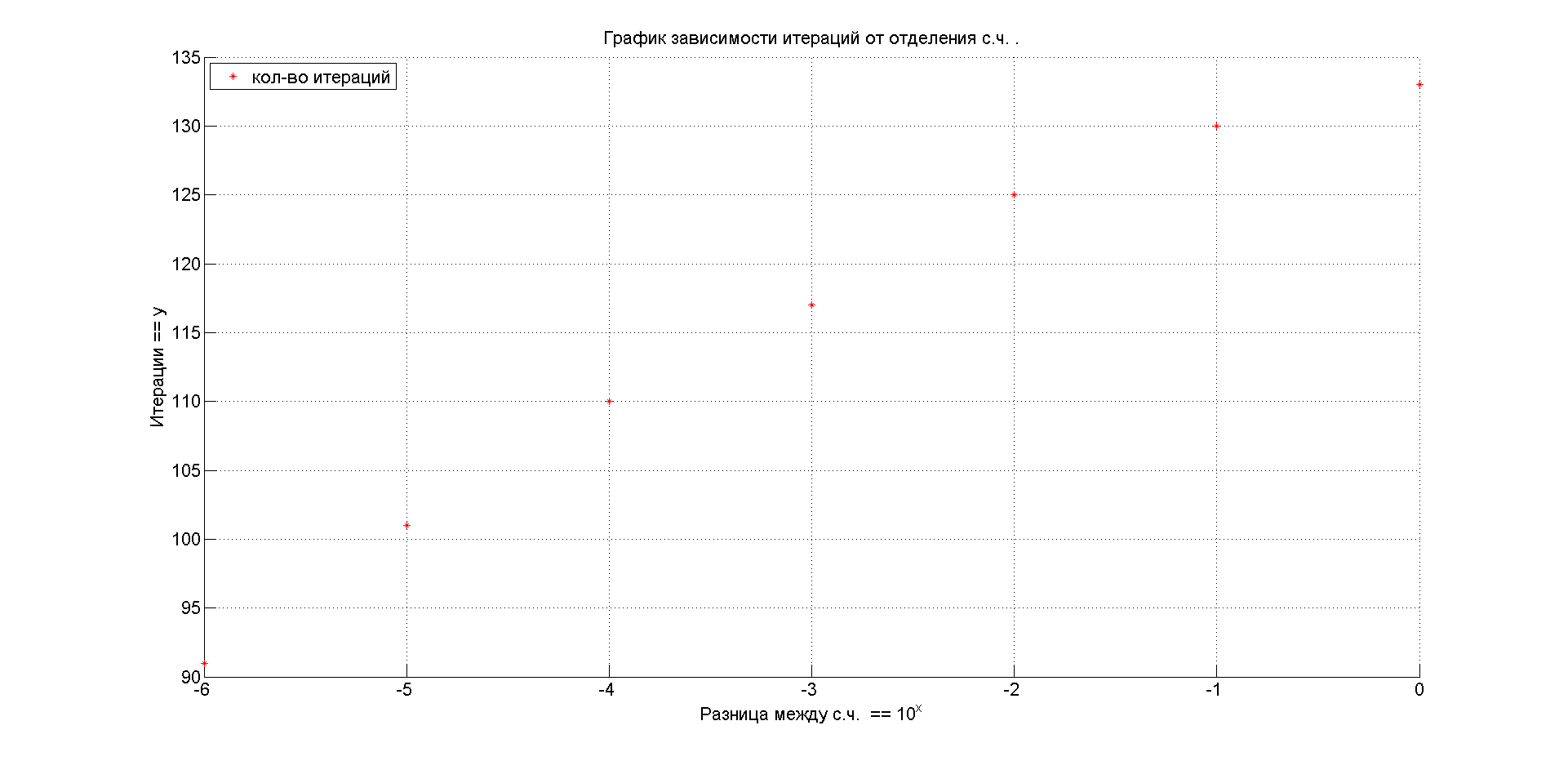
Логарифмическая зависимость кол-ва итераций от порядка точности.

Погрешность AX –  λX остается константной.  
Погрешность  λ имеет экспоненциальную зависимость.  
  




Кол-во итераций остается константным при изменении собственного числа меньше чем на 0.1.

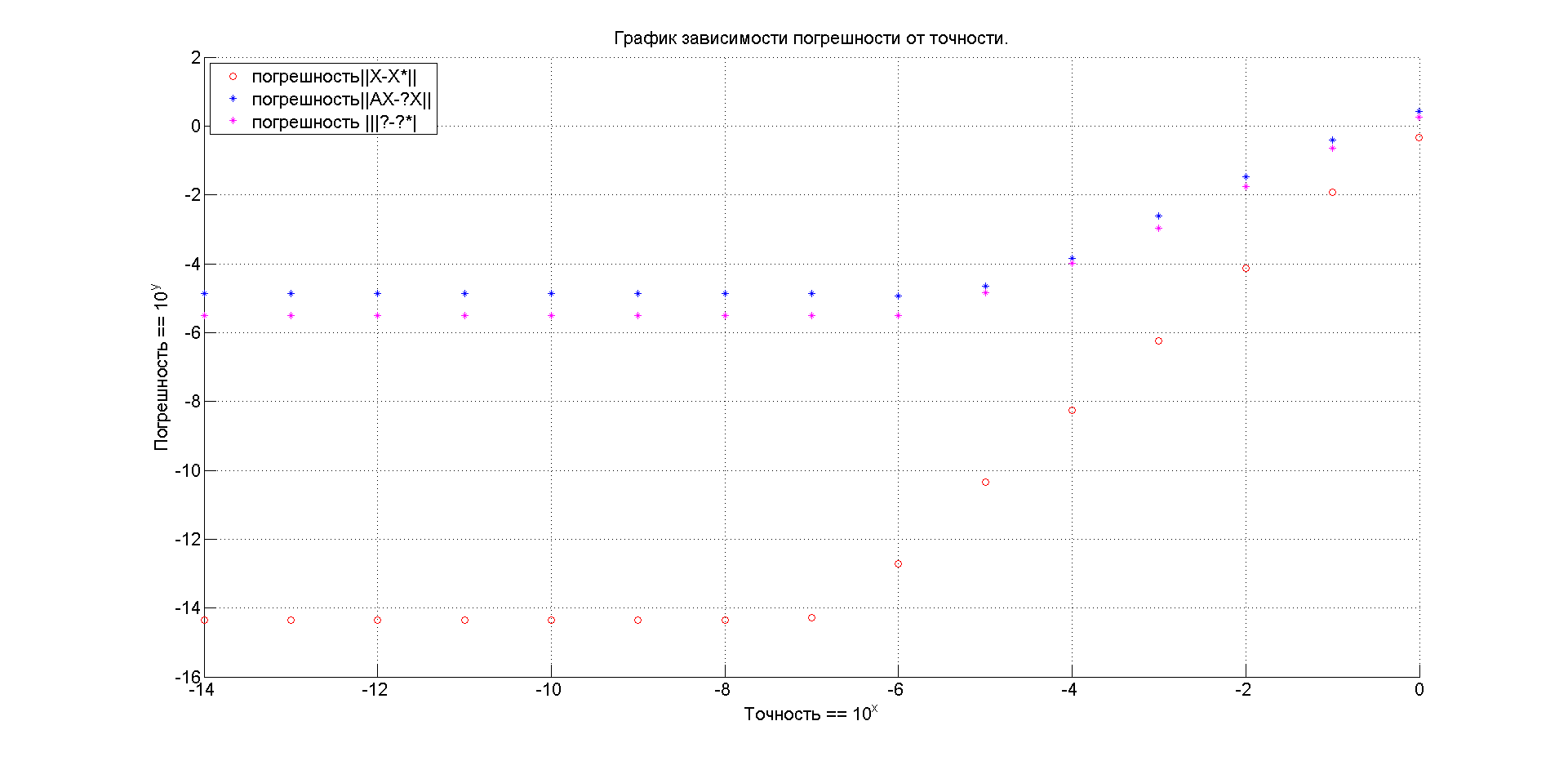
Чем ближе друг к другу с.ч., тем меньше итераций требуется.



# Вывод

Видим, метод Якоби нахождения собственных чисел и соответствующих им собственных векторов решает полную проблемы собственных значений и собственных векторов для задачи малой размерности.  
Для матриц с плохой отделимостью собственных чисел метод вращения Якоби находит собственные числа и собственные векторы быстрее, чем для матриц с хорошей отделимостью.  
Слабо меняется от порядка приближения к собств. числам.

Приложение



дополнительный график