**Национальный Технический Университет Украины “КПИ им. Игоря Сикорского”**

**Учебно-научный комплекс**

**«Институт прикладного системного анализа»**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**По системному анализу**

**Вариант №1**

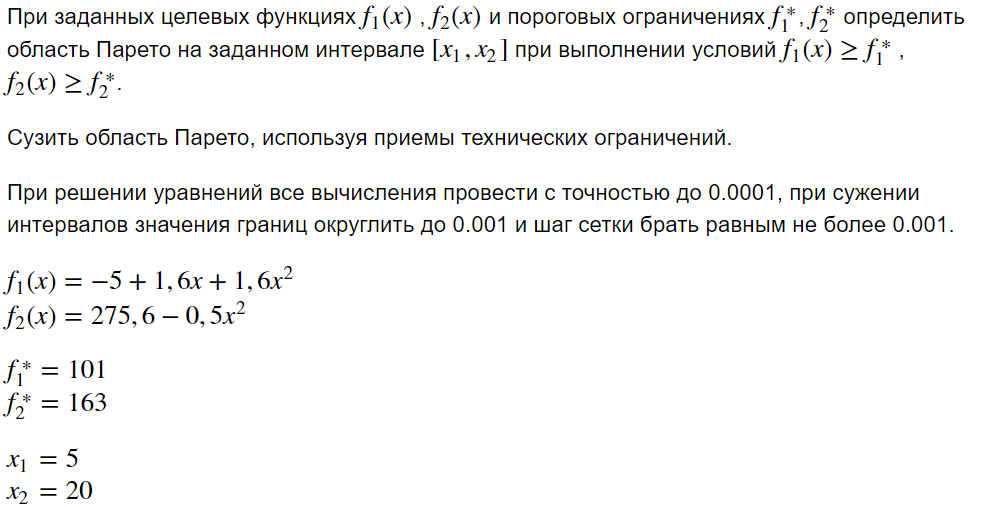
**Студетна 4 курса**

**группы КА-41**

**Барзия Ильи**

Київ 2017

**Задание 1**



**Код программы:**

import math as m

import itertools as it

import functools as ft

import operator as o

from matplotlib import pyplot as plt

import matplotlib

import numpy as np

def every(n,l):

return map(o.itemgetter(1), filter(lambda x: not (x[0] % n), enumerate(l)))

def f1(x):

return -5 + 1.6 \* x + 1.6 \* (x \* x)

def f2(x):

return 275.6 - 0.5 \* (x \* x)

def t(x):

return (round(f1(x) / F1,3) , round(f2(x) / F2, 3), x)

def an(x):

return (f1(x) / F1, f2(x) / F2)

F1, F2 = 101 , 163

x1, x2 = 5 , 20

STEP = 0.001

def first():

steps = int((x2 - x1)/STEP) + 1

set1 = ((x1 + x \* STEP) for x in range(steps))

par1 = filter(lambda x: f2(x) / F2 >= 1, set1)

par = filter(lambda x: f1(x) / F1 >= 1 ,par1)

x = np.array(range(x1,x2))

itr= set(map(t,par))

minmax = ft.reduce(ft.partial(min, key=o.itemgetter(0)),

map(lambda y: (max(y[0], y[1]), y[2]),itr))

maxmin = ft.reduce(ft.partial(max, key=o.itemgetter(1)),

map(lambda y: (min(y[0], y[1]), y[2]),itr))

return(minmax, maxmin)

ans = first()

print("Ответ к заданию 1:")

print("x = "+str(ans[0][1])+"\nminmax = "+str(ans[0][0])+"\nf1/f\*1 = {z[0]}\nf2/f\*2 = {z[1]}\n".format(z=an(ans[0][1])))

print("x = "+str(ans[1][1])+"\nmaxmin = "+str(ans[1][0])+"\nf1/f\*1 = {z[0]}\nf2/f\*2 = {z[1]}".format(z=an(ans[1][1])))

**Ответ (вывод программы):**

Ответ к заданию 1:

x = 9.18

minmax = 1.432

f1/f\*1 = 1.4309291089108909

f2/f\*2 = 1.4322932515337425

x = 7.655

maxmin = 1.0

f1/f\*1 = 1.0000637623762378

f2/f\*2 = 1.5110459355828223

**Дополнительно: проверка и графическое решение**

**Ответ (вывод программы) после проверки:**

Ответ к заданию 1:

x = 9.18

minmax = 1.432

f1/f\*1 = 1.4309291089108909

f2/f\*2 = 1.4322932515337425

x = 15.006

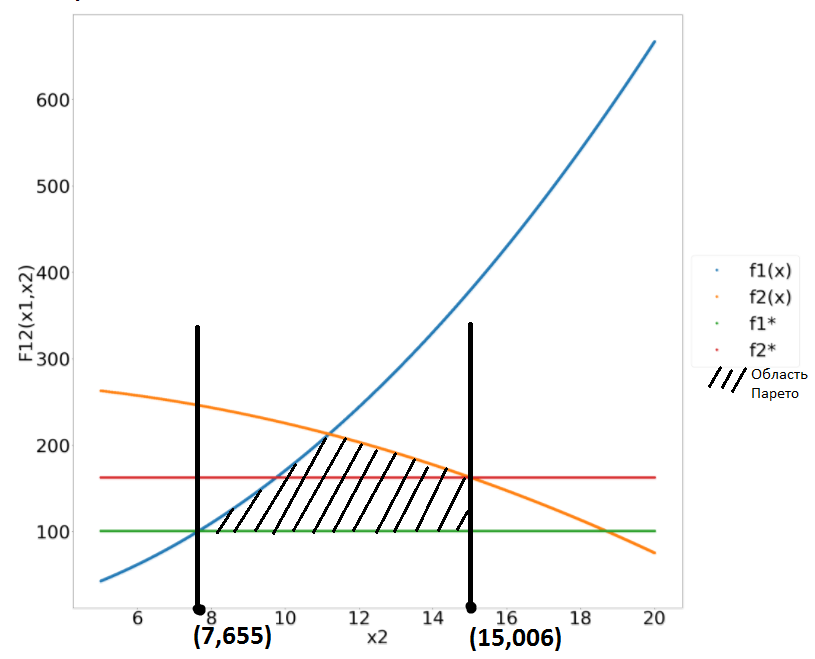
maxmin = 1.0

f1/f\*1 = 3.7554223524752475

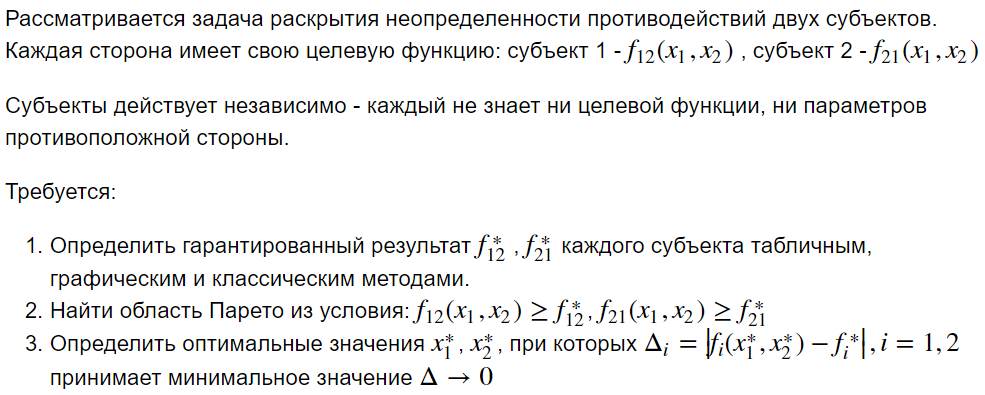
f2/f\*2 = 1.000061239263804

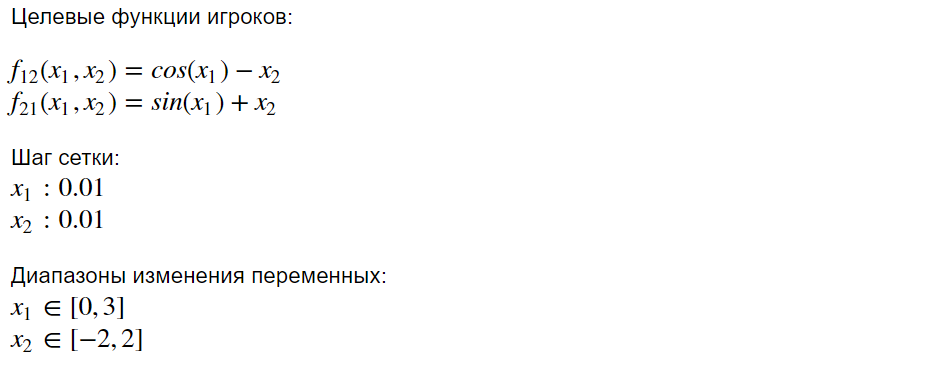
**Причина ошибки: поиск minmin вместо maxmin.**

**Графическое решение:**



**Задание 2**





**Задание 2.1**

**Табличный метод**

**Код программы:**

def F12(x):

return m.cos(x[0]) - x[1]

def F21(x):

return m.sin(x[0]) + x[1]

def F211(x):

return m.sin(x[1]) + x[0]

st = (0.01, 0.01)

ed = ((0,3),(-2,2))

stx1 = int((ed[0][1] - ed[0][0])/st[0]) + 1

stx2 = int((ed[1][1] - ed[1][0])/st[1]) + 1

x1 = {(ed[0][0] + x \* st[0]) for x in range(stx1)}

x2 = {(ed[1][0] + x \* st[1]) for x in range(stx2)}

f12 = ft.reduce(ft.partial(max, key=o.itemgetter(0)),map(lambda il:ft.reduce(ft.partial(min, key=o.itemgetter(0)),

map(lambda x: (F12(x),x), il)),(((x,y) for y in x2) for x in x1)))

f21 = ft.reduce(ft.partial(max, key=o.itemgetter(0)),map(lambda il:ft.reduce(ft.partial(min, key=o.itemgetter(0)),

map(lambda x: (F21(x),x), il)),(((y, x) for y in x1) for x in x2)))

print("Ответ к заданию 2:")

print("\nmax(x1)min(x2)(F12) = "+str(f12[0])+"\nx1 = "+str(f12[1][0])+"\nx2 = "+str(f12[1][1]))

print("\nmax(x2)min(x1)(F21) = "+str(f21[0])+"\nx1 = "+str(f21[1][0])+"\nx2 = "+str(f21[1][1]))

**Ответ (вывод программы):**

Ответ к заданию 2:

max(x1)min(x2)(F12) = -1.0

x1 = 0.0

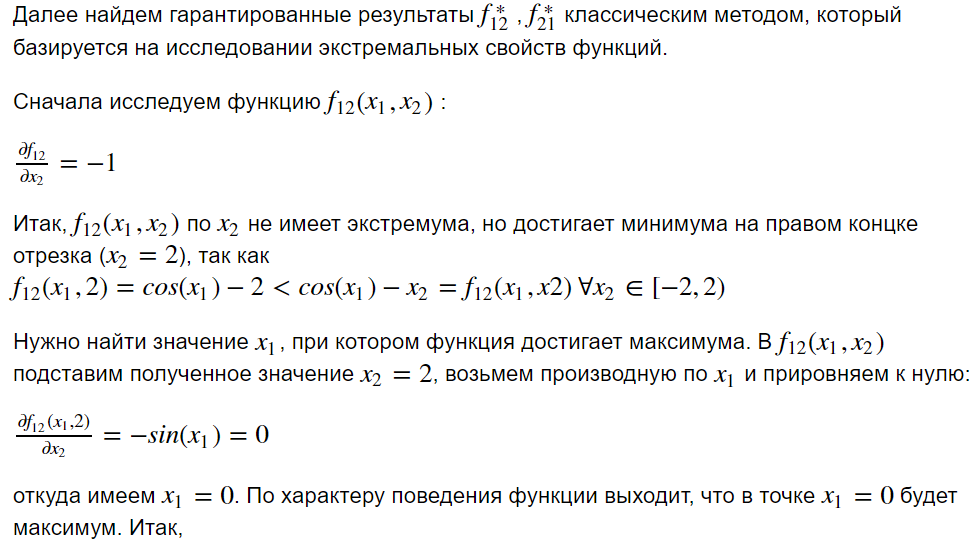
x2 = 2.0

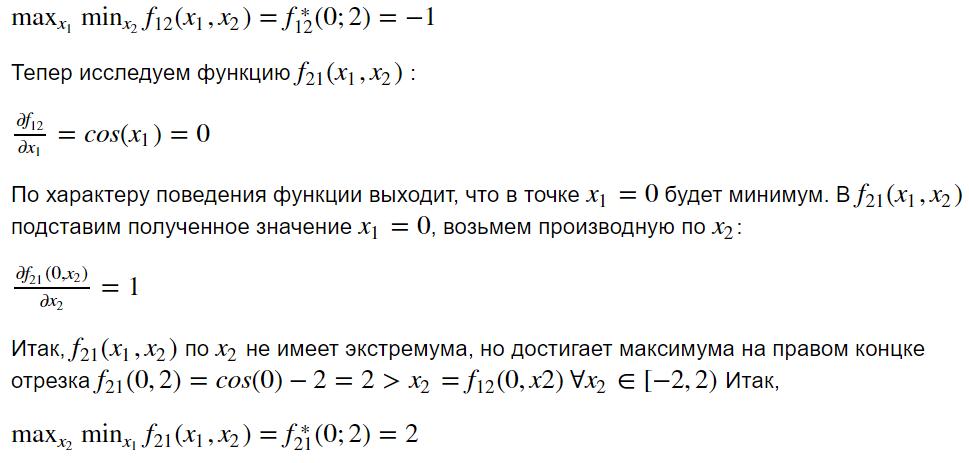
max(x2)min(x1)(F21) = 2.0

x1 = 0.0

x2 = 2.0

**Классический метод**

****

****

**Графический метод**

**Код программы:**

matplotlib.rcParams.update({'font.size': 50})

fst = o.itemgetter(0)

snd = o.itemgetter(1)

plt.figure(figsize=(30, 30))

for splt in ([(x,y) for y in x2] for x in every(30, sorted(x1))):

plt.plot(list(map(snd, splt)), list(map(F12, splt)),'o', label=str(round(splt[0][0],3)))

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5))

plt.xlabel('x2')

plt.ylabel('F12(x1,x2)')

plt.show()

plt.close()

plt.figure(figsize=(30, 30))

for splt in ([(x,y) for y in x1] for x in every(40, sorted(x2))):

plt.plot(list(map(snd, splt)), list(map(F211, splt)),'o', label=str(round(splt[0][0],3)))

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5))

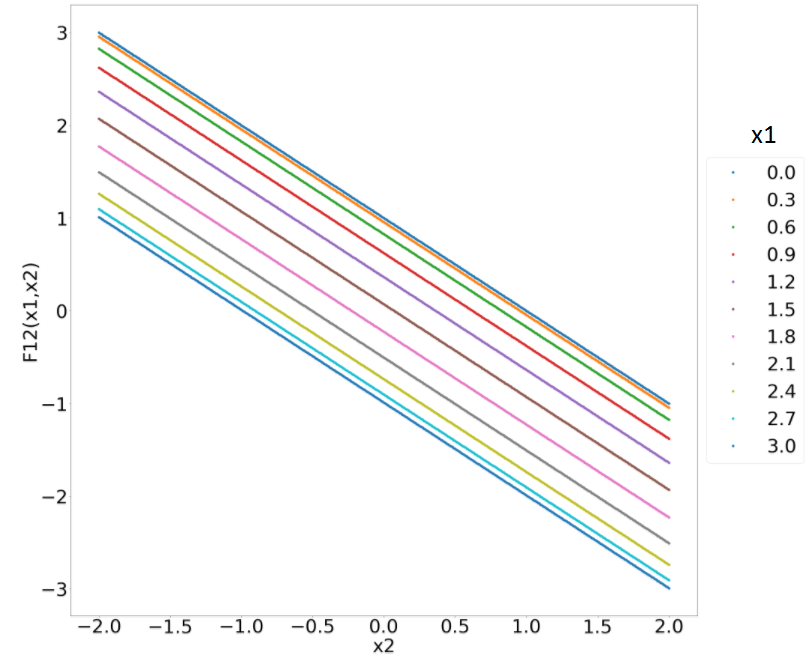
plt.xlabel('x1')

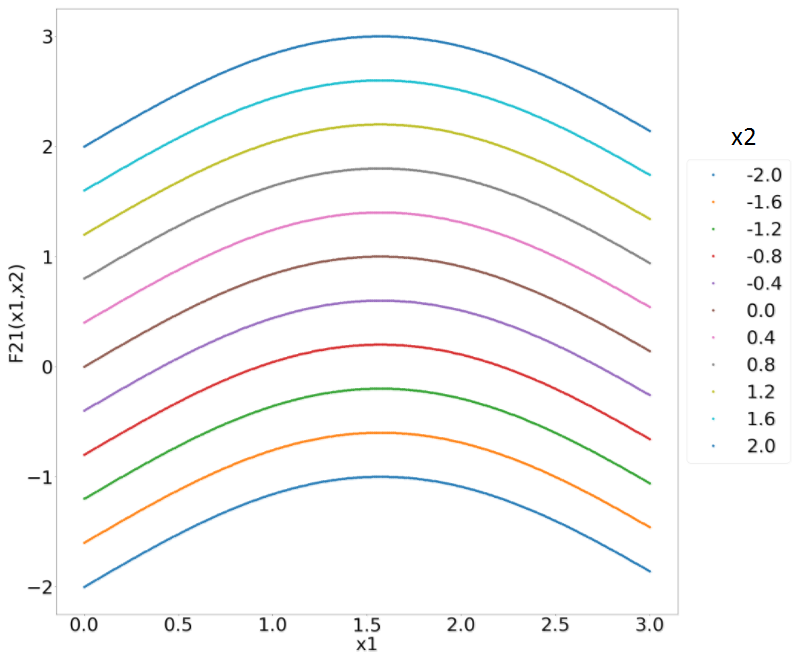
plt.ylabel('F21(x1,x2)')

plt.show()

plt.close

**Ответ (вывод программы):**





**Задание 2.2**

**Код программы:**

def getf1(x,y):

return m.cos(x) + 1 + y

def getf2(x,y):

return 2 - m.sin(x) + y

plt.figure(figsize=(20, 20))

valx = [x for x in sorted(x1)]

valy1 = [getf1(y,0) for y in valx]

valy2 = [getf2(y,0) for y in valx]

plt.plot(valx, valy1,'go', label="F12=f12\*")

plt.fill\_between(valx, valy1,0,facecolor='blue', interpolate=True,label="F12>f12\*")

plt.fill\_between(valx, valy1,2,facecolor='red', interpolate=True,label="F12<f12\*")

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5))

plt.xlabel('x1')

plt.ylabel('x2')

plt.show()

plt.close()

plt.figure(figsize=(20, 20))

plt.plot(valx, valy2,'go', label="F21=f21\*")

plt.fill\_between(valx, valy2,0,facecolor='red', interpolate=True,label="F21<f21\*")

plt.fill\_between(valx, valy2,2,facecolor='blue', interpolate=True,label="F21>f21\*")

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5))

plt.xlabel('x1')

plt.ylabel('x2')

plt.show()

plt.close()

plt.figure(figsize=(20, 20))

plt.plot(valx, valy1,'ro', label="F12=f12\*")

plt.plot(valx, valy2,'bo', label="F21=f21\*")

valx=np.array(valx)

valx=np.array(valx)

valy1=np.array(valy1)

valy1=np.array(valy1)

plt.fill\_between(valx, valy1, valy2, where =valx<=1.6 ,facecolor='green', interpolate=True,label="область Парето")

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5))

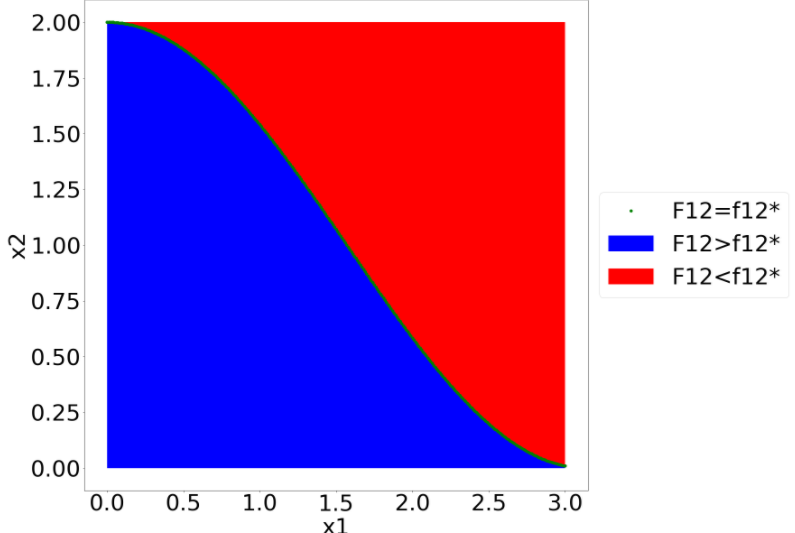
plt.xlabel('x1')

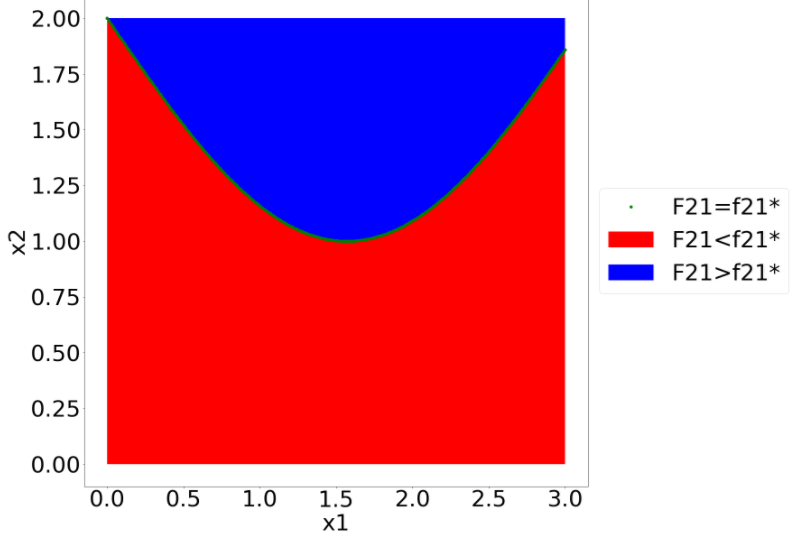
plt.ylabel('x2')

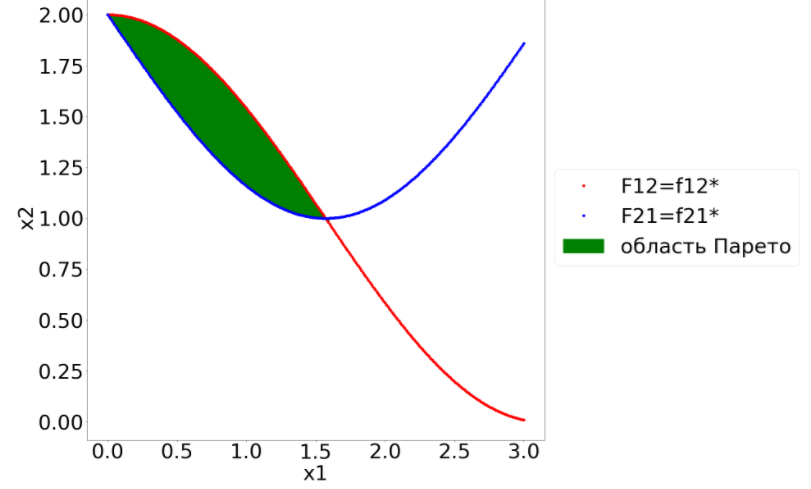
plt.show()

plt.close()

**Ответ (вывод программы):**







**Задание 2.3**

**Код программы:**

delta\_i = lambda x: max(abs(F12(x) - f12[0]), abs(F21(x) - f21[0]))

delta = min(it.product(x1, x2), key=lambda x: delta\_i(x))

print("Оптимальные значения x1\* и x2\*:")

for i in range(delta.shape):

print("x1\* = "+str(delta[0])+ "\nx2\* = "+str(delta[1]))

**Ответ (вывод программы):**

Оптимальные значения x1\* и x2\*:

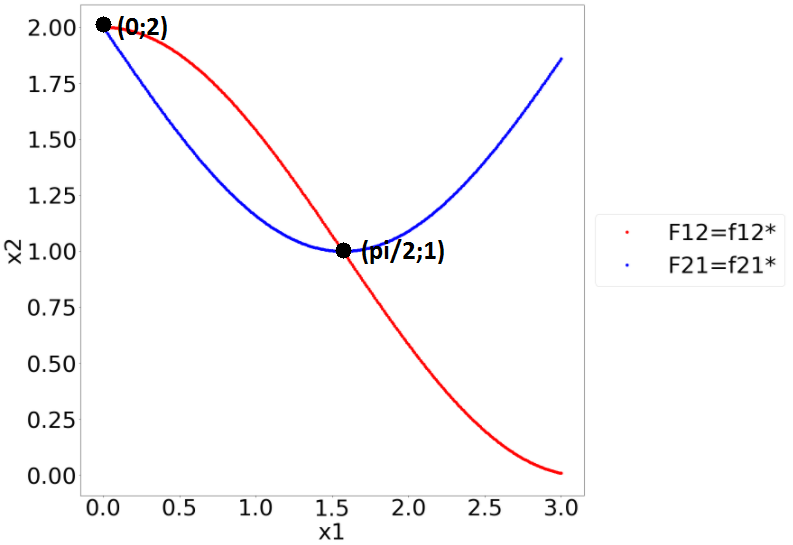
x1\* = 0.0

x2\* = 2.0

x1\* = 1.57

x2\* = 1

**Графическое изображение:**

****