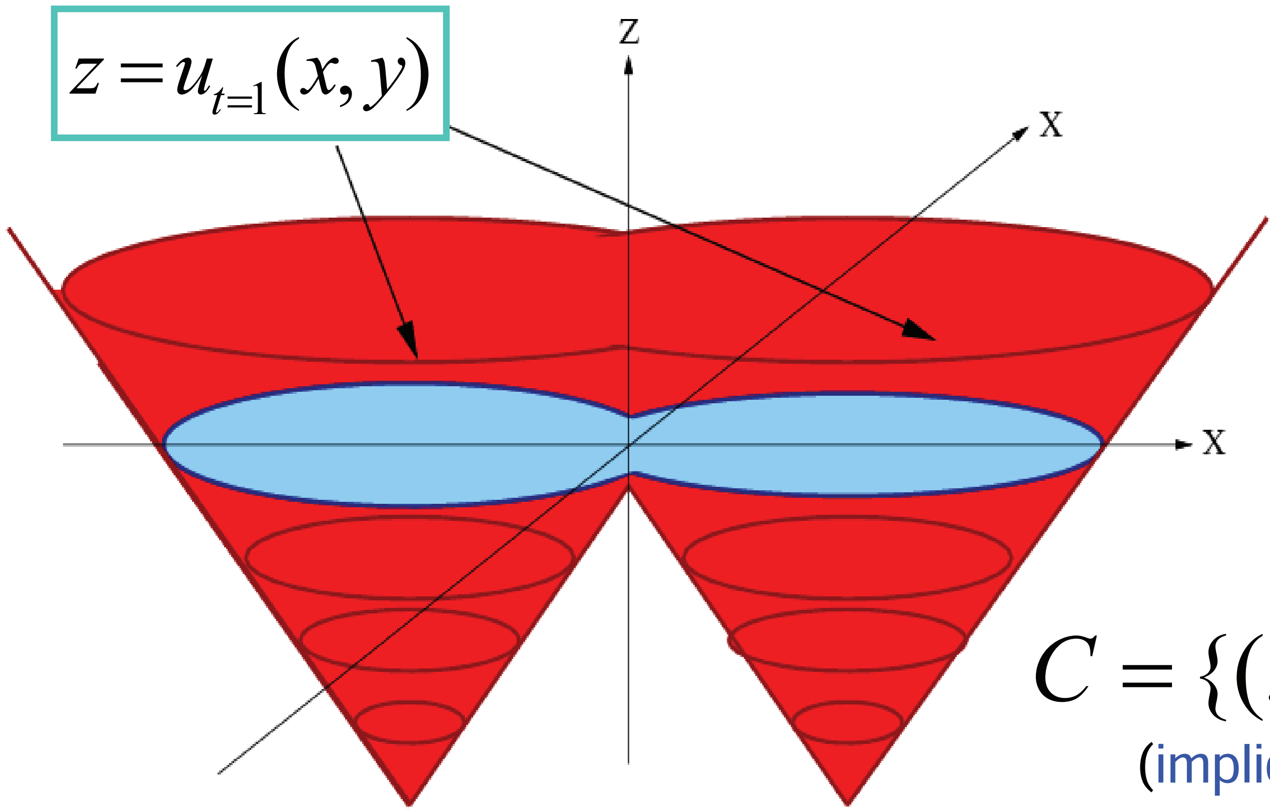
Level sets

- Curve evolution is great. But:
  - Topology changes are a major problem
  - Also, numerical issues and stopping conditions
    - Particles tend to colide, and solutions oscillate
- Level sets address these beautifully
- Basic idea: implicit function for curve
  - Think of a topological map of a park
  - There are curves drawn at a fixed height
    - I.e., the points where height = 1 mile
  - We will use this surface to evolve the curve

Level set surfaces

- To evolve a curve **C**, first compute a surface **u** such that **C** is level set of **u**
  - Obviously, not unique
  - Standard choice: signed distance function
    - **u**(x,y) = distance to nearest point in **C**
    - Negative if inside of **C**
- Instead of evolving the curve, we evolve the surface!
  - Sometimes called "contour propagation", "interface tracking", etc.

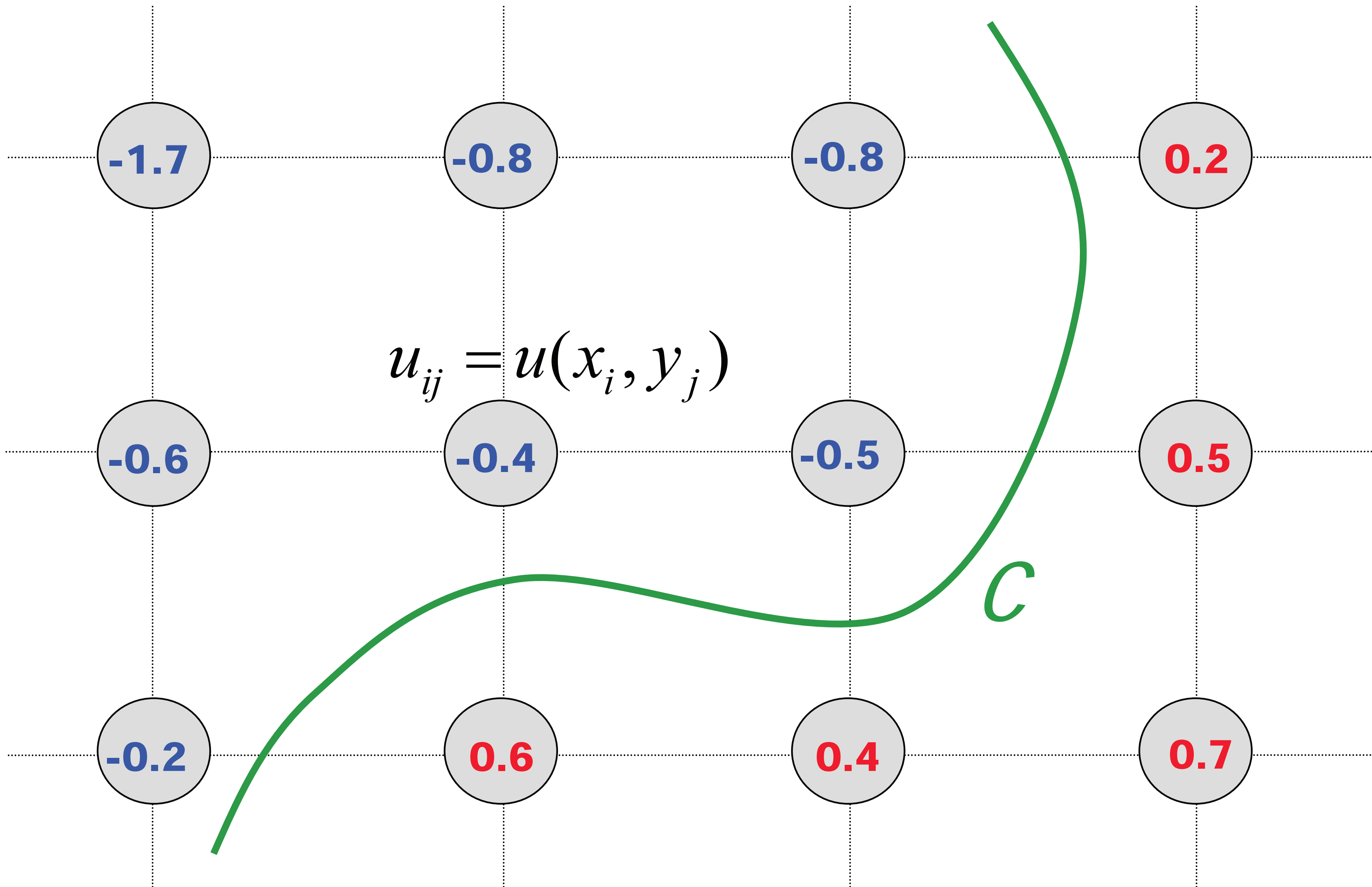
Example



$$C = \{(x, y) : u(x, y) = 0\}$$

(implicit contour representation)

Example

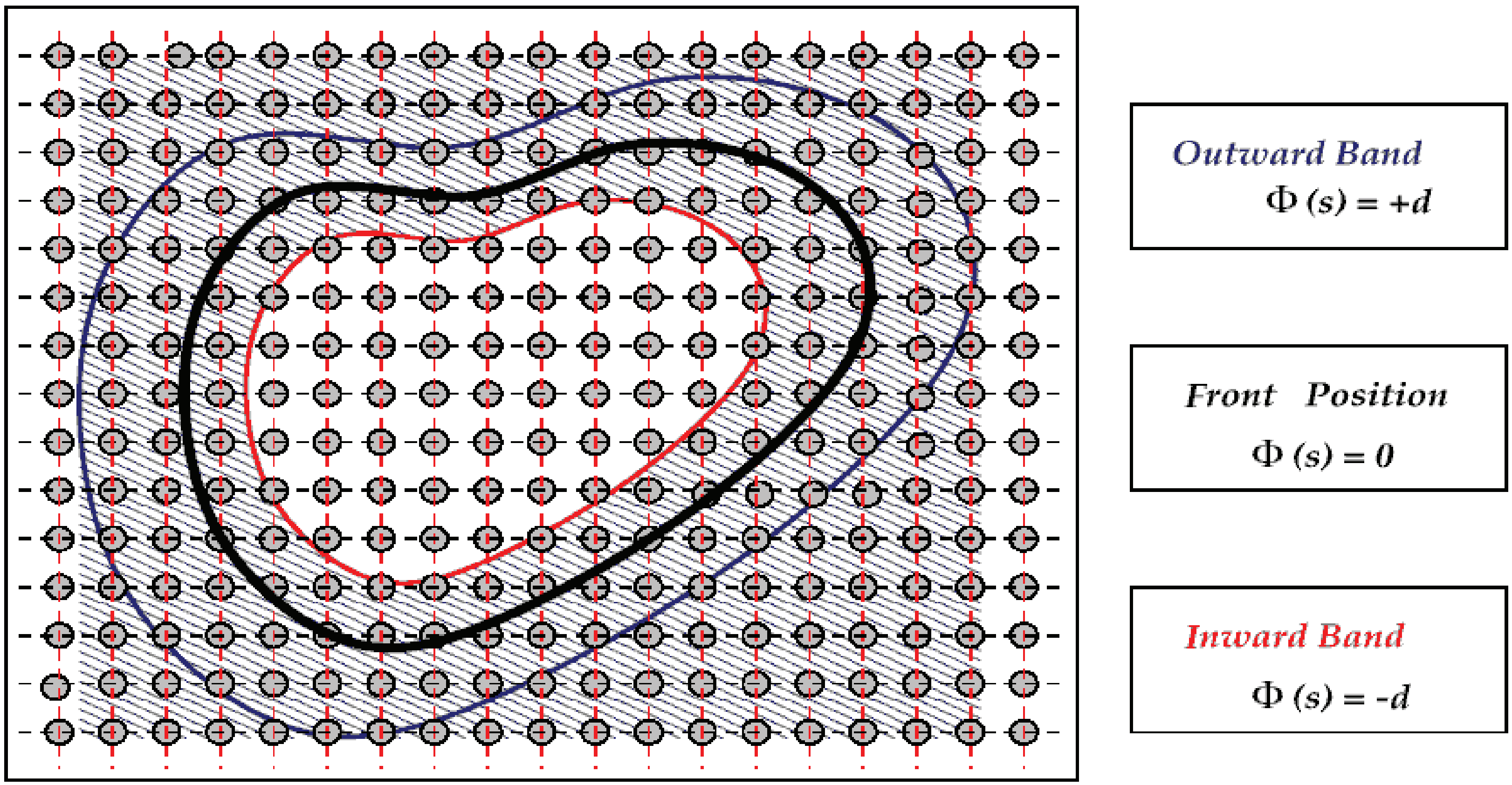


$$u_{ij} = u(x_i, y_j)$$

Level set evolution

- Instead of evolving **C** we evolve **u**
- Curve evolution: $$\frac{\partial \mathbf{C}}{\partial t} = \beta N$$
- Level set evolution: $$\frac{\partial \mathbf{u}}{\partial t} = \beta \|\nabla \mathbf{u}\|$$
- Amazing fact: if you evolve **u** by $\mathbf{u}_t$, *every* level set of **u** will evolve according to $\mathbf{C}_t$

Speedup: narrow band



State of the art

- Almost everyone who does curve evolution these days uses level sets
  - Handles changes in topology
  - Numerical stability
  - Natural generalization to higher dimensions
    - Evolving surfaces ("minimal surfaces")
- Typically add some sort of constant force ("balloon pressure") plus curvature
- It would be great to have a better way to incorporate shape