Programming in Python[1]

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

Academic year 2020/21, II semester

Lecture XIV: Laplacian operator

Gray-Scott systems

Systems driven by the Gray-Scott equation exhibit Turing patterns (D_u, D_v, f, k are constants).

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + f \cdot (1 - u)$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (f + k) \cdot v$$

- These give the change of u and v over time
- The diffusion term can be approximated on a grid by computing the discrete Laplacian

Discrete Laplacian

$\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

- Change on a grid (1-D):

$$\nabla f[n] = f[n+1] - f[n]$$
$$\nabla f[n] = f[n] - f[n-1]$$

- Second order change (1-D):

$$\begin{aligned}\nabla(\nabla f[n]) &= \nabla(f[n+1]) - \nabla(f[n]) \\ &= (f[n+1] - f[n]) - (f[n] - f[n-1]) \\ &= f[n-1] - 2f[n] + f[n+1]\end{aligned}$$

- In 2-D we do this independently on the 2 dimensions n, m:

$$\begin{aligned}\nabla(\nabla f[n,m]) &= f[n-1,m] - 2f[n,m] + f[n+1,m] + \\ &\quad f[n,m-1] - 2f[n,m] + f[n,m+1] \\ &= f[n-1,m] + f[n+1,m] + f[n,m-1] + f[n,m+1] - 4f[n,m]\end{aligned}$$

Vectorization

0	0	0	0	0	0
0	13	14	15	16	0
0	9	10	11	12	0
0	5	6	7	8	0
0	1	2	3	4	0
0	0	0	0	0	0

`X[1:-1, 2:]`

Ignoring the border, the right neighbour of (i, j) is $(i, j + 1)$ in the inner white and (i, j) in the yellow: in the inner white 11 is $(1, 3)$, its neighbour 12 is $(1, 4)$, but $(1, 3)$ in the yellow.

This way one can compute the Laplacian matrix using only vectorized plus.