Group: Amalia Bogri, Christian Berrig & Jonas Bolduan

Spatial models: advection, rection, diffusion

Spatial model: model parameters

Parameter	Unit	Value
Concentration of phytoplankton $C(x,t)$	µM	C_o= 3
Time, t	s	0-300
Depth, L	m	100
Diffusion, D	$m^2 s^{-1}$	0.1
Advection velocity, v	$m\ s^{-1}$	0.1
Maximum growth rate, r_{max}	s^{-1}	0.01
Extension:		
Growth rate: $r_{max}e^{-x/H} - C/K$		
Damping coefficient, H	m^{-1}	0.2
Carrying capacity, K	µM	10

Equations:

General:

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D(t,x)\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial v(t,x)C(t,x)}{\partial x} + r(t,x)C(t,x)$$

1

Constant D, v, r:

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial vC(t,x)}{\partial x} + rC(t,x)$$

2

Constant D, v

Growth rate:

- logistic growth

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial vC(t,x)}{\partial x} + r_{max}\left(1 - \frac{C}{K}\right)C(t,x)$$

3

Constant D, v

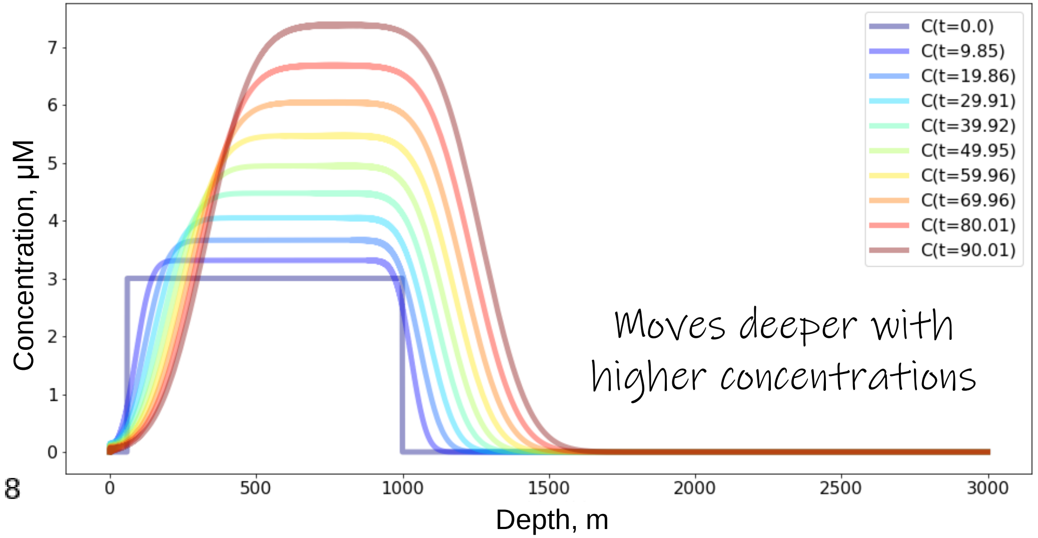
Growth rate:

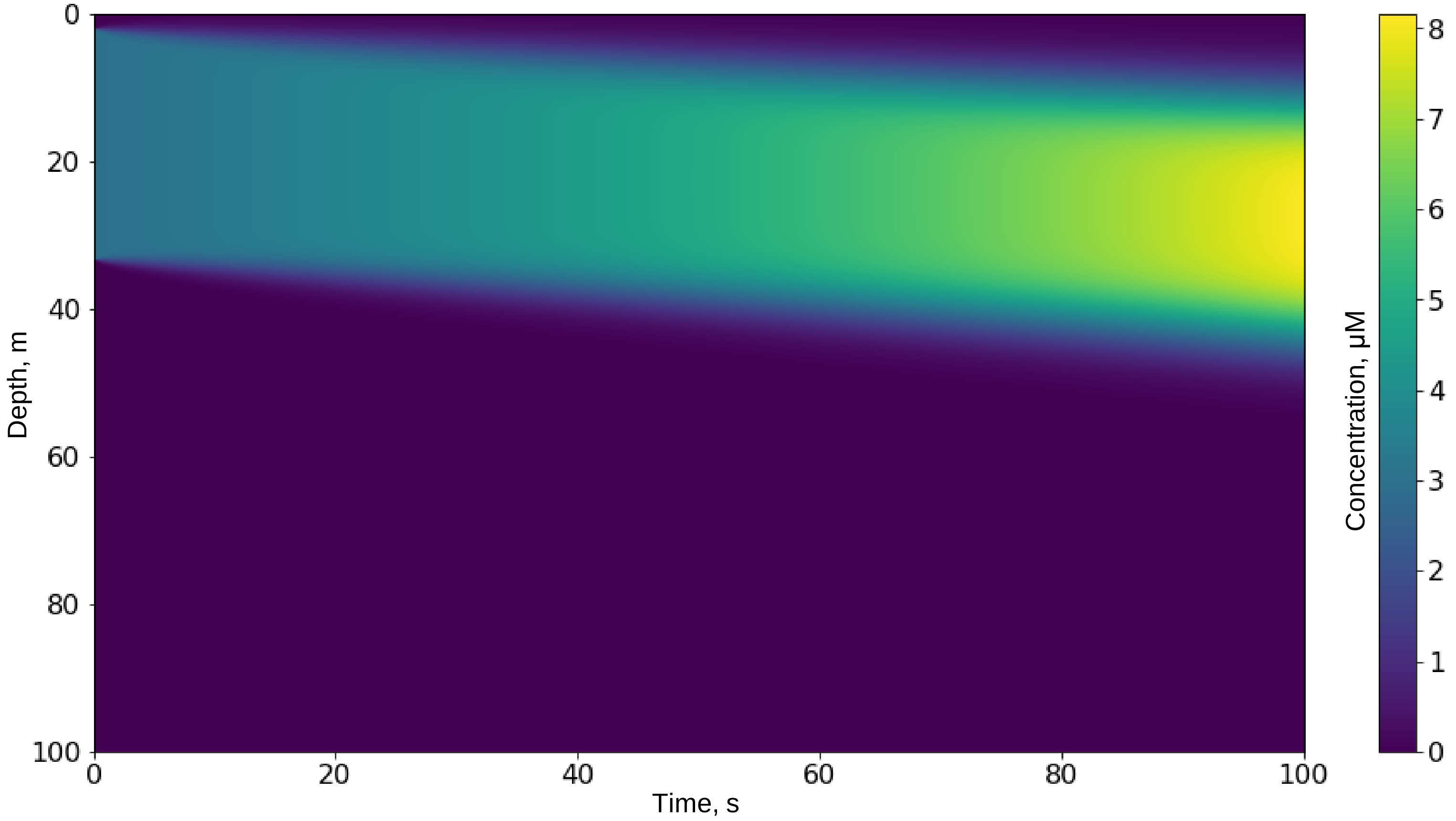
- declines with light &
- logistic growth

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial vC(t,x)}{\partial x} + r_{max}\left(e^{-\frac{x}{H}} - \frac{C}{K}\right)C(t,x)$$

Case 1: Constant D, v, r

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial vC(t,x)}{\partial x} + rC(t,x)$$





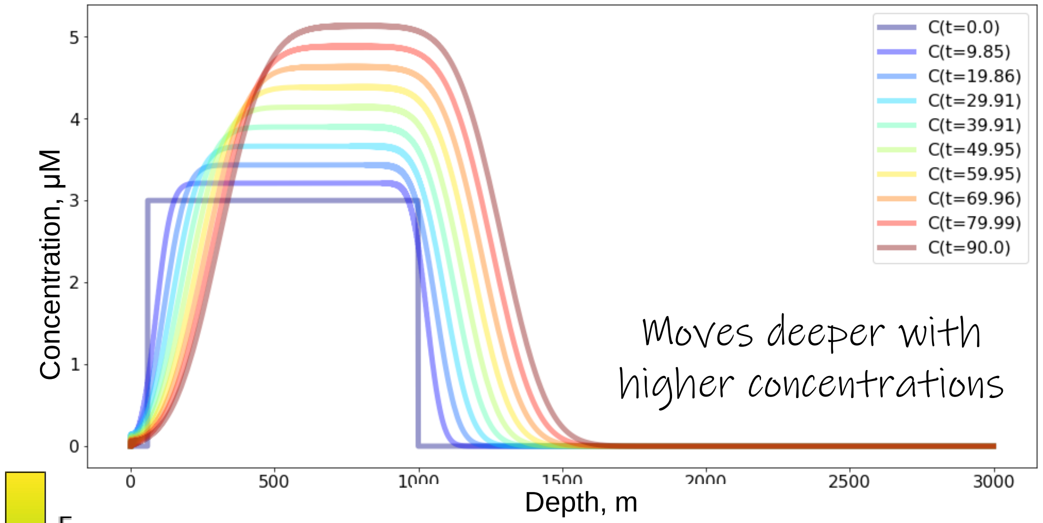
Because of the constant diffusion, the graph becomes more ‘blurry’ as time passes.

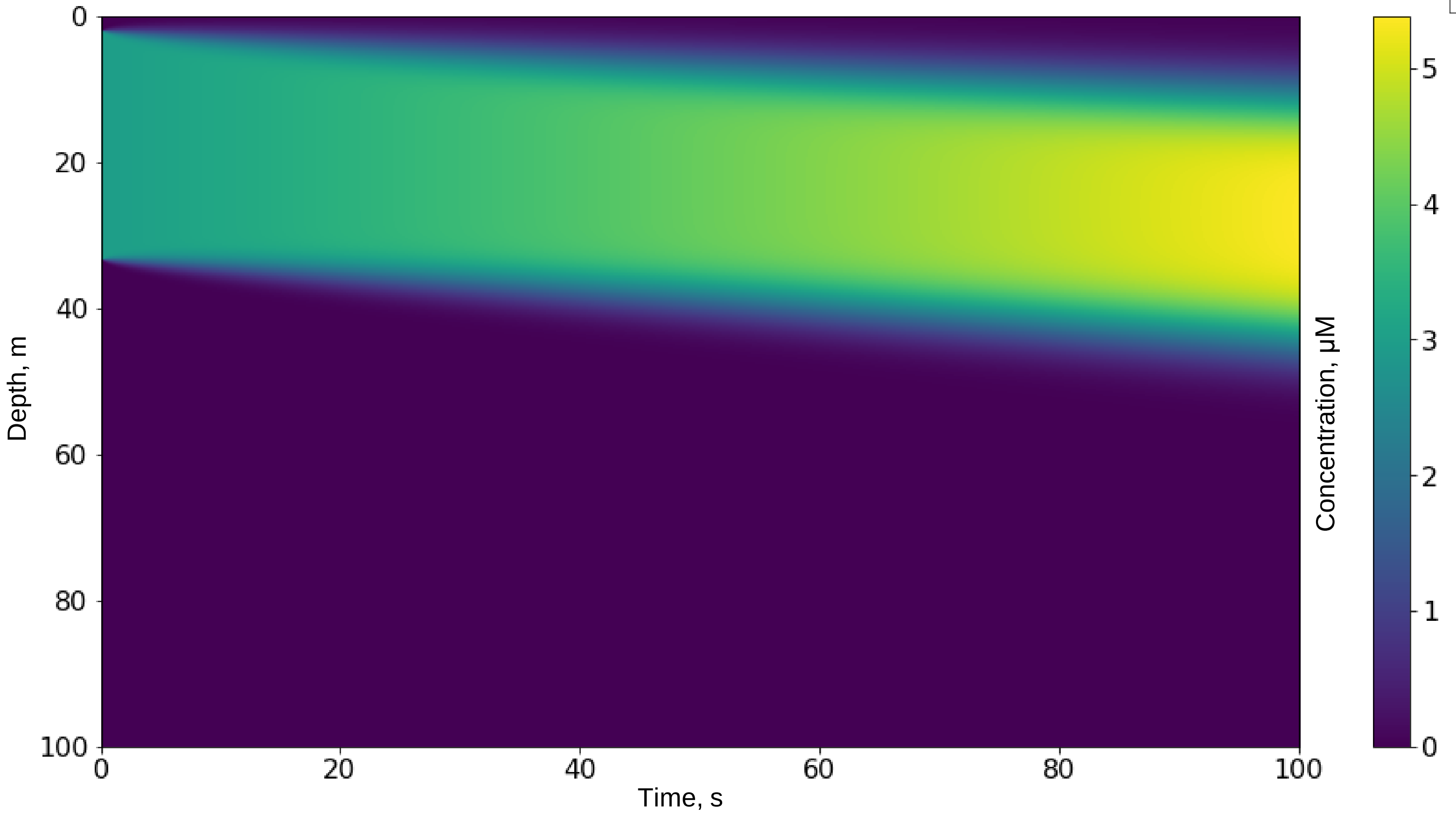
Because of the constant advection towards a deeper depth, the concentration of phytoplankton moves downwards in time.

Because of the exponential growth, the concentration of phytoplankton increases in time (more yellow).

Case 2: Constant D, v, Growth rate depends on C

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial vC(t,x)}{\partial x} + r_{max}\left(1 - \frac{C}{K}\right)C(t,x)$$





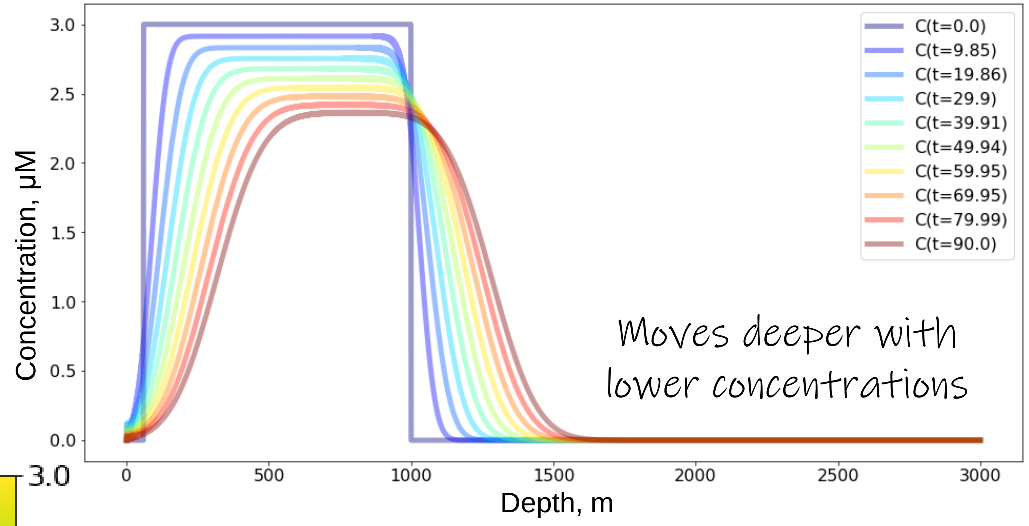
Because of the constant diffusion, the graph becomes more ‘blurry’ as time passes.

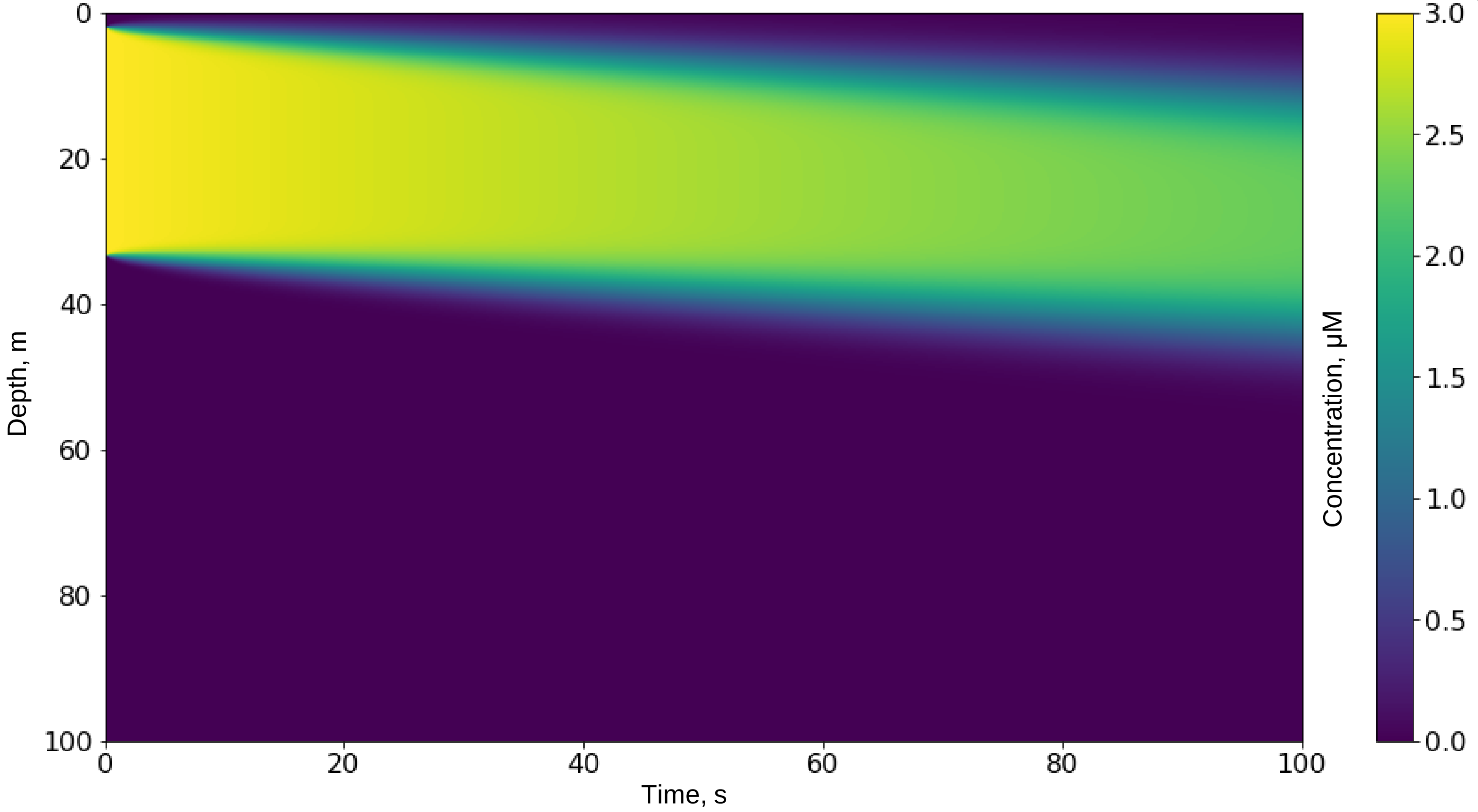
Because of the constant advection towards a deeper depth, the concentration of phytoplankton moves downwards in time.

Because of the logistic growth,
the concentration of phytoplankton increases in time, yet it is limited by the carrying capacity.

Case 3: Constant D, v, Growth rate declines with light & logistic growth

$$\frac{\partial C(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C(t,x)}{\partial x}\right) - \frac{\partial vC(t,x)}{\partial x} + r_{max}\left(e^{-\frac{x}{H}} - \frac{C}{K}\right)C(t,x)$$





Because of the constant diffusion, the graph becomes more ‘blurry’ as time passes.

Because of the constant advection towards a deeper depth, the concentration of phytoplankton moves downwards in time.

Because of the growth rate that is exponentially decaying with depth & is depended on C, the concentration of phytoplankton decreases the more it sinks by advection, in time.

Python code

```python
# Import stuff:
from scipy.integrate import odeint, solve_ivp
import matplotlib.pyplot as plt
import numpy as np

# This is for reasonable fontsize universally defined:
fs_label = 16
parameters = {
                'figure.titlesize': fs_label+6,
                'axes.labelsize': fs_label,
                'axes.titlesize': fs_label+4,
                'xtick.labelsize': fs_label,
                'ytick.labelsize': fs_label,
                'legend.fontsize': fs_label,
                'lines.linewidth': 5
             }
plt.rcParams.update(parameters)

# Parameters:
params_dict = dict(
    L = 100, #100
    N = 3000,
    D_const = 0.1, #0.0001
    v_const = 0.1 , #0.0001
    r_const = 0.01, ##0.0001
    H = 0.2, #2
    K = 10 #10
)

for k, v in params_dict.items():
    assign_str = f"{k} = {v}"
    exec(assign_str)
    print(assign_str)

ddx = lambda arr, delta: np.array([arr[0]] + list((np.diff(arr[:-1]) +
np.diff(arr[1:]))/(2*delta)) + [arr[-1]])



#Simple funtions defining diffusion, advection and reaction
D = lambda x, t: D_const
v = lambda x, t: v_const
r = lambda x, t: r_const
```

```python
# Flux:
J = lambda c, x, t, delta: c*v(x, t) - D(x, t)*ddx(c, delta)

# Intrinsic growth
f = lambda c, x, t: r(x, t)*c*(np.exp(- x/H) - c/K)  # case 1
# Case 2: f = lambda c, x, t: r(x, t)*c*(1 - c/K)
# Case 3: f = lambda c, x, t: r(x, t)*c*(np.exp(- x/H) - c/K)

def DAR(t, state, *params):
    for key, val in params:
        assign_str = f"{key} = {val}"
        exec(assign_str)
    deltax = L/N
    x_list = np.arange(0, L, deltax)
    C = state
    dCdt = -ddx(J(C, x_list, t, deltax), deltax) + f(C, x_list, t)
    return np.array(dCdt)


C_init = np.zeros(N)
C_init[int(N*1/50):int(N*2/6)] = 3
params = tuple(params_dict.items())

num_sol = solve_ivp(DAR, (0, 100), C_init, args=params)



ns = num_sol.y
t_list = num_sol.t

fig, ax = plt.subplots(figsize=(16,8))
im = ax.imshow(ns, aspect="auto", extent=[t_list[0], t_list[-1], L, 0])
ax.set_xlabel("$t$")
ax.set_ylabel("$x$")
fig.colorbar(im,)

filename = "DAR_heatmap"
plt.show()
print(params_dict.items())


fig, ax = plt.subplots(figsize=(16,8))
num_levels = 10
levels = [int(k*len(ns.T)/num_levels) for k in range(num_levels)]
colors = plt.cm.jet(np.linspace(0,1,num_levels))
for i, l in enumerate(levels):
    ax.plot(ns.T[l], color=colors[i], alpha=0.4, label=f"C(t={round(t_list[l],2)})")
    ax.set_xlabel("x (depth)")
ax.legend(loc="upper right")

filename = "DAR_profile"
plt.show()
```