Лабораторная работа №7

Численное решение системы линейных уравнений, численное интегрирование **Выполнила:** Клюшенкова Полина Анатольевна: 429

Цель работы: научиться решать задачу Коши для ОДУ разными методами

In [2]:

```
from sympy import diff, symbols, cos, sin, Function, Symbol, dsolve, exp
from sympy import *
from IPython.display import *
import scipy as sp
import numpy as np
import matplotlib.pyplot as plt
import math as mt
import copy
```

In [3]:

```
f = Function('f')
x = Symbol('x')
```

In [4]:

```
f = Function('f')
x = Symbol('x')
var(' C1 C2')
# - cos(3*x)*exp(-x)
des = dsolve(5*f(x).diff(x, x) + 8*f(x).diff(x) + 4*f(x) )
display(des)
eq1 = des.rhs.subs(x, 1)
display(eq1)
eq2 = des.rhs.diff(x).subs(x, 0)
display(eq2)
resh_sys = solve([eq1, eq2], C1, C2)
display(resh_sys)
res = des.rhs.subs([(C1, resh_sys[C1]), (C2, resh_sys[C2])])
display(res)
```

```
Eq(f(x), (C1*sin(2*x/5) + C2*cos(2*x/5))/exp(x)**(4/5))

(C1*sin(2/5) + C2*cos(2/5))*exp(-4/5)

2*C1/5 - 4*C2/5

{C2: 0, C1: 0}

0
```

In [6]:

```
x = np.arange(0, 2, 0.05)
f = lambdify(x, res, 'numpy')
plt.plot(x, f)
plt.grid()
plt.show()
```

```
  File "<string>", line 1
    lambda 0.0,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5,0.55,0.6,0.65,0.
7,0.75,0.8,0.85,0.9,0.95,1.0,1.05,1.1,1.15,1.2,1.25,1.3,1.35,1.4,1.45,1.5,1.
55,1.6,1.65,1.7,1.75,1.8,1.85,1.9,1.95: (0)
             ^
SyntaxError: invalid syntax
```

Решение задачи Коши неявным методом Адамса Для m=1:

$$f^L(t) = f(t_{i-1}, u_{i-1})\frac{t - t_i}{t_{i-1} - t_i} + f(t_i, u_i)\frac{t - t_i}{t_i - t_{i-1}}$$

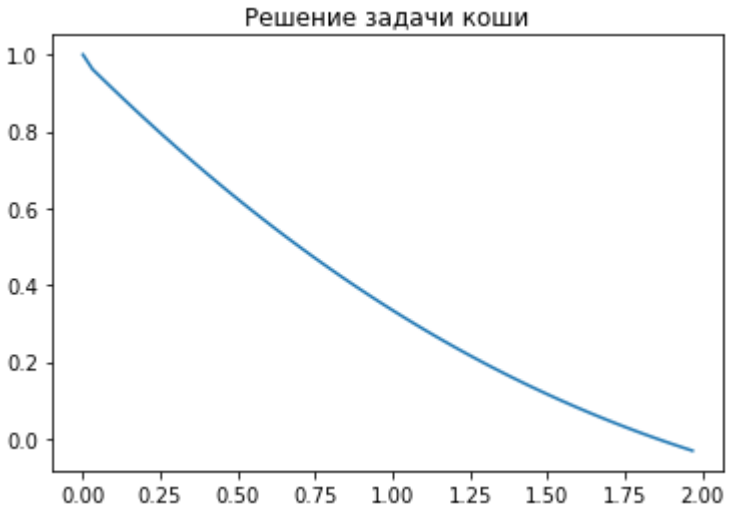
$$u_{i+1} = u_i + 0.5\tau(f(t_i, u_i) + 0.\overline{5(f(t_i, u_i) - f(t_{i-1}, u_{i-1})}$$

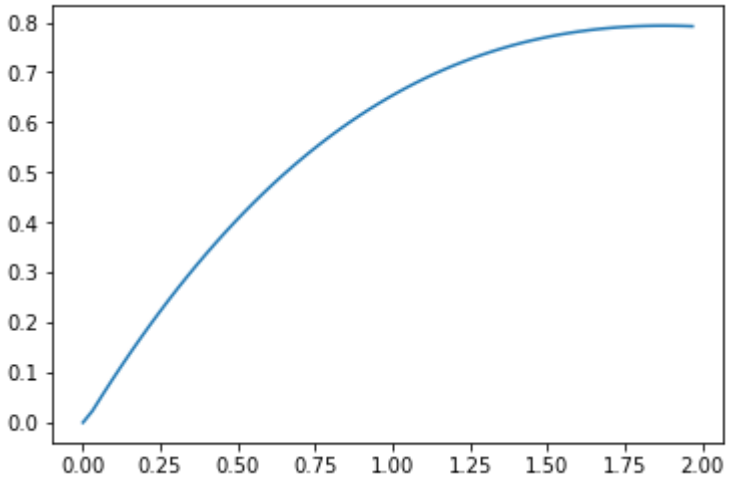
$$u_0 = u^0, i = \overline{0, n-1}$$

In [17]:

```
a = 1./3
b = 2./3
e = 0.01
k = 2
u1 = np.zeros((10000))
u0 = np.zeros((10000))
up = np.zeros((10000))
i = 0
u1[0] = 1
up[0] = 0
n=0
while abs(u1[2*i]-up[i])/abs(u1[2*i])>e :
    n+=1
    x = np.arange(0, 2, 1/k)
    t = 2./k
    up = u1
    u1[0] = 1
    u0[0] = 0
    for j in range(k):
        u1[j+1] = u1[j] + t/2*(-8/5*u1[j]-4/5*u0[j]+x[j]/5*exp(-x[j])+0.5*(-8/5*u1[j]-4/5*u
        u0[j+1] = u0[j] + t/2*(u1[j]+0.5*(u1[j]+u1[j-1]))
#         print('u1: ', u1[j+1])
#         print('u0: ', u0[j+1])
    i = 1
    k = k*2
    if n>5:
        break
f = []
i=0
while u1[i] !=0:
    f.append(u1[i])
    i+=1
fp = []
i=1
fp.append(0)
while u0[i] !=0:
#     print(u0[i])
    fp.append(u0[i])
    i+=1
# print(f)
# print(fp)
plt.plot(np.arange(0, 2, 2/len(f)), f)
plt.title('Решение задачи коши')
plt.show()

plt.plot(np.arange(0, 2, 2/len(f)), fp)
plt.show()
```






```
Найдем точное решение
```

In [13]:

```
import numpy as np
from sympy import *
from IPython.display import *
import matplotlib.pyplot as plt
init_printing(use_latex=True)
var('t C1 C2')
u = Function("u")(t)  # Это переменная, но не функция.
# m=20 #Показатель массы.
# w=10.0#Показатель демпфирования колебаний.
# c=0.3#Показатель жёсткости.
# a=1#Бесконечный импульс силы.
#Все показатели условные(только для исследования характера зависимостей).
t#Текущее время.
# r = 9
# m = 0.33333333
# c = 3
a=0
de = Eq(5*u.diff(t,t)+8*u.diff(t)+4*u- t*exp(-t), a) #-Дифференциальное уравнение колебаний
display(de)#-Вывод на дисплей.
des = dsolve(de,u)#Символьное решение уравнения методом Коши в общем виде.
display(des)#Вывод на дисплей.
eq1=des.rhs.subs(t,0)-1;#Условие равенства нулю перемещения в момент времени t=0.
display(eq1)#Вывод на дисплей.
eq2=des.rhs.diff(t).subs(t,0)#Условие равенства нулю скорости перемещения в момент
# времени t=0.
display(eq2)#Вывод на дисплей.
seq=solve([eq1,eq2],C1,C2)#Решение системы для определения коэффициентов C1,C2.
display(seq)#Вывод на дисплей
rez=des.rhs.subs([(C1,seq[C1]),(C2,seq[C2])])#Вид решения дифференциального
#уравнения с численными значениями коэффициентов.
display(rez)#Вывод на дисплей.
f=lambdify(t, rez, "numpy")#Перевод символьного решения в численное для работы
#с модулем numpy .
x = np.arange(0.0,2,0.05)
plt.plot(x,f(x),color='b', linewidth=3)
plt.xlabel('Time t seconds',fontsize=12)
plt.ylabel('$f(t)$',fontsize=16)
plt.grid(True)
plt.show()
```

$$-te^{-t} + 4u(t) + 8\frac{d}{dt}u(t) + 5\frac{d^2}{dt^2}u(t) = 0$$

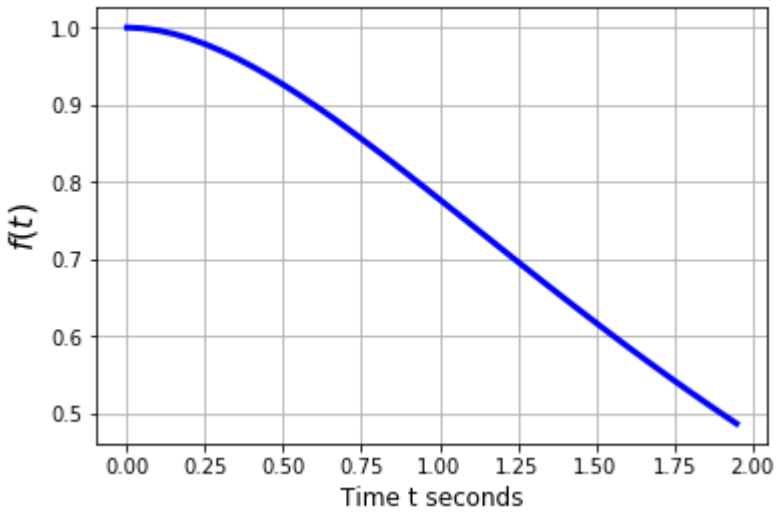
$$u(t) = (t+2)\,e^{-t} + \frac{1}{(e^t)^{\frac{4}{5}}}\left(C_1 \sin\left(\frac{2t}{5}\right) + C_2 \cos\left(\frac{2t}{5}\right)\right)$$

$$C_2 + 1$$

$$\frac{2C_1}{5} - \frac{4C_2}{5} - 1$$

$$\left\{C_1 : \frac{1}{2}, \quad C_2 : -1\right\}$$

$$(t+2)\,e^{-t} + \frac{1}{(e^t)^{\frac{4}{5}}}\left(\frac{1}{2}\sin\left(\frac{2t}{5}\right) - \cos\left(\frac{2t}{5}\right)\right)$$



Решим задачу Коши методом Ронге-Кутты

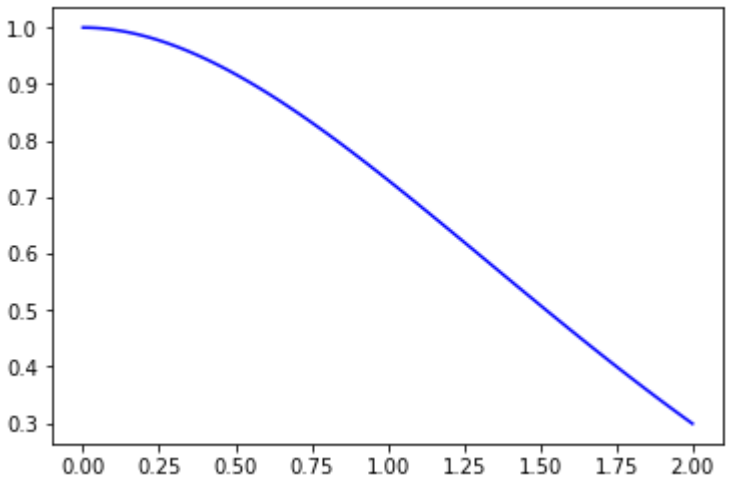
In [15]:

```
import numpy as np
import scipy as sp
from scipy.integrate import odeint
import matplotlib.pyplot as plt

def g(y, x):
    y0 = y[0]
    y1 = y[1]
    y2 = -8/9*y1 -4/5*y0 + x/5*exp(-x)
    return y1, y2

# Initial conditions on y, y' at x=0
init = 1.0, 0.0
# First integrate from 0 to 2
x = np.linspace(0,2,100)
sol=odeint(g, init, x)
# Then integrate from 0 to -2
plt.plot(x, sol[:,0], color='b')

plt.show()
```



Вывод:в ходе лабораторной работы была решена задача Коши для ОДУ второго порядка разлиными методами