БГУИР

Кафедра информатики

**ОТЧЕТ**

по лабораторной работе №2

**Решение систем линейных уравнений методом простой итерации и методом Зейделя**

Выполнил: Проверил:

студент гр.453504 Ероминек К.Р.

Антоненко И.Д.

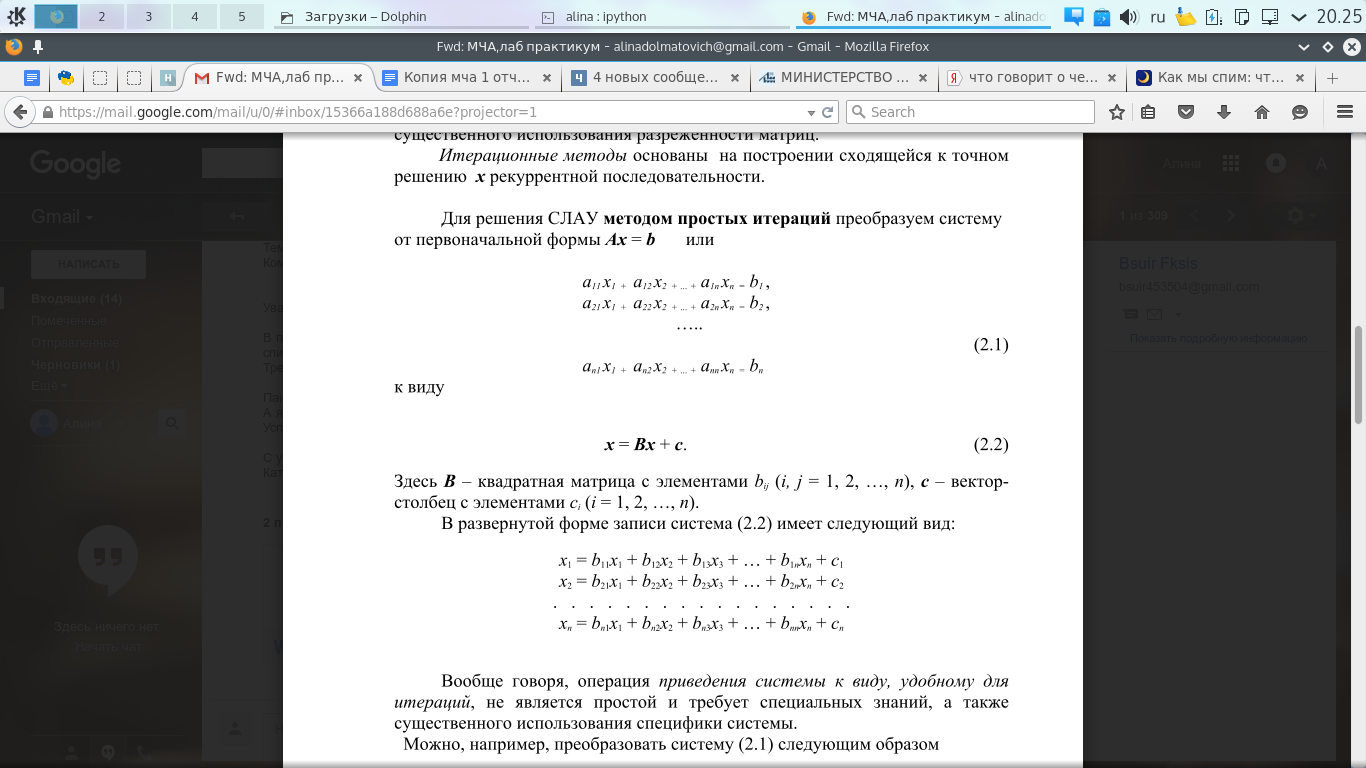
Минск, 2016

**Введение**

Прямые методы применяют главным образом для решения задач малой размерности, когда нет ограничений в доступной оперативной памяти ЭВМ или необходимости выполнения чрезмерно большого числа арифметических операций. Большие системы уравнений, возникающие в основном в приложениях, как правило, являются разреженными. Методы исключения для систем с разреженными матрицами неудобны, например, тем, что при их использовании большое число нулевых элементов превращается в ненулевые, и матрица теряет свойство разреженности. В противоположность им при использовании итерационных методов в ходе итерационного процесса матрица не меняется, и она, естественно, остается разреженной. Большая эффективность итерационных методов по сравнению с прямыми методами тесно связанна с возможностью существенного использования разреженности матриц.

Итерационные методы основаны на построении сходящейся к точном решению x рекуррентной последовательности.

Для решения СЛАУ **методом простых итераций** преобразуем систему от первоначальной формы Ax = b или

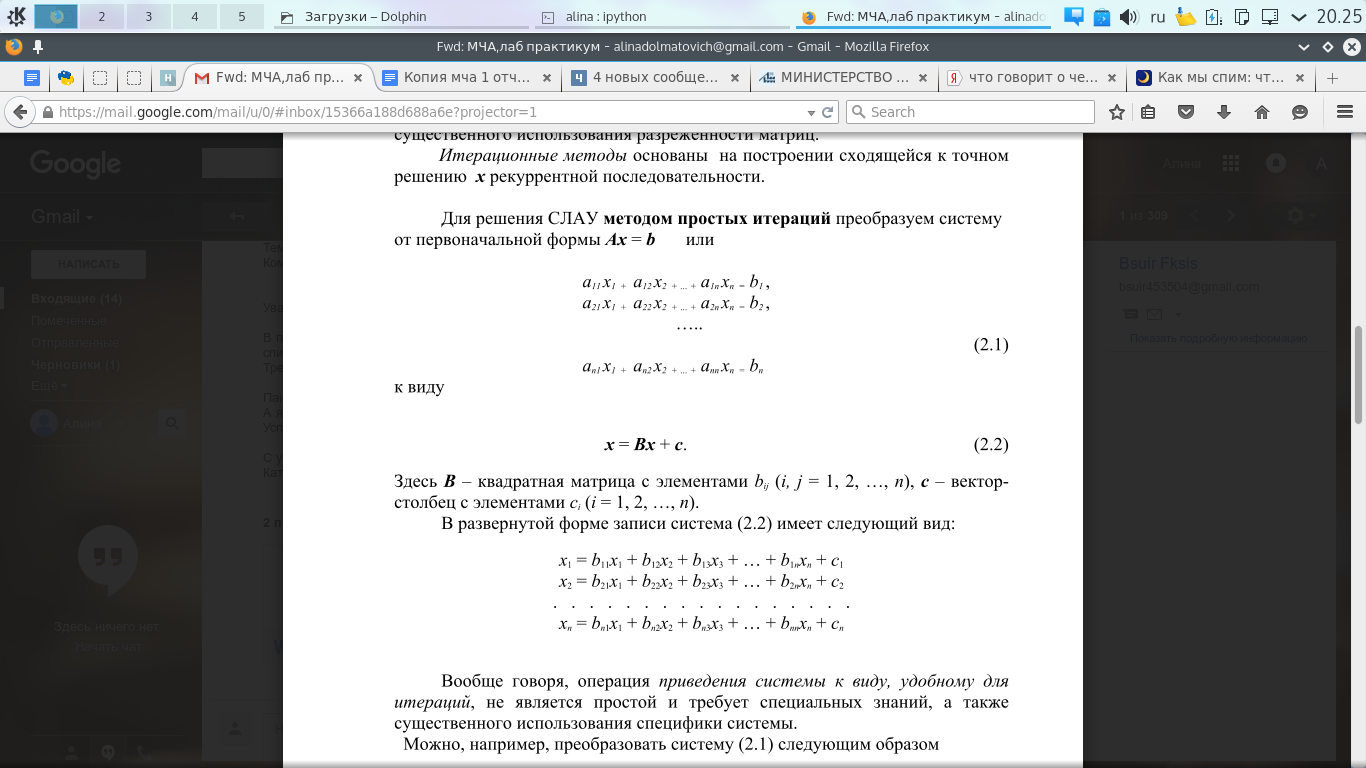


к виду

x = Bx + c.

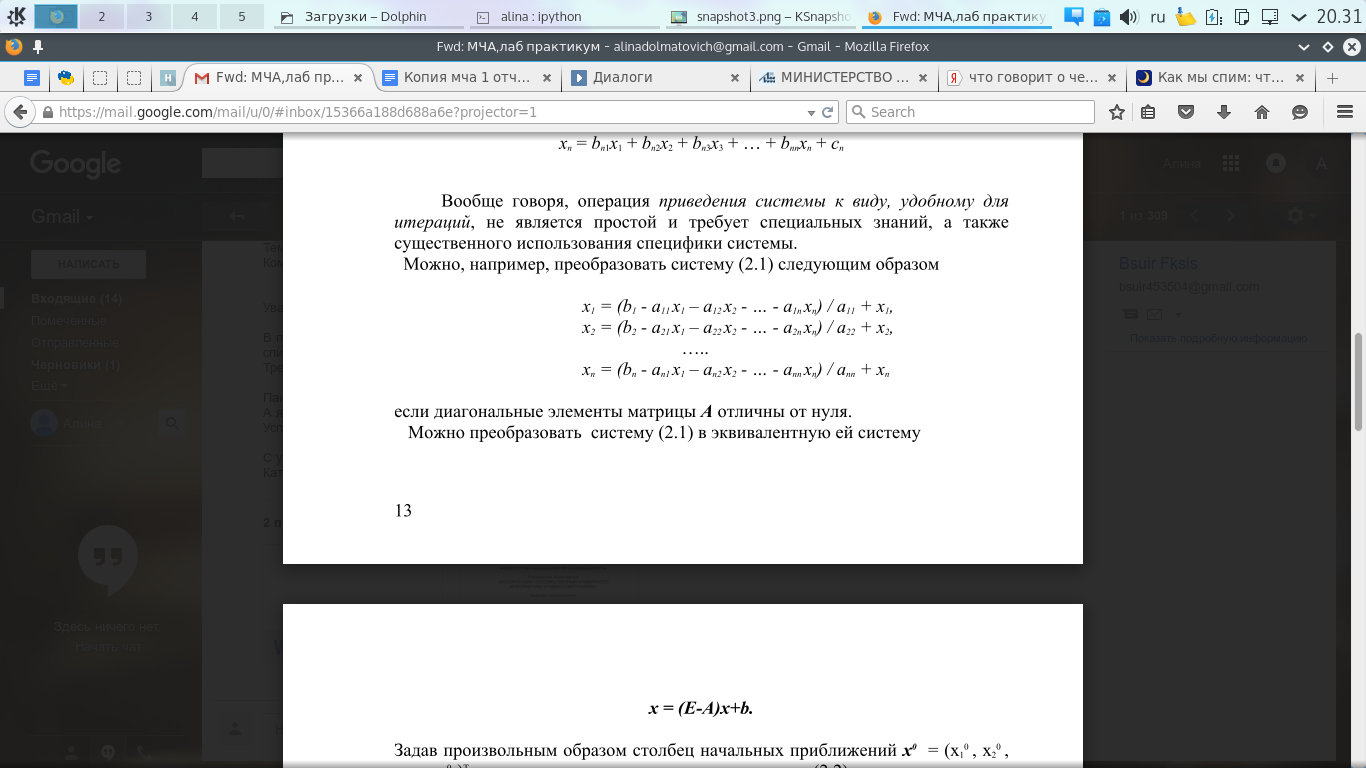
Здесь B – квадратная матрица с элементами bij (i, j = 1, 2, …, n), c – вектор-столбец с элементами ci (i = 1, 2, …, n).

В развернутой форме записи система (2.2) имеет следующий вид:



Вообще говоря, операция приведения системы к виду, удобному для итераций, не является простой и требует специальных знаний, а также существенного использования специфики системы.

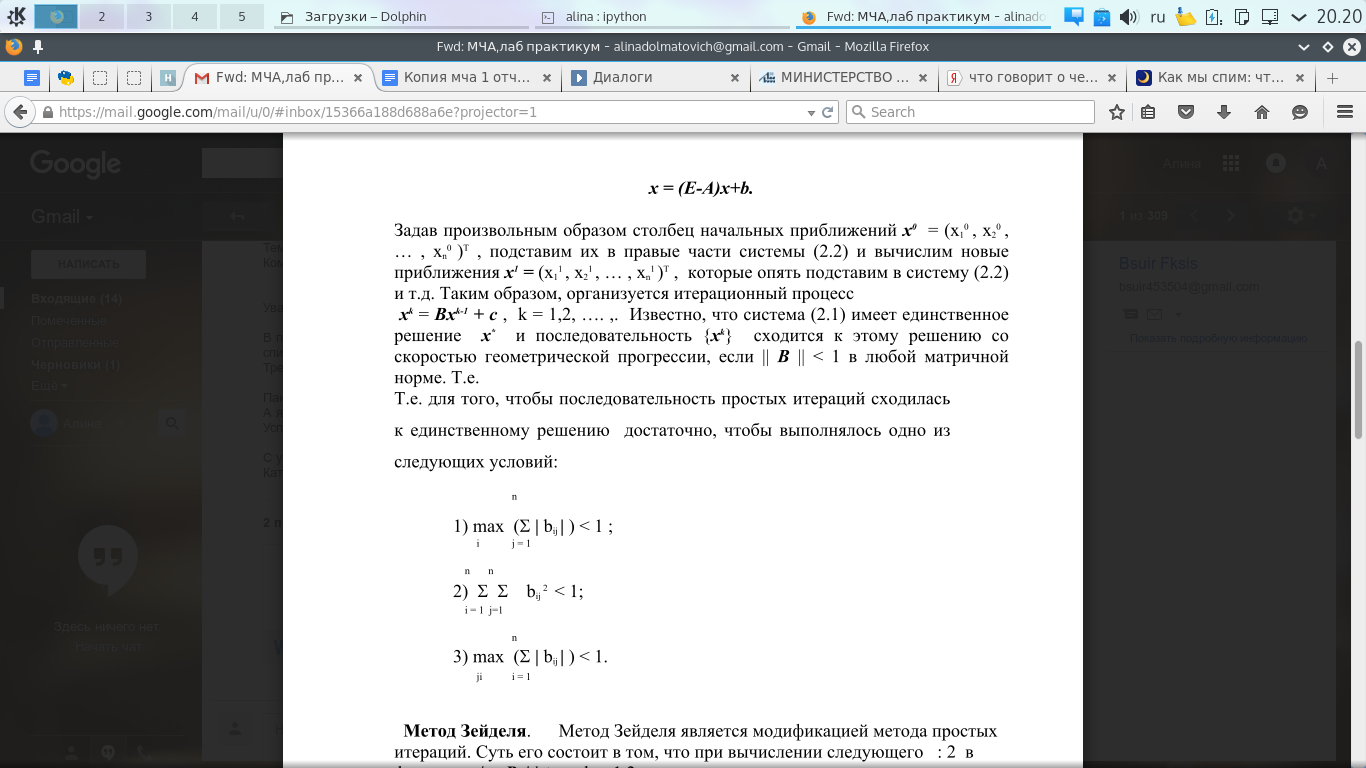
Можно, например, преобразовать систему (2.1) следующим образом



если диагональные элементы матрицы А отличны от нуля.

Можно преобразовать систему в эквивалентную ей систему, задав произвольным образом столбец начальных приближений х0 = (x10 , x20 , … , xn0 )T , подставим их в правые части сисемы (2.2) и вычислим новые приближения x1 = (x11 , x21 , … , xn1 )T , которые опять подставим в систему (2.2) и т.д. Таким образом, организуется итерационный процесс xk = Bxk-1 + c , k = 1,2, …. ,.

Известно, что система (2.1) имеет единственное решение х\* и последовательность {xk} сходится к этому решению со скоростью геометрической прогрессии, если || B || < 1 в любой матричной норме. Т.е. для того, чтобы последовательность простых итераций сходилась к единственному решению достаточно, чтобы выполнялось одно из следующих условий:



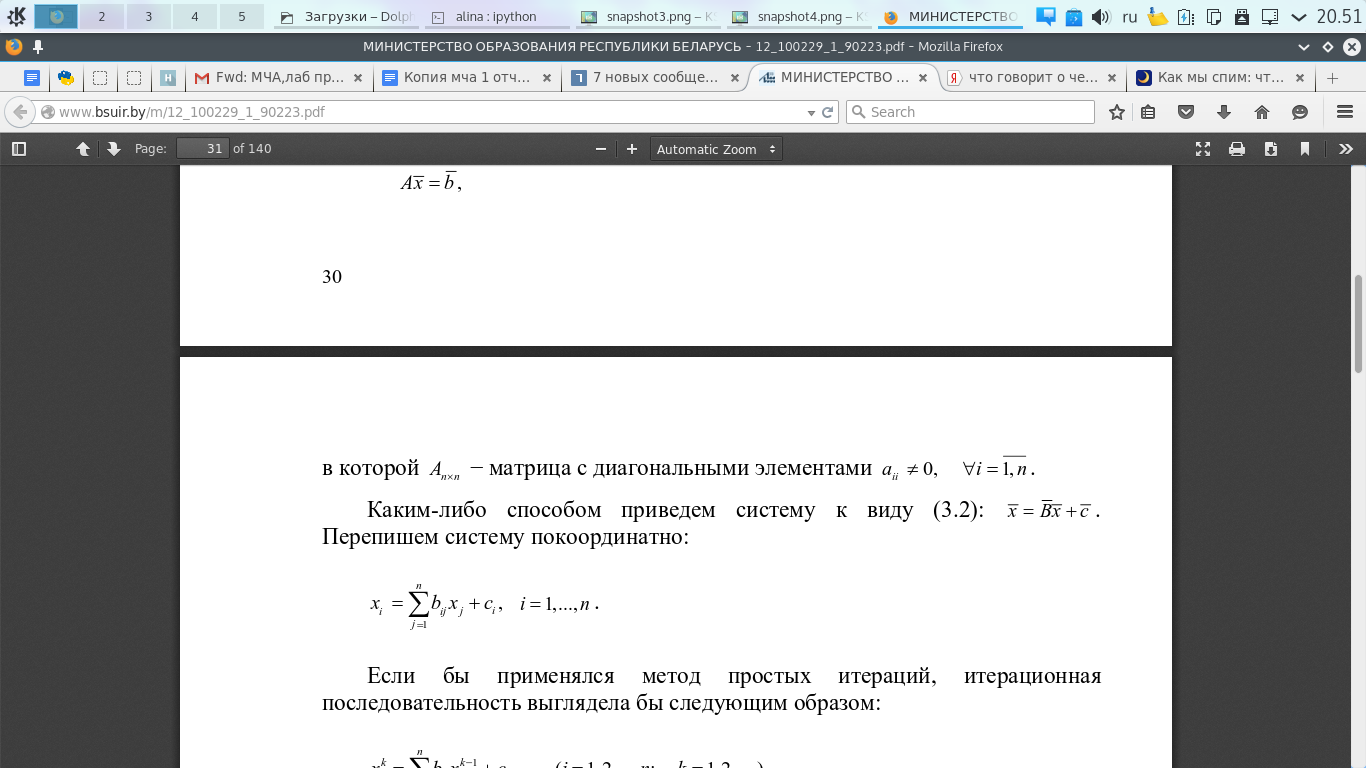
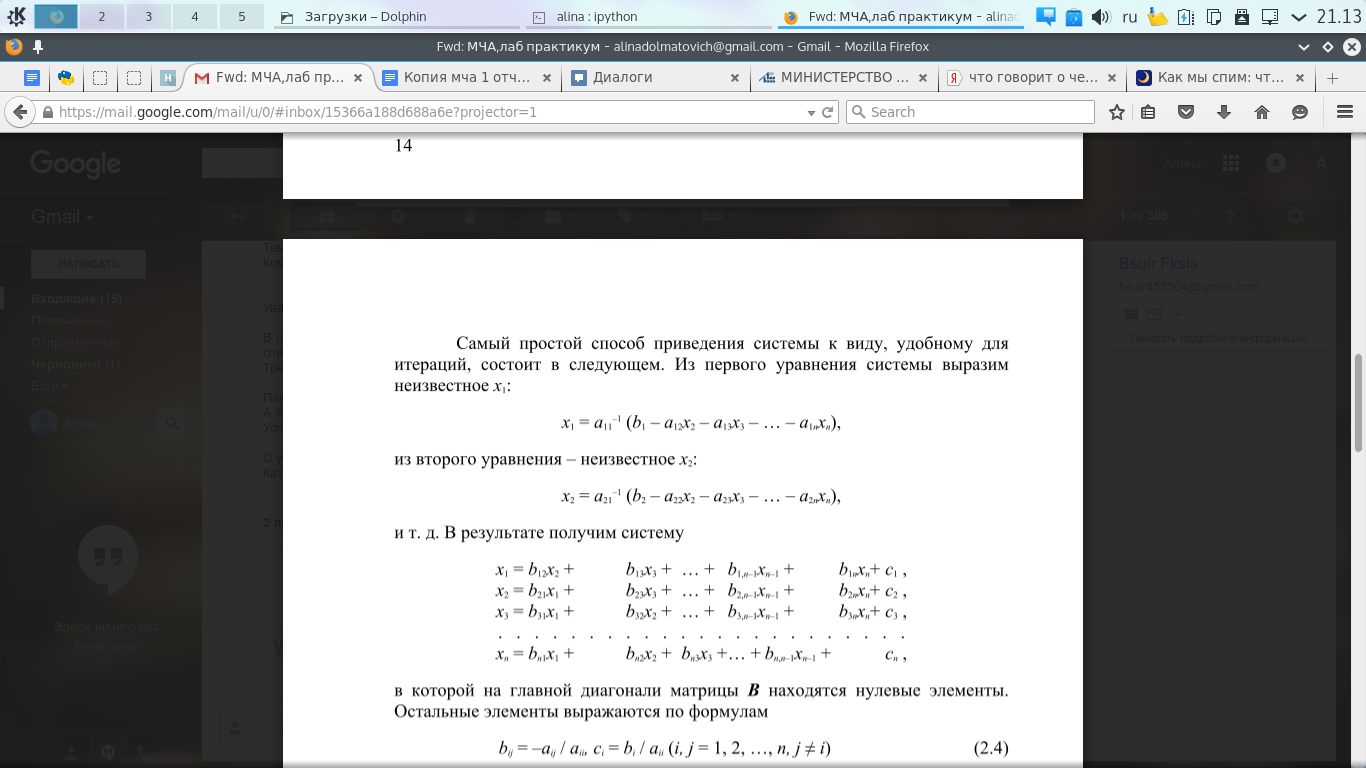
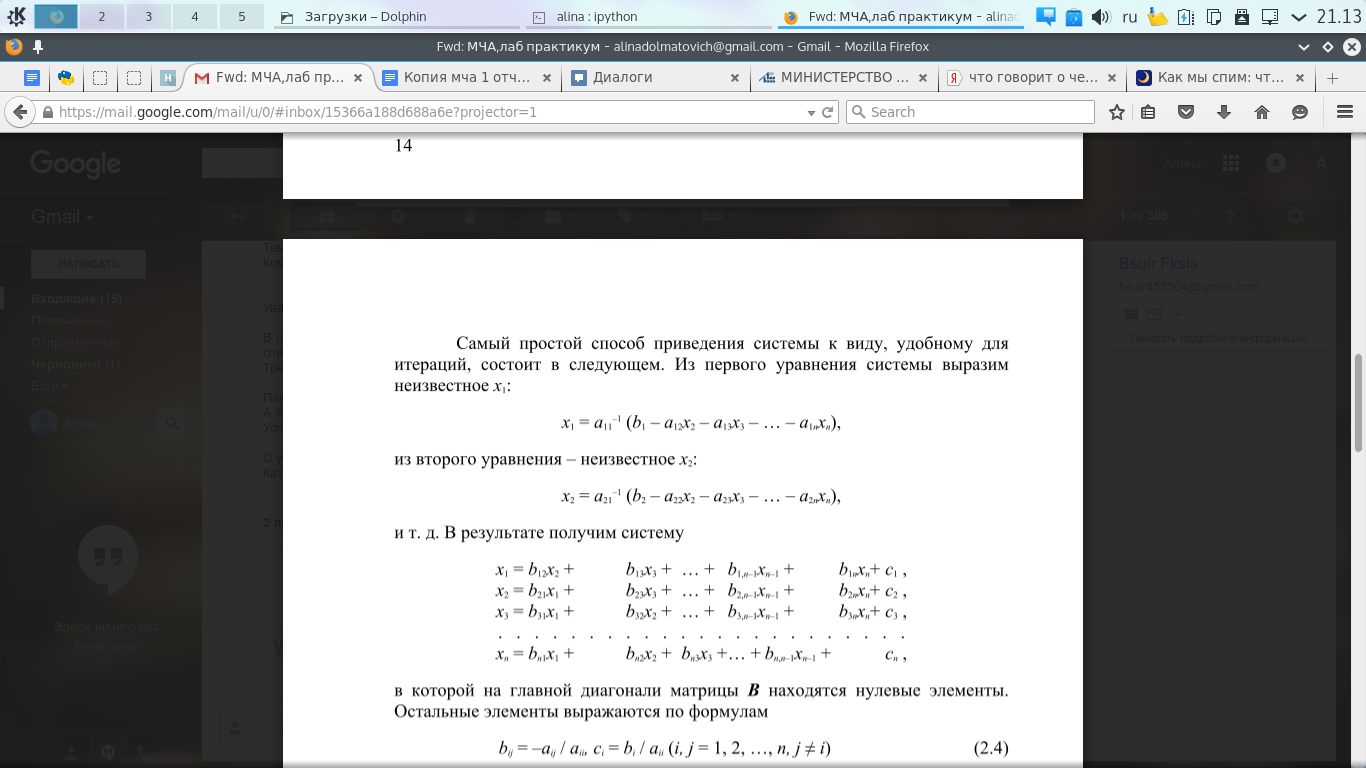
Рассмотрим модификацию метода итераций, называемую **методом Зейделя**. Пусть дана система линейных уравнений, Ax=b, в которой − матрица с диагональными элементами

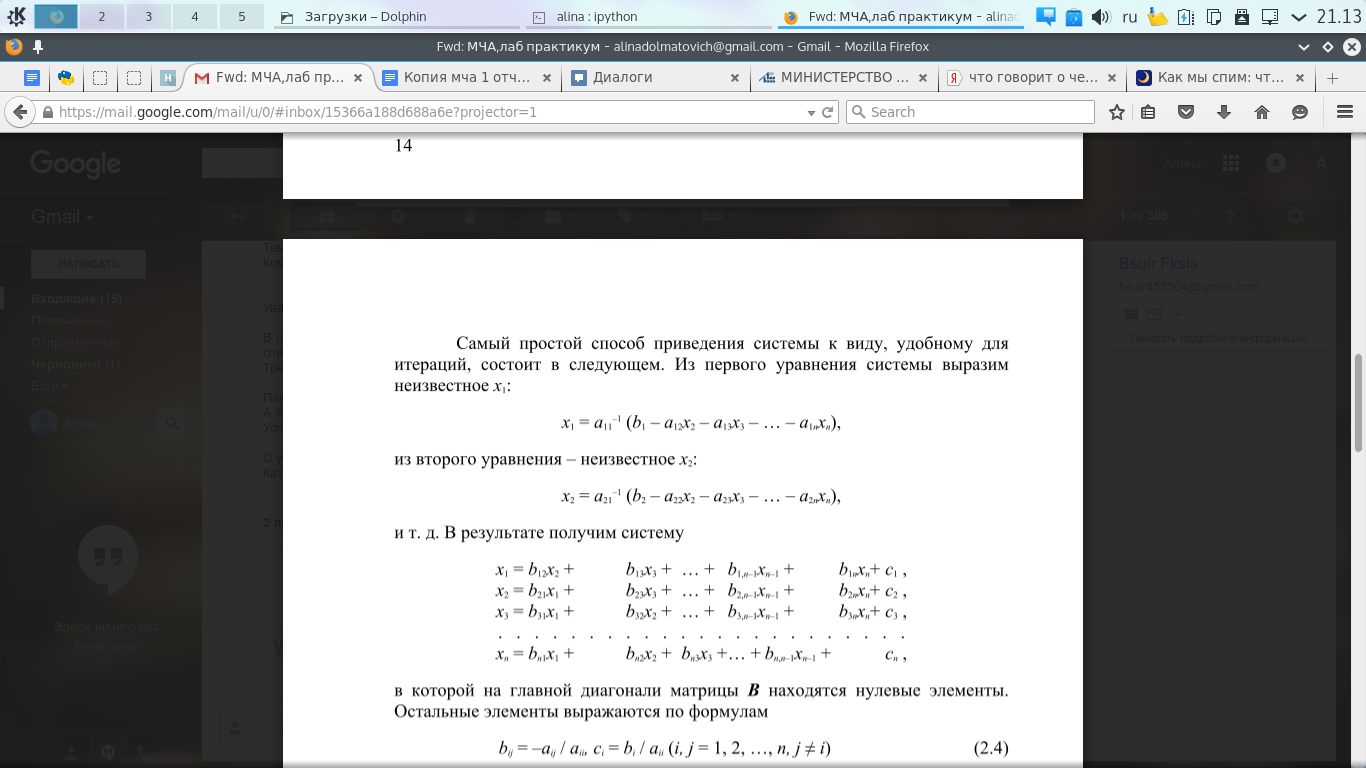
Схема алгоритма аналогична схеме метода простых итераций. Самый простой способ приведения системы к виду, удобному для итераций, состоит в следующем. Из первого уравнения системы выразим неизвестное x1:



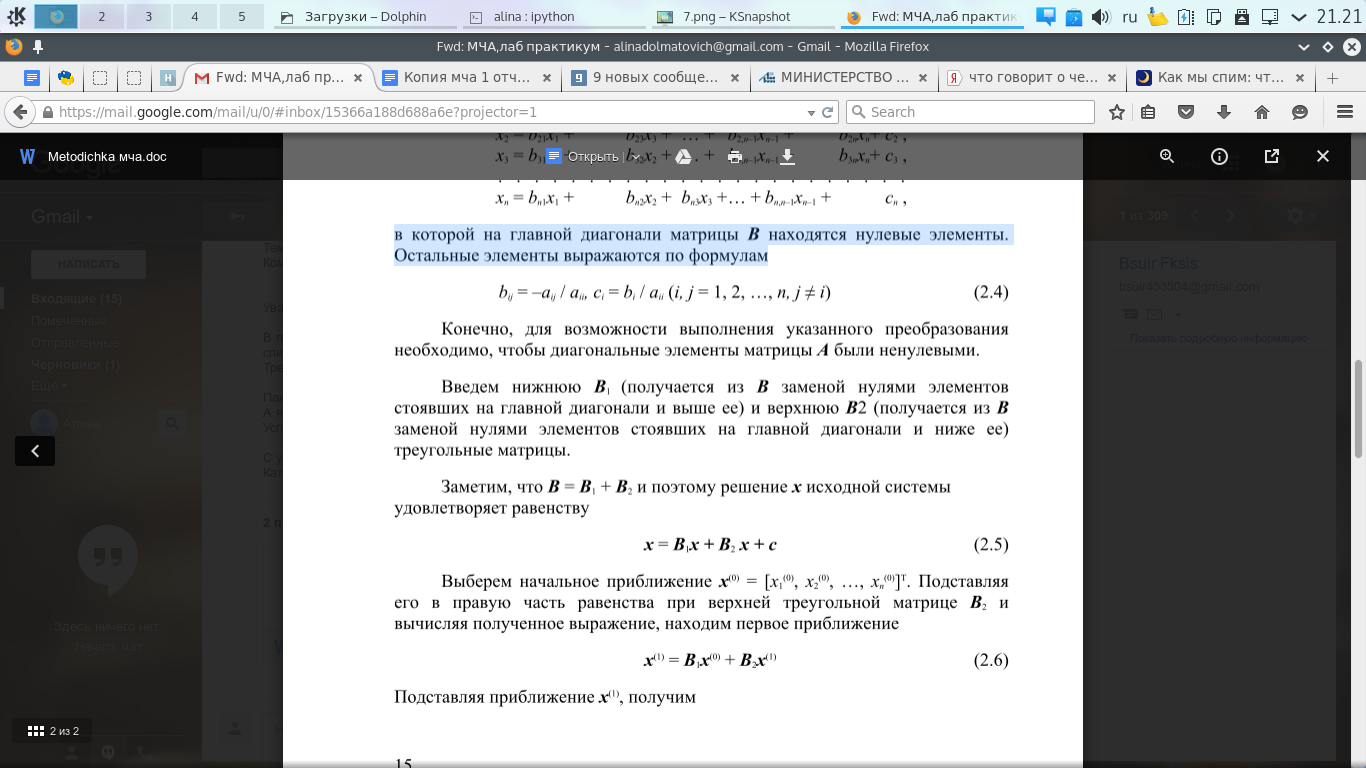
из второго уравнения – неизвестное x2:



и т. д. В результате получим систему



в которой на главной диагонали матрицы B находятся нулевые элементы. Остальные элементы выражаются по формулам



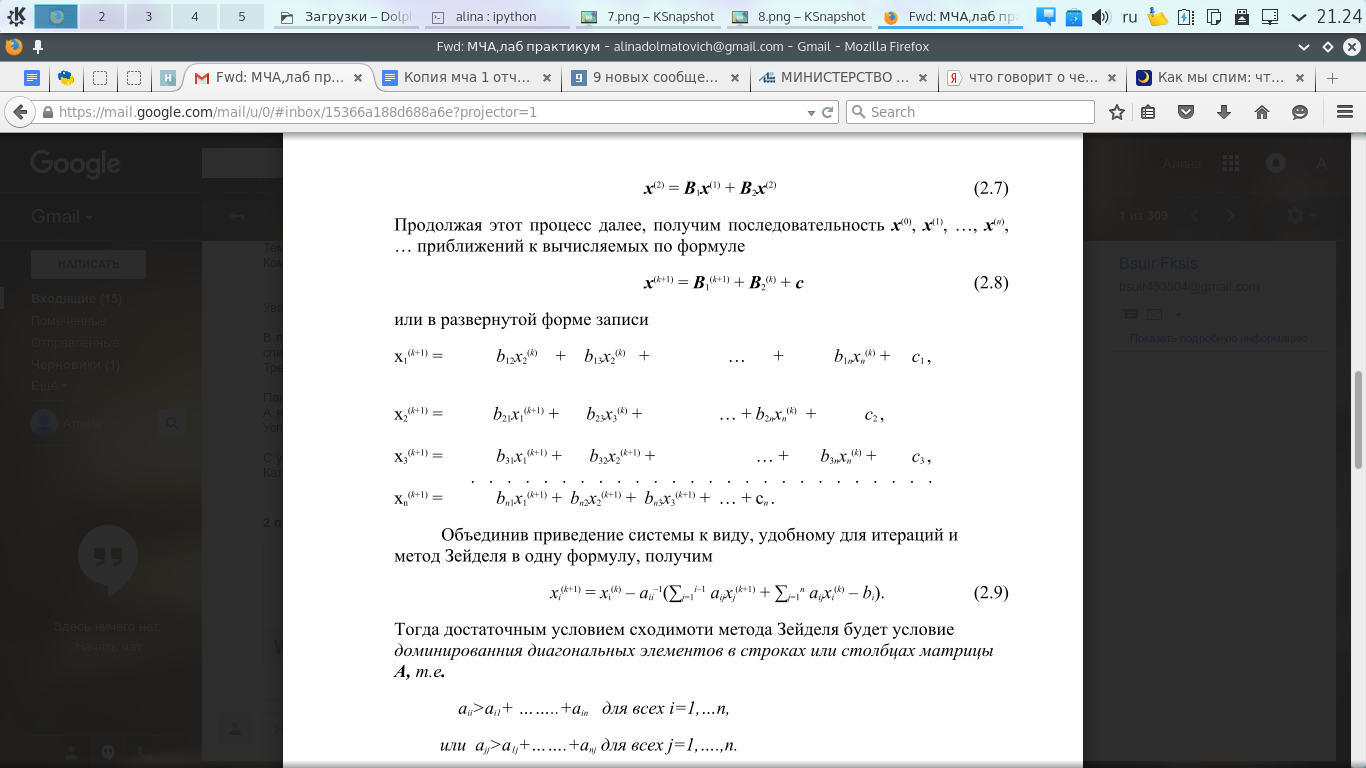
Конечно, для возможности выполнения указанного преобразования необходимо, чтобы диагональные элементы матрицы A были ненулевыми. Введем нижнюю B1 (получается из B заменой нулями элементов стоявших на главной диагонали и выше ее) и верхнюю B2 (получается из B заменой нулями элементов стоявших на главной диагонали и ниже ее) треугольные матрицы.

Заметим, что B = B1 + B2 и поэтому решение x исходной системы удовлетворяет равенству

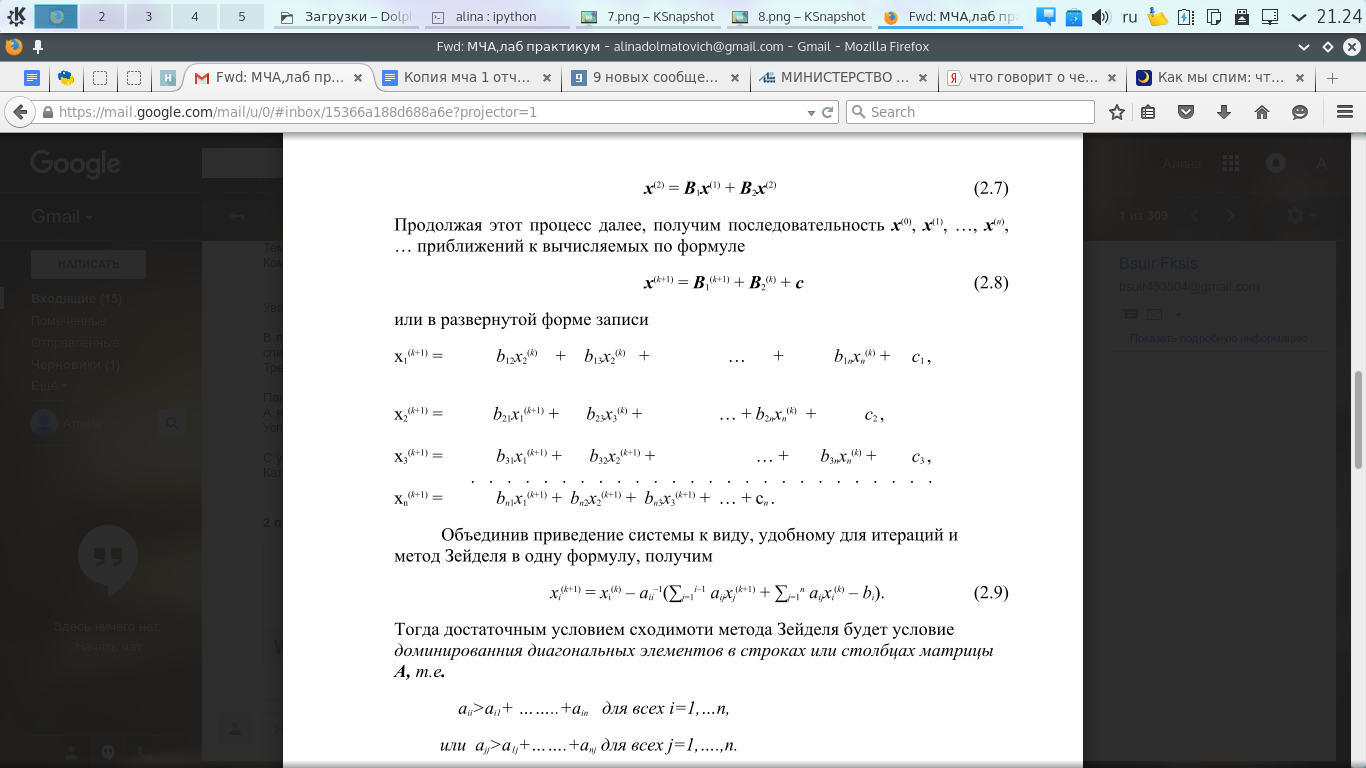
x = B1x + B2 x + c

Выберем начальное приближение x(0) = [x1(0), x2(0), …, xn(0)]T. Подставляя его в правую часть равенства при верхней треугольной матрице B2 и вычисляя полученное выражение, находим первое приближение x(1) = B1x(0) + B2x(1) (2.6)

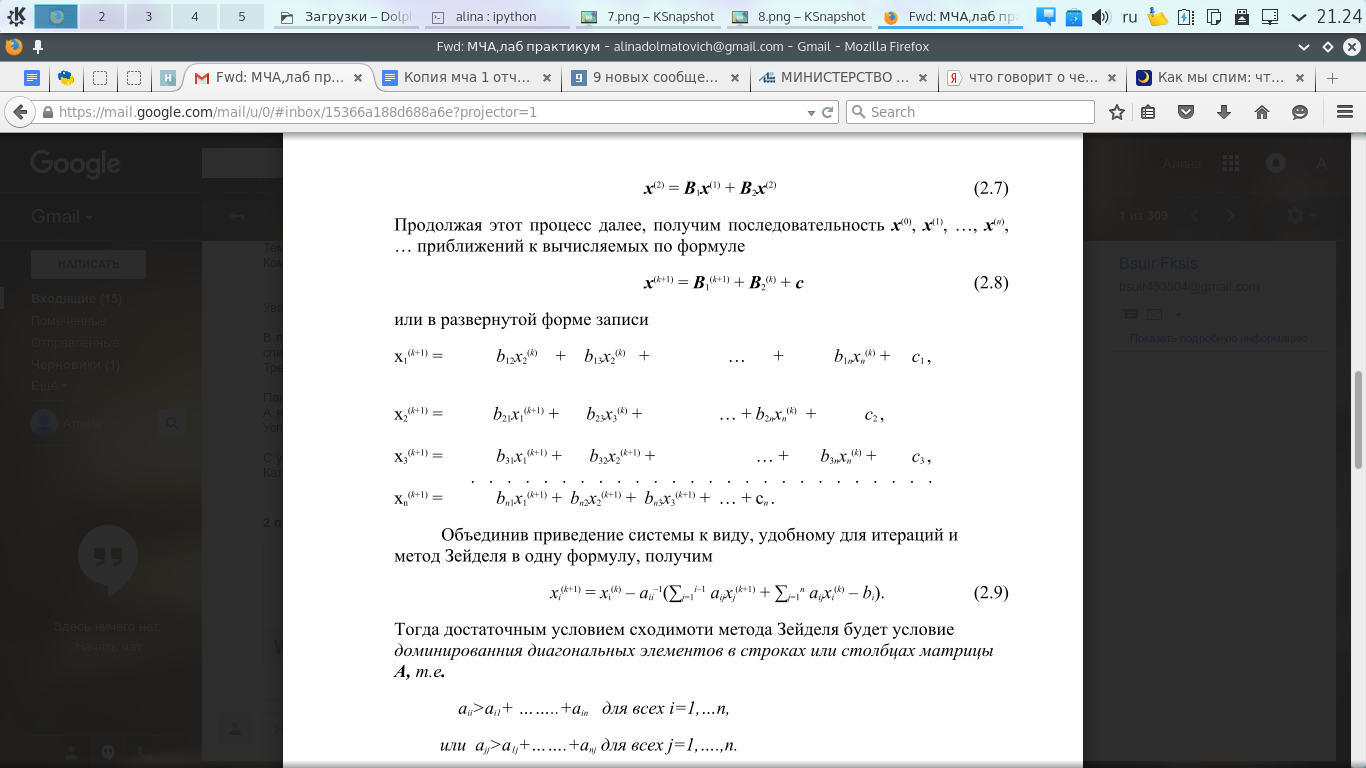
Подставляя приближение x(1), получим



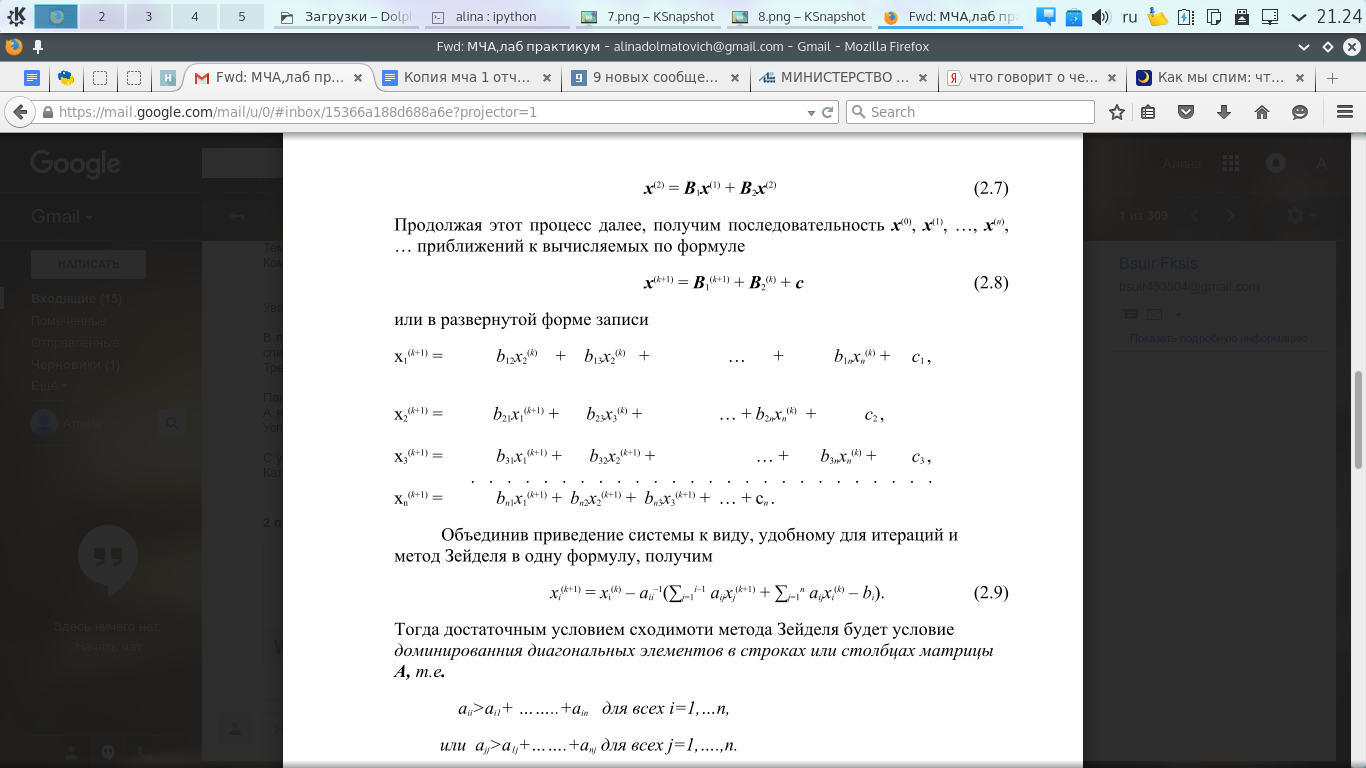
Продолжая этот процесс далее, получим последовательность x(0), x(1), …, x(n), … приближений к вычисляемых по формуле



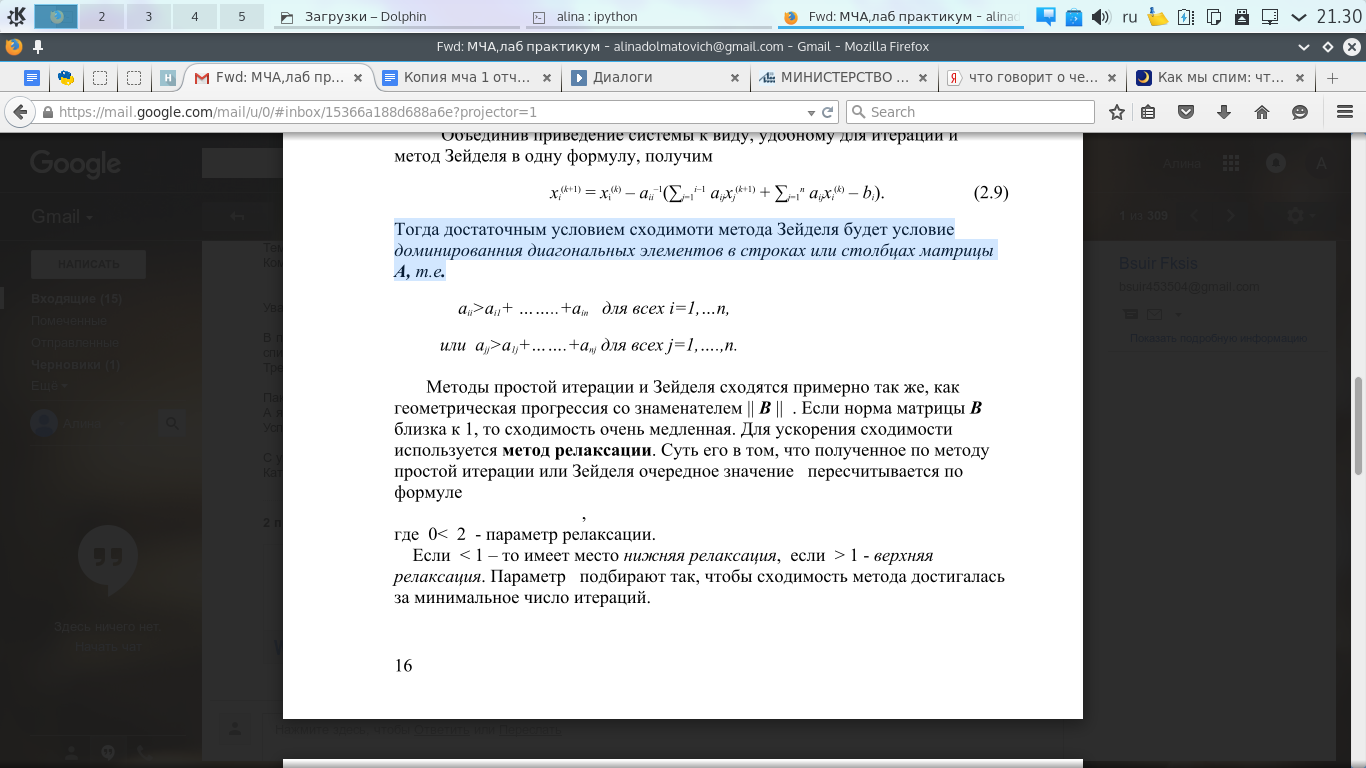
или в развернутой форме записи



Объединив приведение системы к виду, удобному для итераций и метод Зейделя в одну формулу, получим



Тогда достаточным условием сходимоти метода Зейделя будет условие доминированния диагональных элементов в строках или столбцах матрицы A, т.е.



Методы простой итерации и Зейделя сходятся примерно так же, как геометрическая прогрессия со знаменателем || B || . Если норма матрицы B близка к 1, то сходимость очень медленная. Для ускорения сходимости используется метод релаксации.

**Распечатка кода программы**

import numpy

def simpleIterations(A, B, c, e):

xk = numpy.zeros((len(A), 1))

xk = c

x = numpy.dot(B, xk) + c

numberOfIterations = 0

while abs(max((x - xk).flatten())) > e:

xk = x

x = numpy.dot(B, xk) + c

numberOfIterations += 1

print 'Simple iterations method. Result:'

print x

print 'Number of iterations {}'.format(numberOfIterations)

def zeidel(A, B, c, e):

yk = numpy.zeros((len(A), 1))

yk = c

y = numpy.dot(B, yk) + c

numberOfIterations = 1

while abs(max((y - yk).flatten())) > e:

yk = y

for i in range(len(A)):

y[i] = 0

for j in range(i):

y[i] = y[i] + B[i][j] \* y[j]

for j in range(i, len(A)):

y[i] = y[i] + B[i][j] \* yk[j]

y[i] = y[i] + c[i]

numberOfIterations += 1

print 'Zeidel method. Result:'

print y

print 'Number of interations {}'.format(numberOfIterations)

def main():

print("Start working")

A = numpy.array([[24.41, 2.42, 3.85],

[2.31, 31.49, 1.52],

[3.49, 4.85, 28.92]],

"f")

b = numpy.array([[30.24],

[40.95],

[42.81]],

"f")

c = numpy.zeros((len(A), 1))

B = numpy.zeros((len(A), len(A)))

e = 0.00001

for i in xrange(len(A)):

c[i] = b[i] / A[i][i]

for j in xrange(len(A)):

B[i][j] = -A[i][j] / A[i][i]

B[i][i] = 0

print B

print c

maxx = [0 for i in range(len(A))]

for i in range(len(A)):

for j in range(len(A)):

maxx[i] = maxx[i] + abs(B[i][j])

print("Convergence test result:")

print max(maxx)

simpleIterations(A, B, c, e)

zeidel(A, B, c, e)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Вывод:**

На практике для решения достаточно больших СЛАУ используются итерационные методы. Это обусловлено их скоростью, а так же экономией оперативной памяти.

Итерационные методы основаны на использовании повротяющегося процесса и позволяют получить решение в результате последовательных приближений. Итерационные методы устанавливают процедуру уточнения определенного начального приближения к решению. Они полволяют достичь любой точности простым повторением итераций. Преимущество этих методов в том, что они позволяют достичь решения с заданной точностью быстрее прямых методов, а так же позволяют решать большие системы уравнений. При использовании итерационных методов крайне важно помнить о проверке критерия сходимости системы.