**Комп’ютерний практикум №6**

**Розв’язання задачі Коші методами Рунге-Кутта та Адамса**

**Виконав:**

Студент 3 курсу НН ФТІ

групи ФІ-92

Поночевний Назар Юрійович

Варіант 12

**Завдання:**

Методами Рунге-Кутта та Адамса-Башфорта четвертого порядку розв'язати задачу Коші. На початку інтервалу у необхідній кількості точок значення для методу Адамса визначити методом Рунге-Кутта четвертого порядку.

1. З умови маємо:

y' = (1 - x^2) \* y + F(x)

h = 0.1

y(0) = 1

Точний розв'язок: y = cos x

Підставимо:

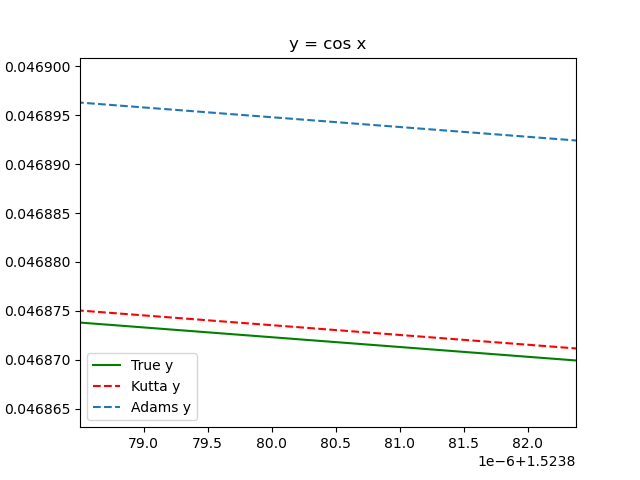
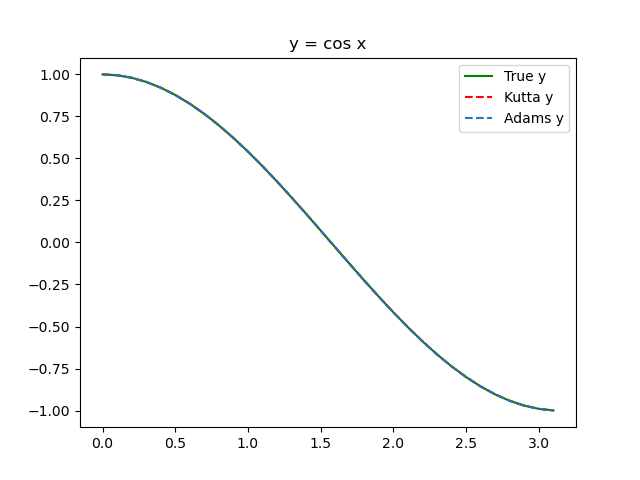
F(x) = y' - (1 - x^2) \* y = - sin x - (1 - x^2) \* cos x

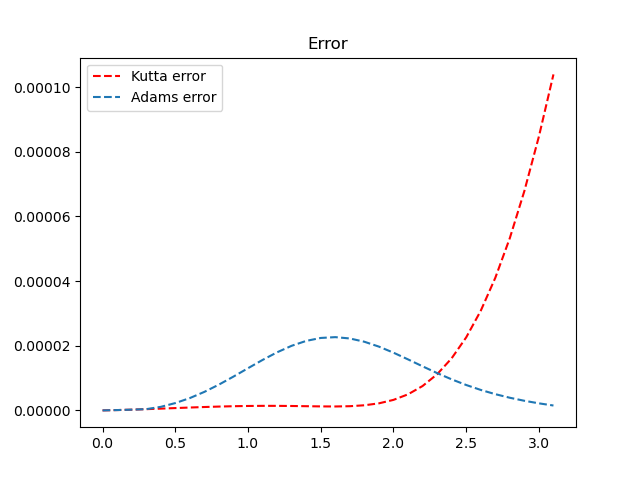
y' = (1 - x^2) \* (y - cos x) - sin x

1. Реалізуємо програму

| """ Solving Cauchy problem  """  import math import numpy as np import matplotlib.pyplot as plt   # ------------ Input ------------   def f(x, y):  return ((1 - x\*\*2) \* (y - math.cos(x))) - math.sin(x)   H, Y0 = 0.1, 1 A, B = 0, math.pi   # ------------ Code ------------   def runge\_kutt(h, y0):  x = np.arange(A, B, h)  y = [y0]  for i in x:  k1 = h \* f(i, y[-1])  k2 = h \* f(i + h / 2, y[-1] + k1 / 2)  k3 = h \* f(i + h / 2, y[-1] + k2 / 2)  k4 = h \* f(i + h, y[-1] + k3)  y.append(y[-1] + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6)  return tuple(zip(x, y))   def adams(h, y0):  runge\_kutt\_ = runge\_kutt(h, y0)[:4]  x = np.arange(A, B, h)  y = [i[1] for i in runge\_kutt\_]  for i in range(3, len(x)):  y.append(y[i] + (55 \* f(x[i], y[i]) -  59 \* f(x[i - 1], y[i - 1]) +  37 \* f(x[i - 2], y[i - 2]) -  9 \* f(x[i - 3], y[i - 3])) \* h / 24)  return tuple(zip(x, y))   def main():  y\_kutt = np.array([i[1] for i in runge\_kutt(H, Y0)])  y\_adam = np.array([i[1] for i in adams(H, Y0)])   x = np.arange(A, B, H)  y\_true = np.array([math.cos(i) for i in x])   e\_kutt = np.abs(y\_true - y\_kutt)  e\_adam = np.abs(y\_true - y\_adam)   plt.plot(x, y\_true, color="green", label="True y")  plt.plot(x, y\_kutt, color="red", linestyle='--', label="Kutta y")  plt.plot(x, y\_adam, linestyle='--', label="Adams y")  plt.title("y = cos x")  plt.legend()  plt.show()   plt.plot(x, e\_kutt, color="red", linestyle='--', label="Kutta error")  plt.plot(x, e\_adam, linestyle='--', label="Adams error")  plt.title("Error")  plt.legend()  plt.show()   if \_\_name\_\_ == "\_\_main\_\_":  main() |
| --- |

1. Результат





1. Контрольні запитання

**Що таке однокрокові та багатокрокові методи розв’язання звичайних диференційних рівнянь?**

В однокрокових методах для знаходження наступної точки на кривій y = f(х) потрібна інформація лише про один попередній крок. В багатокрокових методах для знаходження наступної точки на кривій потрібна інформація більш ніж про одну з попередніх точок.

**Який порядок точності має метод Ейлера?**

Метод Ейлера є явним, однокроковим методом першого порядку точності.