

Programación: mostrar el campo de las pendientes

Objetivos. Programar una función que muestre el campo de las pendientes. Más precisamente, dada una función f de dos argumentos, mostrar en forma normalizada los vectores

$$(1, f(t, x))$$

para cada t y cada x de las listas dadas.

1. Lenguaje de programación. Este texto está orientado al lenguaje Matlab/Octave. Se pueden hacer cosas similares en Python+numpy, Sage, Julia y R.

2. Función meshgrid. Ejecute el siguiente comando para entender qué hace la función meshgrid:

```
[X, Y] = meshgrid(1 : 5, 31 : 34)
```

3. Función quiver. Ejecutamos los siguientes comandos para entender qué hace la función quiver:

```
pt = 0.1 * ones(2, 3)
px = [0.1, 0.2, 0.3; -0.1, -0.2, -0.3];
quiver(1 : 3, 1 : 2, pt, px);
```

4. Programamos la función plotslopefield.

```
function [] = plotslopefield(f, tvalues, xvalues),
    [tmatrix, xmatrix] = meshgrid(tvalues, xvalues);
    dt = tvalues(2) - tvalues(1);
    dx = xvalues(2) - xvalues(1);
    tgridmin = tvalues(1) - dt / 2; tgridmax = tvalues(end) + dt / 2;
    xgridmin = xvalues(1) - dx / 2; xgridmax = xvalues(end) + dx / 2;
    pmatrix = f(tmatrix, xmatrix);
    onesmatrixscaled = ones(size(tmatrix)) / dt;
    pmatrixscaled = abs(pmatrix) / dx;
    s = 0.3 ./ sqrt(onesmatrixscaled .^ 2 + pmatrixscaled .^ 2);
    quiver(tvalues, xvalues, s, s .* pmatrix, 0, '.b'); hold on;
    quiver(tvalues, xvalues, -s, -s .* pmatrix, 0, '.b');
    axis([tgridmin, tgridmax, xgridmin, xgridmax]);
end
```

5. Prueba. Suponemos que `f1` está definida mediante el siguiente código:

```
function [p] = f1(t, x),  
    p = 2 * x ./ (3 + t);  
end
```

Entonces podemos dibujar el campo correspondiente con el siguiente comando:

```
plotslopefield(@f1, -2 : 0.5 : 2, -2 : 0.5 : 2)
```

6. Otra prueba.

```
f2 = @(t, x) - t ./ x;  
plotslopefield(f2, -0.9 : 0.2 : 0.9, -0.9 : 0.2 : 0.9);
```