Generated by Cython 0.29.24

Yellow lines hint at Python interaction.
Click on a line that starts with a "+" to see the C code that Cython generated for it.

Raw output: heat.c

```
+01: import numpy as np
+02: import matplotlib
+03: matplotlib.use('Agg')
+04: import matplotlib.pyplot as plt
 05:
 06: # Set the colormap
+07: plt.rcParams['image.cmap'] = 'jet'
 08:
+09: def evolve(u, u_previous, a, dt, dx2, dy2):
 10:     """Explicit time evolution.
 11:        u:            new temperature field
 12:        u_previous:   previous field
 13:        a:            diffusion constant
 14:        dt:           time step. """
 15:
+16:     n, m = u.shape
 17:
+18:     for i in range(1, n-1):
+19:         for j in range(1, m-1):
+20:             nu = ((u_previous[i-1, j] - 2*u_previous[i, j] + u_previous[i+1, j]) / dx2 + \
+21:             (u_previous[i, j-1] - 2*u_previous[i, j] + u_previous[i, j+1]) / dy2 )
+22:             u[i, j] = u_previous[i, j] + a * dt * nu
 23:
+24:     u_previous[:] = u[:]
 25:
+26: def iterate(field, field0, a, dx, dy, timesteps):
 27:     """Run fixed number of time steps of heat equation"""
 28:
+29:     dx2 = dx**2
+30:     dy2 = dy**2
 31:
 32:     # For stability, this is the largest interval possible
 33:     # for the size of the time-step:
+34:     dt = dx2*dy2 / ( 2*a*(dx2+dy2) )
 35:
+36:     for _ in range(1, timesteps+1):
+37:         evolve(field, field0, a, dt, dx2, dy2)
 38:
 39:
+40: def init_fields(filename):
 41:     # Read the initial temperature field from file
+42:     field = np.loadtxt(filename)
+43:     field0 = field.copy() # Array for field of previous time step
+44:     return field, field0
 45:
+46: def write_field(file, field, step):
+47:     plt.gca().clear()
+48:     plt.imshow(field)
+49:     plt.axis('off')
+50:     f, l = file.rfind('/')+1, file.rfind('.')
+51:     plt.savefig(f"./outputs/{file[f:l]}_heat_{step}.png")
 52:
 53:
```