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: cyheat.c

```
+01: #cython: language_level=3
 02: cimport cython
 03:
+04: cdef evolve(double [:, :] u, double [:, :] u_previous, double a, double dt, double dx2, double dy2):
 05:     """Explicit time evolution.
 06:        u:            new temperature field
 07:        u_previous:   previous field
 08:        a:            diffusion constant
 09:        dt:           time step. """
 10:
 11:     cdef Py_ssize_t n, m, i, j
 12:     cdef double nu
 13:
+14:     n = u.shape[0]
+15:     m = u.shape[1]
 16:
+17:     for i in range(1, n-1):
+18:         for j in range(1, m-1):
+19:             nu = ((u_previous[i-1, j] - 2*u_previous[i, j] + u_previous[i+1, j]) / dx2 + (u_previous[i, j-1] - 2*u_previous[i, j] + u_pr
+20:             u[i, j] = u_previous[i, j] + a * dt * nu
 21:
+22:     u_previous[:] = u[:]
 23:
 24: @cython.boundscheck(False)  # Deactivate bounds checking
 25: @cython.wraparound(False)   # Deactivate negative indexing.
+26: def iterate(double [:, :] field, double [:, :] field0, double a, double dx, double dy, int timesteps):
 27:     """Run fixed number of time steps of heat equation"""
 28:
+29:     cdef double dx2 = dx * dx
+30:     cdef double dy2 = dy * dy
 31:
 32:     # For stability, this is the largest interval possible
 33:     # for the size of the time-step:
+34:     cdef double dt = dx2*dy2 / ( 2*a*(dx2+dy2) )
 35:
 36:     cdef int i
+37:     for i in range(1, timesteps+1):
+38:         evolve(field, field0, a, dt, dx2, dy2)
```