It is important to notice that a given interface, say \( x^2 + y^2 = R^2 \), will have many different level set functions. For example

\[
\phi(x,y) = (x^2 + y^2)^{p/2} - R^p
\]

all have \( \{ \phi(x,y) = 0 \} = \{ x^2 + y^2 = R^2 \} \)

**Boolean Operations**

Let

\[
\phi_1(x,y) = (x-x_1)^2 + (y-y_1)^2 - R_1^2
\]

\[
\phi_2(x,y) = (x-x_2)^2 + (y-y_2)^2 - R_2^2
\]

\[
\phi(x,y) = \min(\phi_1, \phi_2)
\]

Then, loosely speaking, the zero level set of \( \phi \) are two the circles given by \( \phi_1 = 0 \) and \( \phi_2 = 0 \). More precisely,

\[
\{ \phi = 0 \} = \text{Boundary of } \{ \phi_1 < 0 \} \cup \{ \phi_2 < 0 \}
\]
To see why let us look at a one dimensional case:

\[
\phi_1 = |x - x_1| - R_1 \quad \phi_2 = |x - x_2| - R_2
\]

**Case 1**

\[\phi_1 < 0 \quad \phi_2 < 0\]

\[\min(\phi_1, \phi_2)\]

\[\phi = \min(\phi_1, \phi_2)\] is the union operation.

**Case 2**

\[\phi_1 \Rightarrow \phi_2\]

\[\min(\phi_1, \phi_2)\]

\[\phi = \max(\phi_1, \phi_2)\] is the intersection operation.
One can also perform set substraction

\[ \phi = \max(\phi_1, -\phi_2) \]

Other Shapes (see matlab codes 1-3)

a) Let \( \phi_1 = |x| - R \quad \phi_2 = |y| - R \)

Square = \( \max(\phi_1, \phi_2) \)

b) Let \( \Gamma \) be an open curve of finite length. Let \( d(x,y) \) be the closest distance to \( \Gamma \) from \( (x,y) \).

Let \( \phi = d(x,y) - W \)

for an example, see the construction of the initial condition in matlab code 7.
Geometric Properties

Let \( \phi = \phi(x, y, \ldots) \) then

\[
\{ \phi = 0 \} = \begin{cases} \text{curve in } \mathbb{R}^2 \\ \text{surface in } \mathbb{R}^3 \end{cases}
\]

Notice \( \frac{\nabla \phi}{|\nabla \phi|} \) is a unit vector that is perpendicular to the level curves of \( \phi \).

\[\Rightarrow \quad \frac{\nabla \phi}{|\nabla \phi|} \bigg|_{\phi=0} = \vec{n} \quad \text{outward drawn normal vector} \]

Recall \( \kappa = \frac{\nabla_s \cdot \vec{n}}{\text{surface divergence}} \)
Now \( \nabla = \hat{n} \hat{n} \cdot \nabla + \nabla_s \), in component form

\[ \Rightarrow \quad \partial_i = n_i n_j \partial_j + (\nabla_s)_i ; \]

\[ \Rightarrow \quad \nabla \cdot \hat{n} = \partial_i n_i = n_i n_j \partial_j n_i + (\nabla_s)_i n_i \]

\[ = \frac{1}{2} n_j \partial_j n_i n_i + (\nabla_s)_i n_i \]

\[ = \nabla_s \cdot \hat{n} \]

Therefore the curvature of a level curve is given by

\[ \kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \]

Example \( \phi = x^2 + y^2 - R^2 \)

\[ \Rightarrow \quad \frac{\nabla \phi}{|\nabla \phi|} = \frac{x}{\sqrt{x^2 + y^2}} \hat{i} + \frac{y}{\sqrt{x^2 + y^2}} \hat{j} \]

\[ \Rightarrow \quad \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} = \frac{1}{\sqrt{x^2 + y^2}} \]

\[ \Rightarrow \quad \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \bigg|_{\phi=0} = \frac{1}{R} \]
One can also compute arc length or surface area of \( \{ \phi = 0 \} \):

\[
L = \int_{\mathbb{R}^2} \| \nabla \phi \| \, dx \, dy,
\]

arc length

\[
S = \int_{\mathbb{R}^3} \| \nabla \phi \| \, dx \, dy \, dz,
\]

surface area

Why? \( \Rightarrow \int_a^b \phi(f(x)) |f'(x)| \, dx = 1 \)

provided \( f(x) = 0 \) has one root in \([a, b]\).

**Moving Level Sets Around**

Zero level set moves with the velocity field.
Now \( \Phi = \Phi(\vec{x}, t) \) and our goal is to find the time evolution of \( \Phi \). We will let \( \vec{V} = V(\vec{x}, t) \) be the velocity field. Let \( \vec{X} \) denote tracer particles

\[
\Rightarrow \quad \frac{d\vec{x}}{dt} = \vec{V}
\]

Since the zero level set moves with \( \vec{V} \) it is convenient to demand that all level sets move with \( \vec{V} \).

\[
\Rightarrow \quad \frac{d\Phi}{dt}(\vec{x}(t), t) = 0
\]

\[
\Rightarrow \quad \frac{\partial \Phi}{\partial t} + \frac{d\vec{x}}{dt} \cdot \nabla \Phi = 0
\]

\[
\Rightarrow \quad \frac{\partial \Phi}{\partial t} + \nabla \cdot \nabla \Phi = 0 \quad \text{(advection)}
\]

Notice since \( \vec{n} = \nabla \Phi / |\nabla \Phi| \) we have

\[
\frac{\partial \Phi}{\partial t} + \vec{n} \cdot \nabla |\nabla \Phi| = 0
\]
Geometric Motions

\[ \nabla n = 1 \Rightarrow \vec{\nabla} = \hat{n} = \frac{\nabla \phi}{|\nabla \phi|} \]

\[ \Rightarrow \frac{\partial \phi}{\partial t} + \nabla \phi \cdot \nabla \phi = 0 \]

\[ \Rightarrow \frac{\partial \phi}{\partial t} + |\nabla \phi| = 0 \] Hamilton-Jacobi Equation

\[ \nabla n = -\kappa \text{ curvature flow} \]

\[ \vec{\nabla} = -\kappa \frac{\nabla \phi}{|\nabla \phi|} \]

\[ \Rightarrow \frac{\partial \phi}{\partial t} = \kappa |\nabla \phi| \]

\[ \kappa = \frac{\phi_{xx} \phi_{yy} - 2 \phi_x \phi_y \phi_{xy} + \phi_{yy} \phi_{xx}}{|\nabla \phi|^3} \]

\[ \Rightarrow \text{nonlinear heat equation} \]
Numerical Implementation

Advection in one dimension \((v = \text{const})\)

\[
\frac{\partial \phi}{\partial t} + v \frac{\partial \phi}{\partial x} = 0
\]

\(\phi_j^n = \phi(n \Delta t, j \Delta x)\) \hspace{1cm} \Delta t = \text{time step}

\(\Delta x = \text{grid size}\)

Notice

\[
\frac{\partial \phi}{\partial t} = \frac{\phi_{j}^{n+1} - \phi_{j}^{n}}{\Delta t}
\]

\[
\frac{\partial \phi}{\partial x} = \frac{\phi_{j+1}^{n} - \phi_{j}^{n}}{\Delta x} \hspace{1cm} \text{or} \hspace{1cm} \frac{\phi_{j}^{n} - \phi_{j-1}^{n}}{\Delta x}
\]

Both of these are equally good approximations of \(\phi_x\)

Crucial observation: information comes upwind from your location. Therefore we choose the approximation to best capture this observation.
Therefore we have
\[
(\nabla \phi_x)_i^n = \begin{cases} 
\nabla \left( \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x} \right) & v < 0 \\
\nabla \left( \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} \right) & v > 0
\end{cases}
\]

Consequently we have the following numerical scheme
\[
\phi_j^{n+1} = \phi_j^n - \Delta t \ (\nabla \phi_x)_i^n
\]  \(\star\)

This is a first order explicit upwind scheme. One can prove that the scheme is stable provided
\[
\frac{\sqrt{\Delta t}}{\Delta x} < 1
\]

This is a very dissipative scheme. In fact one can prove \(\star\) more accurately solves:
\[
\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \nabla \phi) = \frac{1}{2} \frac{(\Delta x - \Delta t)}{|v|} \phi_{xx}
\]
\(\text{numerical diffusion}\)
Normal Velocity = 1 in one dimension

\[ \frac{\partial \phi}{\partial t} + |\phi_x| = 0 \Rightarrow \frac{\partial \phi}{\partial t} + \text{sgn}(\phi_x) \phi_x = 0 \]

Upwind?

Let \( D_j^+ \phi = \frac{\phi_{j+1} - \phi_j}{\Delta x} \) and \( D_j^- \phi = \frac{\phi_j - \phi_{j-1}}{\Delta x} \)

So try:

\[ |\phi_x| = \begin{cases} |D_j^+ \phi| & \text{if } D_j^+ \phi < 0 \\ |D_j^- \phi| & \text{if } D_j^- \phi > 0 \end{cases} \]

But what if \( D_j^+ \phi > 0 \neq D_j^- \phi < 0 \)?

Here is what you need to do:

\[ |\phi_x| = \max \left( |\max(D_j^- \phi, 0)|, |\min(D_j^+ \phi, 0)| \right) \]

(\( \Box \))

Notice, for example, if \( D_j^+ \phi < 0 \), \( D_j^- \phi < 0 \)

then

\[ |\phi_x| = |D_j^+ \phi| \quad \text{as you would like.} \]

(\( \Box \)) is a Godunov scheme for a Hamilton-Jacobi Equation. The reasoning behind (\( \Box \)) is a long story which we will now try to summarize.
Godunov Schemes for Hamilton Jacobi Eqs.

\[ \partial_t \phi + \partial_x \phi = 0 \Rightarrow \partial_t \phi_x + \partial_x \phi_x = 0 \]

let \( u = \phi_x \Rightarrow \partial_t u + \partial_x u = 0 \) (conservation law) (1D only)

\( \Rightarrow \) Apply Godunov Schemes for conservation laws \( \Rightarrow \) Godunov Schemes for HJ.

Burgers Equation

In this section we will discuss how one computes solutions to Burgers equation:

\[ u_t + uu_x = 0 \Rightarrow u_t + \partial_x \frac{1}{2} u^2 = 0 \]

A big problem with this equation is that the solutions become multivalued. For example:

\[ u(0, x) \quad \Rightarrow \quad u(t, x) \]

This is considered unphysical.
Therefore one could consider the following modification

\[ u_t + 2x \frac{1}{2} u^2 = v \ u_{xx} \quad \text{for} \quad v \ll 1. \]

\[ \Rightarrow \text{Entropy satisfying solutions of } u_t + 2x \frac{1}{2} u^2 = 0. \]

Now,

\[ \text{Solution of the Riemann Problem } \quad v \ll 1. \]

\[ u(x, 0) = \begin{cases} u_L & x < 0 \\ u_R & x > 0 \end{cases} \]

Two cases:

\[ u_L > u_R \quad \text{shock} \]

\[ u_R > u_L \quad \text{rarefaction} \]

\[ \Rightarrow \text{Entropy satisfying weak solutions of } u_t + 2x \frac{1}{2} u^2 = 0. \]
\textbf{Shock:} \quad \text{Speed} = \frac{1}{2}(u_L + u_R) = s

\textbf{Rarefaction:}

\textbf{Important Fact}

\begin{align*}
\text{If} \quad u_L > u_R & \implies u(0,t) = \begin{cases} u_L, & s > 0 \\ u_R, & s < 0 \end{cases} \\
\text{If} \quad u_R > u_L & \implies u(0,t) = \begin{cases} u_L, & u_R > u_L > 0 \\ u_R, & 0 > u_R > u_L \\ 0, & u_R > 0 > u_L \end{cases}
\end{align*}

\text{Let} \quad U_\star(u_L, u_R) = \begin{cases} u_L, & \text{if } s > 0 \text{ and } u_L > 0 \\ u_R, & \text{if } s < 0 \text{ and } u_R < 0 \\ 0, & \text{otherwise} \end{cases}

\implies u(0,t) = U_\star(u_L, u_R)
Now we can explain Godonov's Method

$$\Rightarrow \frac{\partial u}{\partial t} + \frac{\partial (\frac{1}{2} u^2)}{\partial x} = 0 \Rightarrow \frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0$$

\[ u_{i-1} \quad u_i \quad u_{i+1} \]

\[ F_{i-\frac{1}{2}} \quad F_{i+\frac{1}{2}} \]

$$\Rightarrow \frac{u_{i+1}^n - u_i^n}{\Delta t} + \frac{F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n}{\Delta x} = 0$$

How to relate \( u_i \) to \( F_{i+\frac{1}{2}} \)?

Approximate \( u \) by a step function. Solve the Riemann Problem in the intervals \([i, i+1]\). Use this to find \( F_{i+\frac{1}{2}} \).
So if we solve the initial value problem for a short time then we know that

\[ u_{i+\frac{1}{2}} = u_* (u_i, u_{i+1}) \]

Therefore Godunov's method for Burgers equation is:

\[ \frac{u_c^{n+1} - u_c^n}{\Delta t} + \frac{1}{2} u_*^2 (u_i, u_{i+1}) - \frac{1}{2} u_*^2 (u_{i-1}, u_i) = 0 \]

⇒ This converges to solutions of

\[ u_t + u u_x = \nu u_{xx} \quad \text{for} \quad \nu \to 0. \]

Or equivalently

Entropy satisfying weak solutions of

\[ u_t + \partial_x (\frac{1}{2} u^2) = 0. \]

The numerical scheme generates enough, but not too much, dissipation to find the entropy satisfying solutions.
General Case

First Notice:

\[ \frac{1}{2} U_x^2 (u_L, u_R) = \begin{cases} \arg \max_{u \in (u_R, u_L)} \frac{1}{2} u^2 & u_L > u_R \\ \arg \max_{u \in (u_L, u_R)} \frac{1}{2} u^2 & u_R > u_L \end{cases} \]

For \( \frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0 \), Godunov's method is

\[ \frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}}{\Delta x} = 0 \]

With

\[ F_{i+\frac{1}{2}} = F(u_j^n, u_{j+1}^n) \]

where

\[ F(u_L, u_R) = \begin{cases} \arg \max_{u \in (u_R, u_L)} F(u) & u_L > u_R \\ \arg \max_{u \in (u_L, u_R)} F(u) & u_R > u_L \end{cases} \]

So for \( u_L + \Delta x / |u| = 0 \) we use \( F(u) = |u| \)

but notice that for \( F(u) = |u| \) one has

\[ F(u_L, u_R) = \max \left( \max(u_L, 0), |\min(u_R, 0)| \right) \]
But recall we really wanted to solve \( \phi_t + F(\phi_x) = 0 \). Given Godunov's method for \( u_t + 2_x F(u) = 0 \), what is the corresponding scheme for \( \phi_t + F(\phi_x) = 0 \)? Recall \( u = \phi_x \), therefore we let:

\[
\phi^n_i = (\phi^n_{i+\frac{1}{2}} - \phi^n_{i-\frac{1}{2}})/\Delta x
\]

Substitute this into (G) to obtain

\[
D^+ \left( \frac{\phi^{n+1}_{i+\frac{1}{2}} - \phi^n_{i+\frac{1}{2}}}{\Delta t} + F_{i+\frac{1}{2}} \right) = 0
\]

where \( D^+(\phi_{i+\frac{1}{2}}) = (\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}})/\Delta x \).

Now let \( i+\frac{1}{2} = j \) and we have

\[
\Rightarrow \quad \frac{\phi^{n+1}_{j} - \phi^n_{j}}{\Delta t} + F_{j} = 0
\]

where \( F_j = F(D^+\phi, D^{-}\phi) \),

\[
D^-\phi = (\phi_i - \phi_{i-1})/\Delta x \quad \text{;} \quad D^+\phi = (\phi_{i+1} - \phi_i)/\Delta x
\]
Therefore the Godunov scheme for
\[ \frac{\phi_t + \phi_x}{\Delta t} = 0 \]
is given by
\[ \frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + F_j = 0 \]
where
\[ F_j = \max\left(\max(D^-_j \phi_j, 0), \min(D^+_j \phi_j, 0)\right) \]
This has been implemented in Matlab code 5.

Example

\[ V_n = \begin{cases} a & \phi_x > 0 \\ b & \phi_x < 0 \end{cases} \]

\[ V_n = b - (a - b) \text{H}(\phi_x) \]

\[ F = (b - (a-b) \text{H}(\phi_x)) \mid \phi_x \mid \]

Notice the convexity of \( F \) depends on \( a, b \). For example if \( a < 0 \) & \( b < 0 \) then \( F \) is concave our formula needs to changed.