CORRIGÉ MODÉLISATION TPC 2015

```python
import matplotlib.pyplot as plt
import numpy as np
import sys

def calc_norme(v):
    N = len(v)
    s = 0
    for i in range(N):
        s += v[i] ** 2
    return np.sqrt(s)

def schema_explicite(T0, e, alpha, dt , Tint, Text):
    N = len(T0)
    dx = e / (N+1)
    r = dt / (alpha * dx ** 2)
    if r >= 0.5:
        print("Valeur de r incoprrecte. Stop")
        sys.exit(0)
    UnMoins2r = 1 - 2 * r

    T_tous_k = np.zeros( (N, ItMax) )
    T_tous_k[:,0] = T0

    T_tous_k[0,1] = r*Tint + UnMoins2r*T_tous_k[0,0]+r*T_tous_k[1,0]
    T_tous_k[N-1,1] = r*T_tous_k[N-2,0] + UnMoins2r*T_tous_k[N-1,0]+r*Text
    for i in range(2,N):
        T_tous_k[i-1,1]=r*T_tous_k[i-2,0] + UnMoins2r*T_tous_k[i-1,0]+r*T_tous_k[i,0]

    k = 1
    while calc_norme(T_tous_k[:,k]-T_tous_k[:,k-1]) >= 0.005 and k < ItMax - 1:
        k += 1
        T_tous_k[0,k] = r*Tint + UnMoins2r*T_tous_k[0,k-1]+r*T_tous_k[1,k-1]
        T_tous_k[N-1,k] = r*T_tous_k[N-2,k-1] + UnMoins2r*T_tous_k[N-1,k-1]+r*Text
        for i in range(2,N):
            T_tous_k[i-1,k]=r*T_tous_k[i-2,k-1] + UnMoins2r*T_tous_k[i-1,k-1]+r*T_tous_k[i,k-1]

    nbIter = k
    return nbIter, T_tous_k

def CalcTkp1(M, d):
    N = len(d)
    cprime = np.zeros( (N,1) )
    dprime = np.zeros( (N,1) )
    u = np.zeros( (N,1) )
    cprime[0] = M[0,1] / M[0,0]
    dprime[0] = d[0] / M[0,0]
    for i in range(1, N-1):
        denom = M[i,i] - M[i,i-1] * cprime[i-1]
        cprime[i] = M[i,i+1] / denom
        dprime[i] = (d[i] - M[i,i-1] * dprime[i-1]) / denom
    u[N-1] = (d[N-1] - M[N-1,N-2]*dprime[N-2])/(M[N-1,N-1] - M[N-1,N-2]*cprime[N-2])
    for i in range(N-2, -1, -1):
        u[i] = dprime[i] - cprime[i] * u[i+1]
    return u

def schema_implicite(T0, e, alpha, dt , Tint, Text):
    N = len(T0)
```

```
    dx = e / (N+1)
    r = dt / (alpha * dx ** 2)
    UnPlus2r = 1 + 2 * r

    T_tous_k = np.zeros( (N, ItMax) )
    T_tous_k[:,0] = T0
    M = np.zeros( (N,N) )
    # une grosse matrice avec plein de zéros, quel gâchis!
    for i in range(N):
        M[i,i] = UnPlus2r
        if i != 0:
            M[i,i-1] = -r
            M[i-1,i] = -r
    v = np.zeros( (N,1) )
    v[0] = Tint
    v[N-1] = Text

    T_tous_k[:,1] = CalcTkp1(M, T_tous_k[:,0] + r*v[:,0])[:,0]

    k = 1
    while calc_norme(T_tous_k[:,k]-T_tous_k[:,k-1]) >= 0.005 and k < ItMax - 1:
        k += 1
        T_tous_k[:,k] = CalcTkp1(M, T_tous_k[:,k-1] + r*v[:,0])[:,0]
    nbIter = k
    return nbIter, T_tous_k

# programme principal

# Constantes
epaisseur = 0.4
conductivite = 1.65
rho = 2150
cp = 1000
Tint = 20
Text1 = 10
Text2 = -20
N = 200
Delta_t = 2

# on calcule a et b
b = Tint
a = (Text1 - Tint) / epaisseur

# Les initialisations
ItMax = 30000
Delta_x = epaisseur / (N+1)
x = [i * Delta_x for i in range(1,N+1)]
T0 = [ a*x[i]+b for i in range(N)]
alpha = rho * cp / conductivite

ch = int(input("Votre choix: schéma explicite=0, schéma implicite =1 : "))
if ch == 0:
    nbIter, T_tous_k = schema_explicite(T0, epaisseur, alpha, Delta_t , Tint, Text2)
else:
    nbIter, T_tous_k = schema_implicite(T0, epaisseur, alpha, Delta_t , Tint, Text2)

#print("nbIter = ", nbIter)
if nbIter == ItMax - 1:
    plt.title("Régime permanent pas atteint, augmentez ItMAx")
else:
    temps = nbIter * Delta_t
```

```
        plt.title("Régime permanent atteint au bour de "+str(int(100*temps/60)/100)+" heures.")
    if ch == 0:
        plt.text(0, 0, "Schéma explicite. "+str(nbIter)+" itérations")
    else:
        plt.text(0, 0, "Schéma implicite. "+str(nbIter)+" itérations")
    plt.text(0, -10, "Tint="+str(Tint)+" Text1="+str(Text1)+" Text2="+str(Text2)+" Epaisseur="+str(epai
    for k in range((nbIter+1)//500):
        plt.plot(x,T_tous_k[:,k])

    plt.xlabel('Abscisse')
    plt.ylabel('Temperature')
    plt.show()
```