Víctor Torres Varas 201173070-3

Declaración de Trabajo: Juro o Prometo que la totalidad.

Pregunta 3.

$U_t(x,t) = U_{xx}(t) + U(x,t) - U^3(x,t)$ $\quad U(-1,t) = -1$

$U(1,t) = 1$

$U(x,0) = \frac{1}{100}\left(53x + 47\sin\left(-\frac{3\pi}{2}x\right)\right)$

Para $t=0$, la condición $-1 = x$

$U(-1,0) = \frac{1}{100}\left(53\cdot(-1) + 47\cdot\sin\left(\frac{3\pi}{2}\right)\right) = -1$

$\frac{-53 + 47\sin(3\pi/2)}{100} = -1$

$\frac{\overset{-100}{-53 + 47(-1)}}{100} = -1$ ✓✓

$x = 1,\ t = 0.$

$U(1,t) = 1$

$U(1,0) = \frac{1}{100}\left(53 + 47\sin\left(\cancel{-3\pi/2}\right)\right)$

$\frac{1}{100}(53 + 47\cdot 1) = 1$

$\frac{100}{100} = 1$ ✓✓

(a) Discretización:

$U_{xx} = \frac{U_{x+1,t} - 2U_{x,t} + U_{x-1,t}}{\Delta x^2}$, $\quad U(x,t) = U_{x,t}$ ; $U_t = \frac{U_{x,t+1} - U_{x,t}}{\Delta t}$

$U_t(x,t) = \frac{U_{x+1,t} - 2U_{x,t} + U_{x-1,t}}{\Delta x^2} + U_{x,t} - U^3_{x,t}$

$U_{-1,t} = -1$

$U_{1,t} = 1$

$\frac{U_{x,t+1} - U_{x,t}}{\Delta t} = \frac{U_{x+1,t} - 2U_{x,t} + U_{x-1,t}}{\Delta x^2} + U_{x,t} - U^3_{x,t}$

$U_{x,t+1} - U_{x,t} = \frac{\Delta t}{\Delta x^2}\cdot\left(U_{x+1,t} - 2U_{x,t} + U_{x-1,t}\right) + \Delta t\left(U_{x,t} - U^3_{x,t}\right)$

$U_{x,t+1} = \frac{\Delta t}{\Delta x^2}\cdot\left(U_{x+1,t} - 2U_{x,t} + U_{x-1,t}\right) + \Delta t\left(U_{x,t} - U^3_{x,t}\right) + U_{x,t}$

Algoritmo:

Con la discretización podemos usar un método iterativo el cual nos permite conocer los $w_{x,t}$, esto dado que conocemos las condiciones iniciales y de Borde

```
def Algoritmo(Nx, Nt)
    dx = 1/Nx
    dt = 1/Nt
    dx2 = dx^2
    resultado = np.zeros([num_pasos, Nx, Ny])
    dt2 = dt^2
    u  # Se define U que contiene los datos, de Tamaño [0,80] con T
    for n in range(num_pasos):
        u = esolver(u, u_AUX, dt, dx2, dy2)
        u_aux = u
        resultado [n, :, :] = u

    return resultado


def esolver(u, u2, dt, dx2, dy2)
    u[1:-1, 1:-1] = u2[1:-1, 1:-1]
    u[1:-1, 1:-1] += (dt/dx2)*(u2[2:, 1:-1] - 2 u2[1:-1, 1:-1] + u2[:-2, 1:-1])
    u(1:-1, 1:-1] += dt*(u2[1:-1, 1:-1] - (u2[1:-1, 1:-1]**3))
    u[1:-1, 1:-1] += u2[1:-1, 1:-1]
  return U
```