Themodynamique
Corrigé Série Supplémentaire 3:
Patterns de Turing

S. Guinchard*
Section de Physique, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Suisse
(Supervised by Prof. J.P. Ansermet)†
(Dated: May 25, 2022)

I. EXERCICE 1: PATTERNS DE TURING DANS LE CAS DE DEUX MODÈLES DE RÉACTION-DIFFUSION

A. Implémentation du problème

[1]:
```
##############################
# Written by S.Guinchard on  #
#         05/25/22           #
##############################
import numpy as np
import matplotlib.pyplot as plt
#%matplotlib inline

######################################
#     PARAMETERS FITZHUGH-NAGUMO     #
######################################

a = 2.8e-4
b = 5e-3
tau = .1
k = -.005

#####################################
#   PARAMETERS DIFFUSION REACTION   #
#####################################

CU = 5.0e-3
DU = .1
CV = 2.8e-4
DV = .1

#####################################

size = 100  # size of the 2D grid
dx = 2. / size  # space step
T = 9.0  # total time
dt = .001  # time step
n = int(T / dt)  # number of iterations

U = np.random.rand(size, size)
V = np.random.rand(size, size)

def laplacian(Z):
    Ztop = Z[0:-2, 1:-1]
```

* salomon.guinchard@epfl.ch
† Laboratoire de Physique des Matériaux Nanostructurés, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Suisse

```
    Zleft = Z[1:-1, 0:-2]
    Zbottom = Z[2:, 1:-1]
    Zright = Z[1:-1, 2:]
    Zcenter = Z[1:-1, 1:-1]
    return (Ztop + Zleft + Zbottom + Zright -
            4 * Zcenter) / dx**2

def show_patterns(U, ax=None):
    ax.imshow(U, cmap=plt.cm.copper,
              interpolation='bilinear',
              extent=[-1, 1, -1, 1])
    ax.set_axis_off()

fig, axes = plt.subplots(3, 3, figsize=(8, 8))
step_plot = n // 9
# We simulate the PDE with the finite difference
# method.
for i in range(n):
    # We compute the Laplacian of u and v.
    deltaU = laplacian(U)
    deltaV = laplacian(V)
    # We take the values of u and v inside the grid.
    Uc = U[1:-1, 1:-1]
    Vc = V[1:-1, 1:-1]
    # We update the variables.

###### TODO: COMMENT OR UNCOMMENT TO CHOOSE THE SET OF EQUATIONS YOU WANT ######

    ##### 1ST SET : FITZHUGH_NAGUMO EQUATION ######
    U[1:-1, 1:-1], V[1:-1, 1:-1] = \
         Uc + dt * (a * deltaU + Uc - Uc**3 - Vc + k),\
         Vc + dt * (b * deltaV + Uc - Vc) / tau

    ##### 2ND SET:  REACTION DIFFUSION MODEL ######
    #U[1:-1, 1:-1], V[1:-1, 1:-1] = \
    #   Uc + dt * ((CU * Uc*Uc) / Vc - Uc + DU * deltaU),\
    #    Vc + dt * (CV * Uc*Uc - Vc + DV * deltaV)

    # Neumann conditions: derivatives at the edges
    # are null.
    for Z in (U, V):
        Z[0, :] = Z[1, :]
        Z[-1, :] = Z[-2, :]
        Z[:, 0] = Z[:, 1]
        Z[:, -1] = Z[:, -2]

    # We plot the state of the system at
    # 9 different times.
    if i % step_plot == 0 and i < 9 * step_plot:
        ax = axes.flat[i // step_plot]
        show_patterns(U, ax=ax)
        ax.set_title(f'$t={i * dt:.2f}$')


fig, ax = plt.subplots(1, 1, figsize=(8, 8))
show_patterns(U, ax=ax)
plt.show
```

[1]: <function matplotlib.pyplot.show(close=None, block=None)>

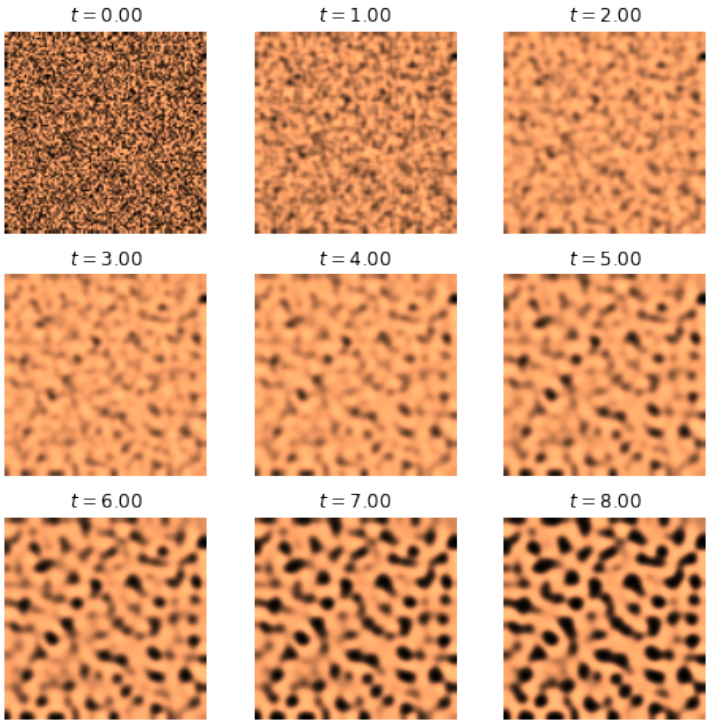

FIG. 1. Grid with the two substances interacting for 9 different times (Fitzhugh - Nagumo)

FIG. 2. Final distribution of the two substances on the 2D grid: leopard motive from Fitzhugh - Nagumo

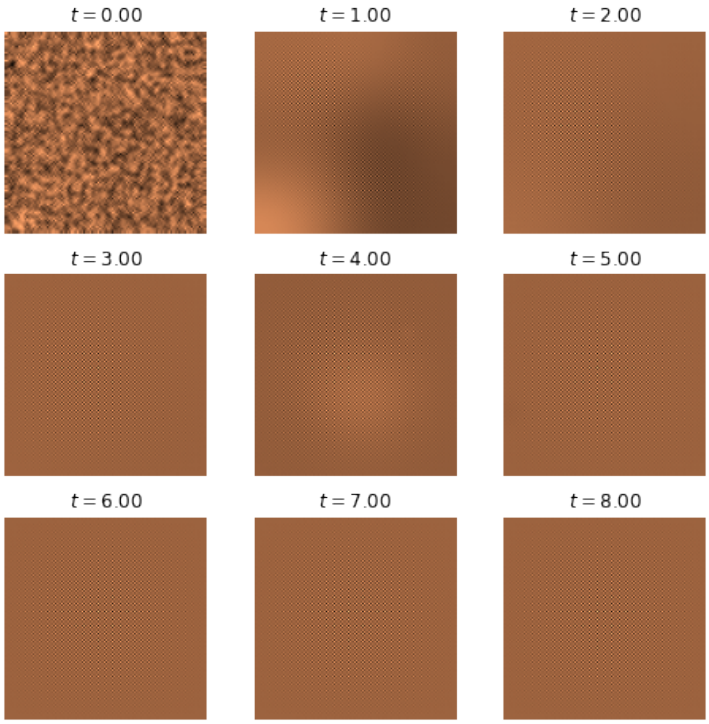

FIG. 3. Grid with the two substances interacting for 9 different times (Reaction - Diffusion model)

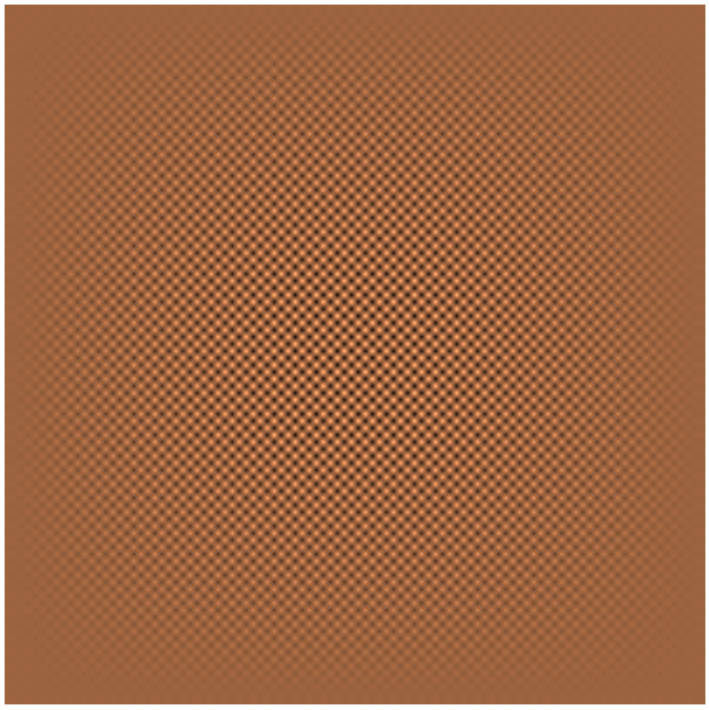

FIG. 4. Final distribution of the two substances on the 2D grid: checkerboard (Reaction - Diffusion model)