Corrigé

Résolution numérique d'une équation différentielle

Exercice 1

a) On définit la fonction

```
def euler(f, x0, t):
    x = [x0] * len(t)
    for k in range(len(t)-1):
        x[k+1] = x[k] + (t[k+1] - t[k]) * f(x[k], t[k])
    return x
```

b) On définit la fonction

```
def euler(f, x0, t0, T, n):
    h = T / n
    x = [x0] * (n + 1)
    for k in range(n):
        x[k+1] = x[k] + h * f(x[k], t[k])
    return x
```

Exercice 2

```
def heun(f, x0, t):
    x = [x0] * len(t)
    for k in range(len(t)-1):
        a = x[k] + (t[k+1] - t[k]) * f(x[k], t[k])
        x[k+1] = x[k] + (t[k+1] - t[k]) * (f(x[k], t[k]) + f(a, t[k+1])) / 2
    return x
```

Exercice 3

```
def euler2(f, x0, y0, t):
    x = [x0] * len(t)
    y = [y0] * len(t)
    for k in range(len(t)-1):
        h = t[k+1] - t[k]
        x[k+1] = x[k] + h * y[k]
        y[k+1] = y[k] + h * f(x[k], y[k], t[k])
    return x, y
```

Exercice 4

```
def adaptatif(f, x0, t0, T, h, epsilon):
    T, X = [t0], [x_0]
    t, x = t0, x0
    while t < t0 + T:
        p1 = f(x, t)
        while True:
            p2 = f(x + p * h / 2, t + h / 2)
            e = h / 2 * abs(p2 - p1)
            if e < epsilon:
                break
            h *= .9 * np.sqrt(epsilon / e)
        t += h
        T.append(t)
        x += p2 * h
        X.append(x)
        h *= .9 * np.sqrt(epsilon / e)
    return T, X
```

Exercice 5

```
def verlet(f, x0, y0, t):
    x = [x0] * len(t)
    y = [y0] * len(t)
    for k in range(len(t)-1):
        h = t[k+1] - t[k]
        x[k+1] = x[k] + h * y[k] + h**2 * f(x[k]) / 2
        y[k+1] = y[k] + h * (f(x[k]) + f(x[k+1])) / 2
    return x, y
```