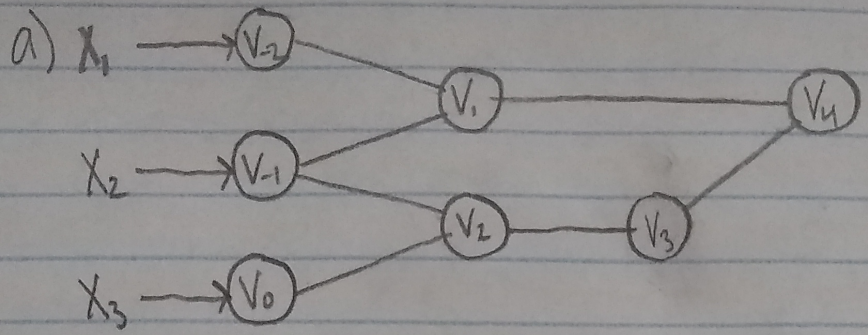
1) $f(X_1, X_2, X_3) = X_1 X_2 \sin(X_2 X_3)$

a)



b)

$V_{-2} = X_1 = -1$
$V_{-1} = X_2 = 2$
$V_0 = X_3 = 3$
$V_1 = V_{-2} V_{-1} = -2$
$V_2 = V_{-1} V_0 = 6$
$V_3 = \sin(V_2) = -0.279$
$V_4 = V_1 V_3 = 0.559$

For $\dot{X}_1$

$\dot{V}_{-2} = \dot{X}_1 = 1$
$\dot{V}_{-1} = \dot{X}_2 = 0$
$\dot{V}_0 = \dot{X}_3 = 0$
$\dot{V}_1 = \dot{V}_{-2} V_{-1} + V_{-2} \dot{V}_{-1} = (1)(2) + (-1)(0) = 2$
$\dot{V}_2 = \dot{V}_{-1} V_0 + V_{-1} \dot{V}_0 = (0)(3) + (2)(0) = 0$
$\dot{V}_3 = \dot{V}_2 \cos(V_2) = (0)(1) = 0$
$\dot{V}_4 = \dot{V}_1 V_3 + \dot{V}_3 V_1 = (2)(-0.279) + (0)(-2) = \boxed{-0.559}$

For $\dot{X}_2$

$\dot{V}_{-2} = 0$
$\dot{V}_{-1} = 1$
$\dot{V}_0 = 0$
$\dot{V}_1 = (0)(2) + (-1)(1) = -1$
$\dot{V}_2 = (1)(3) + (2)(0) = 3$
$\dot{V}_3 = (3)\cos(6) = 2.88$
$\dot{V}_4 = (-1)(-0.279) + (2.88)(-2)$
$= \boxed{-5.48}$

For $\dot{X}_3$

$\dot{V}_{-2} = 0$
$\dot{V}_{-1} = 0$
$\dot{V}_0 = 1$
$\dot{V}_1 = (0)(2) + (-1)(6) = 0$
$\dot{V}_2 = (0)(3) + (2)(1) = 2$
$\dot{V}_3 = (2)\cos(6) = 1.92$
$\dot{V}_4 = (0)(-0.279) + (1.92)(-2) = \boxed{-3.84}$

$$\boxed{\frac{\partial f}{\partial X_1} = -0.559 \quad \frac{\partial f}{\partial X_2} = -5.48 \quad \frac{\partial f}{\partial X_3} = -3.84}$$

c) $\bar{V}_4 = 1$

$\bar{V}_3 = \bar{V}_4 \frac{\partial V_4}{\partial V_3} = \bar{V}_4 V_1 = (1)(-2) = -2$

$\bar{V}_2 = \bar{V}_3 \frac{\partial V_3}{\partial V_2} = \bar{V}_3 \cos(V_2) = (-2)\cos(6) = -1.92$

$\bar{V}_1 = \bar{V}_4 \frac{\partial V_4}{\partial V_1} = \bar{V}_4 V_3 = (1)(-0.279) = -0.279$

$\bar{V}_0 = \bar{V}_2 \frac{\partial V_2}{\partial V_0} = \bar{V}_2 V_{-1} = (-1.92)(2) = -3.84$

$\bar{V}_{-1} = \bar{V}_2 \frac{\partial V_2}{\partial V_{-1}} + \bar{V}_1 \frac{\partial V_1}{\partial V_{-1}} = \bar{V}_2 V_0 + \bar{V}_1 V_{-2} = (-1.92)(3) + (-0.279)(-1) = -5.48$

$\bar{V}_{-2} = \bar{V}_1 \frac{\partial V_1}{\partial V_{-2}} = \bar{V}_1 V_{-1} = (-0.279)(2) = 0.559$

$$\boxed{\frac{\partial f}{\partial x_1} = -0.559 \quad \frac{\partial f}{\partial x_2} = -5.48 \quad \frac{\partial f}{\partial x_3} = -3.84}$$

d) $$\frac{\partial f}{\partial x_1} \approx \frac{1}{0.002}\left\{(-1+0.001)(2)\sin[(2)(3)] + (-1-0.001)(2)\sin[(2)(3)]\right\}$$

$$= \boxed{-0.559}$$

$$\frac{\partial f}{\partial x_2} \approx \frac{1}{0.002}\left\{(-1)(2+0.001)\sin[(2+0.001)(3)] - (-1)(2-0.001)\sin[(2-0.001)(3)]\right\}$$

$$= \boxed{-5.48}$$

$$\frac{\partial f}{\partial x_3} \approx \frac{1}{0.002}\left\{(-1)(2)\sin[(2)(3+0.001)] - (-1)(2)\sin[(2)(3-0.001)]\right\}$$

$$= \boxed{-3.84}$$

Problem 1:

```python
import torch as t

x1 = t.tensor([-1.0], requires_grad=True).float()
x2 = t.tensor([2.0], requires_grad=True).float()
x3 = t.tensor([3.0], requires_grad=True).float()


def f(a, b, c):
    return a*b*t.sin(b*c)


F = f(x1, x2, x3)
F.backward()

print("Autograd")
print(f"df/dx1: {x1.grad}")
print(f"df/dx2: {x2.grad}")
print(f"df/dx3: {x3.grad}")

h = t.tensor([0.001], requires_grad=False).float()

dx1 = (f(x1 + h, x2, x3) - f(x1 - h, x2, x3)) / (2*h)
dx2 = (f(x1, x2 + h, x3) - f(x1, x2 - h, x3)) / (2*h)
dx3 = (f(x1, x2, x3 + h) - f(x1, x2, x3 - h)) / (2*h)

print("\nCenter Finite Difference")
print(f"df/dx1: {dx1}")
print(f"df/dx2: {dx2}")
print(f"df/dx3: {dx3}")

```

```
Autograd
df/dx1: tensor([-0.5588])
df/dx2: tensor([-5.4816])
df/dx3: tensor([-3.8407])

Center Finite Difference
df/dx1: tensor([-0.5589], grad_fn=<DivBackward0>)
df/dx2: tensor([-5.4817], grad_fn=<DivBackward0>)
df/dx3: tensor([-3.8404], grad_fn=<DivBackward0>)

Process finished with exit code 0
```

Problem 2:

```python
from keras.models import Sequential
from keras.layers import Dense
from numpy.random import rand, randn
import numpy as np


def def_discriminator(n_inputs=2):
    model = Sequential()
    model.add(Dense(25, activation='relu', kernel_initializer='he_uniform', input_dim=n_inputs))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model


def def_generator(latent_dim, n_outputs=2):
    model = Sequential()
    model.add(Dense(15, activation='relu', kernel_initializer='he_uniform', input_dim=latent_dim))
    model.add(Dense(n_outputs, activation='linear'))
    return model


def def_gan(gen, disc):
    model = Sequential()
    model.add(gen)
    model.add(disc)
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

```

```python
def generate_real_samples(n):
    u = 0
    s = np.sqrt(0.2)

    X1 = 2*rand(n) - 1
    X2 = 1/(s*np.sqrt(2*np.pi)) * np.exp(-0.5 * (X1-u)**2 / s**2)

    X1 = X1.reshape(n, 1)
    X2 = X2.reshape(n, 1)
    X = np.hstack((X1, X2))

    y = np.ones((n, 1))
    return X, y


def generate_fake_samples(n):
    X1 = 2*rand(n) - 1
    X2 = 2*rand(n) - 1

    X1 = X1.reshape(n, 1)
    X2 = X2.reshape(n, 1)
    X = np.hstack((X1, X2))

    y = np.zeros((n, 1))
    return X, y


def generate_latent_points(latent_dim, n):
    x_input = randn(latent_dim * n)
    x_input = x_input.reshape(n, latent_dim)
    return x_input

```

```python

def train(d_model, gan_model, latent_dim, n_epochs=100, n_batch=128):
    half_batch = int(n_batch / 2)

    for i in range(n_epochs):
        x_real, y_real = generate_real_samples(half_batch)
        x_fake, y_fake = generate_fake_samples(half_batch)

        d_model.trainable = True
        d_model.train_on_batch(x_real, y_real)
        d_model.train_on_batch(x_fake, y_fake)

        x_gan = generate_latent_points(latent_dim, n_batch)
        y_gan = np.ones((n_batch, 1))

        d_model.trainable = False
        acc = gan_model.train_on_batch(x_gan, y_gan)
        print(f'Accuracy: {acc[1]:0.4f}')


discriminator = def_discriminator()
generator = def_generator(1)
gan = def_gan(generator, discriminator)

train(discriminator, gan, 1)

X, y_real = generate_real_samples(30)

```