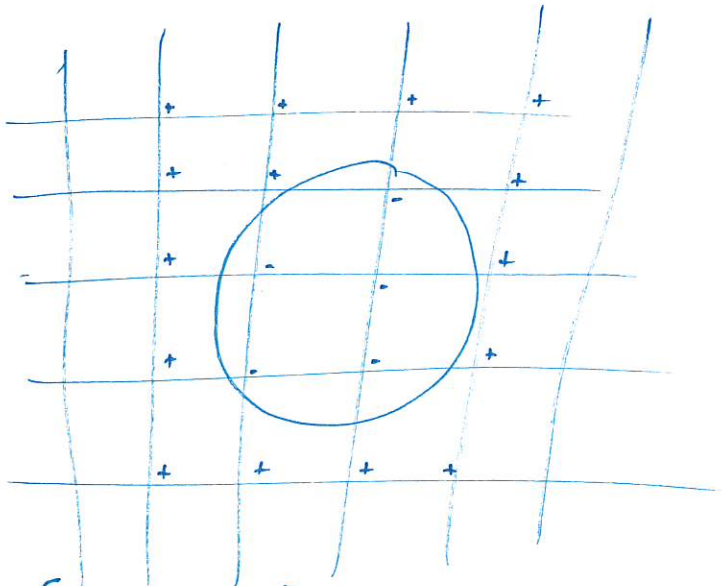
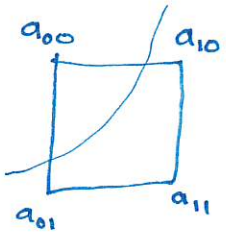
Grid-based implicit surface

* each grid node stores $F(x,y)$ at that location

* may be signed distance (not required)



within cell



$f(x,y,a_{00},a_{10},a_{01},a_{11})$

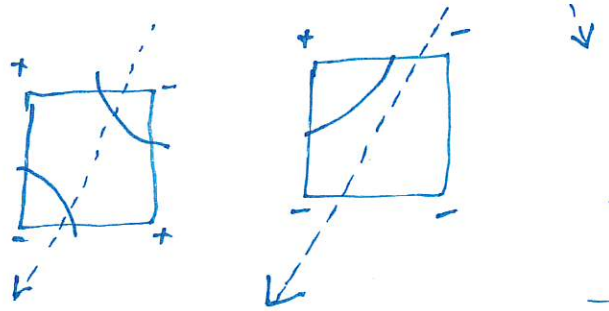
eg $F = (1-y)\left[(1-x)a_{00} + x a_{10}\right] + y\left[(1-x)a_{01} + x a_{11}\right]$

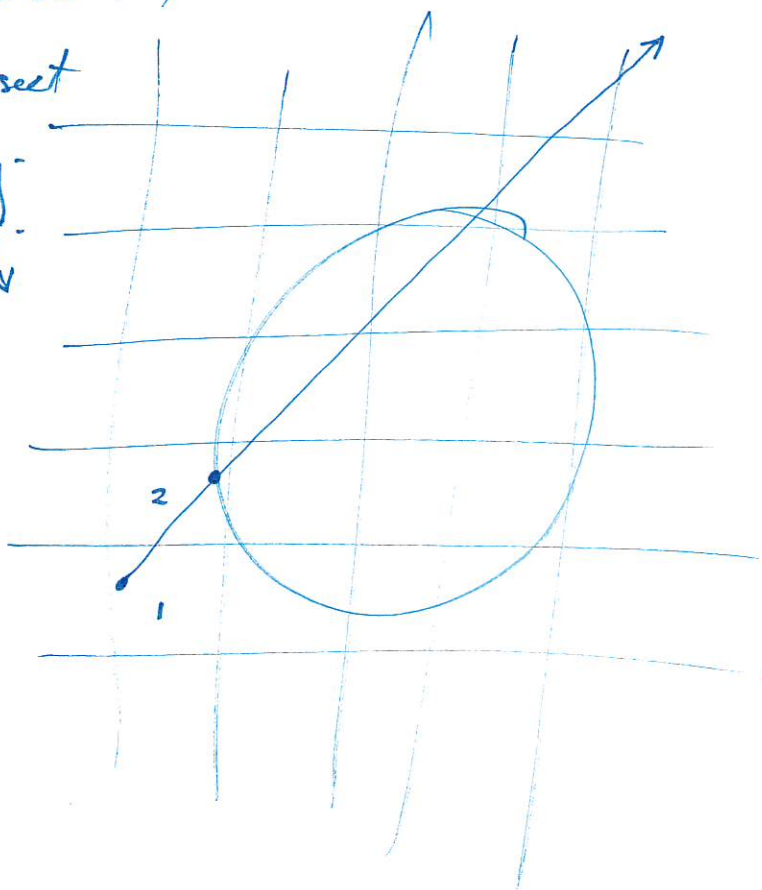
* defines an analytic surface in each cut cell

* if all nodes same sign, no intersect

* march cells (as with accel struct!)

* if cell cut, might have intersect (or more than one!)





* stop when intersect found

* $N = \dfrac{\nabla F}{\|\nabla F\|}$

shade as usual.

Ray tracing implicit surfaces

Note that in 2d, the $F = axy + bx + cy + d$

$\Rightarrow F(\vec{e} + \vec{u}t) = p + qt + rt^2$

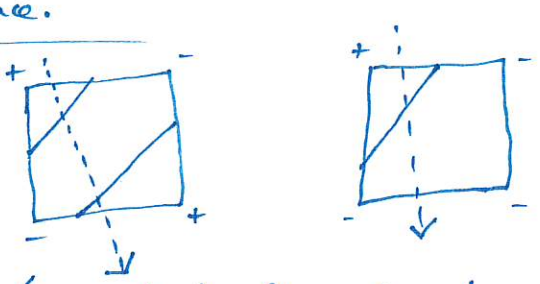
$\Rightarrow F(\vec{x}) = 0$ $\Rightarrow$ solve quadratic equation

in 3D $\rightarrow$ cubic

* fully smooth surface.

alternative

$\rightarrow$ surface is a



simplified type (eg. a line) for intersections

- cheaper

$\rightarrow$ compute normal from interpolation? or from segments?

- 3D? triangles?