

TØ-opgaver uge 11

Noah Rahbek Bigum Hansen

12. November 2024

Opg. 1

Vi skal differentiere funktionen $f(x)$, givet ved

$$f(x) = \sin(e^{3x}).$$

1.

Differentier f ved hjælp af Python.

Følgende prompt er givet til Anaconda Assistant 4.0.39:

python code for differentiating $\sin(e^{3x})$ (using no context from Notebook).

Anaconda Assistant gav da følgende python-kode:

```
import sympy as sp

# Define the variable and the function
x = sp.symbols('x')
function = sp.sin(sp.exp(3 * x))

# Differentiate the function
derivative = sp.diff(function, x)
derivative
```

Hvilket giver følgende resultat når koden køres

$$3e^{3x} \cos(e^{3x}).$$

2.

Differentier f i hånden ved hjælp af matematik. Får du det samme resultat i begge tilfælde?

Kædereglen siger at

$$y' = f'(g(x)) \cdot g'(x).$$

I tilfældet ovenfor sætter vi

$$f(u) = \sin(u), \text{ hvor } u = e^{3x}.$$

Vi får dermed

$$y' = 3e^{3x} \cos(3^{3x}).$$

Dette er altså det samme resultat som blev fundet vha. Python.

Opg. 2

I denne opgave skal vi beregne integralet

$$\int_0^1 e^{-x^2} dx$$

ved hjælp af Python. Det kan bemærkes, at integralet ikke kan udregnes ved hjælp af elementære funktioner.

Følgende prompt er givet til Anaconda Assistant 4.0.39:

```
python code for integrating  $e^{-x^2}$  from 0 to 1
```

Anaconda Assistant gav da følgende python-kode:

```
import numpy as np
from scipy.integrate import quad

# Define the function to integrate
def integrand(x):
    return np.exp(-x**2)

# Perform the integration from 0 to 1
result, error = quad(integrand, 0, 1)
```

Hvilket giver følgende resultat når koden køres

$$0,746\,824\,132\,812\,427\,1.$$

Opg. 3

I denne opgave skal vi beregne integralet

$$\int_0^1 \int_0^y x^2 y^2 dx dy \tag{1}$$

1.

Beregn integralet (1) ved hjælp af Python.

Følgende prompt er givet til Anaconda Assistant 4.0.39:

```
python code for calculating the integral of the integral of  $x^2y^2dx dy$  from 0 to  $y$  from 0 to 1
```

Anaconda Assistant gav da følgende python-kode:

```
import sympy as sp

# Define the variables
x, y = sp.symbols('x y')

# Define the function
function = x**2 * y**2

# Perform the first integral with respect to x from 0 to y
inner_integral = sp.integrate(function, (x, 0, y))

# Now perform the second integral with respect to y from 0 to 1
result = sp.integrate(inner_integral, (y, 0, 1))

result
```

Hvilket giver følgende resultat når koden køres

$$\frac{1}{18}.$$

2.

Beregn integralet (1) i hånden ved hjælp af matematik. Får du det samme resultat i begge tilfælde?

Først beregnes det inderste integrale som

$$\begin{aligned} y^2 \int_0^y x^2 dx &= \left[\frac{1}{3} x^3 \right]_0^y \\ &= \frac{1}{3} y^3 \end{aligned}$$

Og det yderste integrale som

$$\begin{aligned}\frac{1}{3} \int_0^1 y^5 dy &= \left[\frac{1}{6} y^6 \right]_0^1 \\ &= \frac{1}{3 \cdot 6} \\ &= \frac{1}{18}.\end{aligned}$$

Hvilket altså er det samme resultat som Python-koden gav.

Opg. 4

Betragt differentiaalligningen

$$y' = y^2 + 4 \tag{2}$$

1.

Er differentiaalligningen (2) lineær?

Nej, idet der indgår et y^2 -led er differentiaalligningen ikke en linearkombination af y og y 's afledede

2.

Benyt Python til at løse ligningen (eksakt).

Følgende prompt er givet til Anaconda Assistant 4.0.39:

```
python code for solving the differential equation  $y' = y^2 + 4$ 
```

Anaconda Assistant gav da følgende python-kode:

```
import sympy as sp

# Define the variables
y = sp.Function('y')
x = sp.symbols('x')

# Define the differential equation
differential_eq = sp.Eq(y(x).diff(x), y(x)**2 + 4)

# Solve the differential equation
solution = sp.dsolve(differential_eq, y(x))

solution
```

Hvilket giver følgende resultat når koden køres

$$y(x) = -2 \tan(C_1 - 2x).$$

3.

Plot løsningen til (2).

Følgende prompt er givet til ChatGPT 4o:

```
python code for solving the differential equation  $y' = y^2 + 4$ 
```

Anaconda Assistant gav da følgende python-kode:

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

# Define the differential equation with correct arguments
def dydx(y, x):
    return y**2 + 4

# Set the range of x values
x = np.linspace(0, 10, 100)

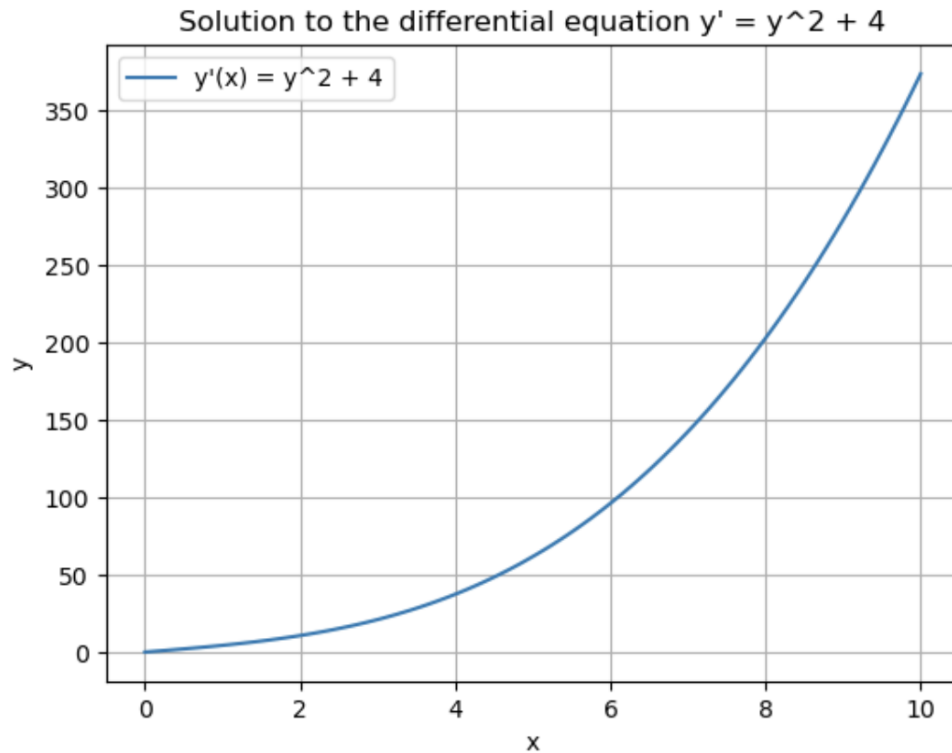
# Initial condition
y0 = 0.0

# Solve the differential equation using odeint
y = odeint(dydx, y0, x, tfirst=True)

# Plot the solution
plt.plot(x, y[:, 0], label="y'(x) = y^2 + 4")
plt.xlabel('x')
plt.ylabel('y')
plt.title("Solution to the differential equation y' = y^2 + 4")
plt.legend()
plt.grid(True)
plt.show()
```

Hvilket giver følgende resultat når koden køres

Figur 1: Resultatet af Python-koden



Opg. 5

Lad $f(x, y)$ betegne funktionen

$$f(x, y) = x^3y - y^3x.$$

Maksimer funktion $f(x, y)$ over området $0 \leq x \leq 1$ og $0 \leq y \leq 1$ ved hjælp af Python.
Ufordring (lidt svært): Overvej om du kan løse denne opgave i hånden vha. matematik

Følgende prompt er givet til Anaconda Assistant 4.0.39:

```
Write some python code that maximizes  $x^3y - y^3x$  over the interval  $0 \leq x \leq 1$  and  $0 \leq y \leq 1$ 
```

Anaconda Assistant gav da følgende python-kode:

```
import numpy as np
from scipy.optimize import minimize

# Define the function to maximize
def objective(vars):
    x, y = vars
```

```
    return -(x**3 * y - y**3 * x) # Negate to convert maximization to minimization

# Define the bounds for x and y
bounds = [(0, 1), (0, 1)]

# Initial guess
initial_guess = [0.5, 0.5]

# Perform the optimization
result = minimize(objective, initial_guess, bounds=bounds)

# Extract the optimal values
optimal_x, optimal_y = result.x
max_value = -result.fun # Negate again to get the maximum value

optimal_x, optimal_y, max_value
```

Hvilket giver følgende resultat når koden køres

$(1.0, 0.5773501788044133, 0.38490017945973637)$.

I koden kan det aflæses at dette skal tolkes som at punktet hvor funktionen er maksimeret er $(1, 0; 0, 577)$, hvor funktionsværdien er 0,385