W. Ross Morrow
Smooth Optimization via a Level Set Formulation

1 Let $f : \mathbb{R}^n \to \mathbb{R}$ be a C^1 function whose minima are attained. We wish to find these minima, numerically, through evolving an auxiliary function $\varphi : \mathbb{R}_+ \times \mathbb{R}^2 \to \mathbb{R}$ according to

$$\dot{\varphi} - \nabla f \cdot \nabla \varphi = 0, \qquad \varphi(0, \mathbf{x}) = \varphi_0(\mathbf{x}). \tag{1}$$

The key benefit of this approach is the following stipulation: Suppose some minimizer $\mathbf{x}_*$ of f satisfies $\varphi(0, \mathbf{x}_*) \leq 0$; then $\varphi(t, \mathbf{x}_*) \leq 0$ for all $t > 0$, where φ satisfies (1).[1] Thus, if we let $\Omega(t) = \{\mathbf{x} \in \mathbb{R}^2 : \varphi(t, \mathbf{x}) \leq 0\}$, then we stipulate that minima of f contained in $\Omega(0)$ remain in $\Omega(t)$ for all $t > 0$. Furthermore, we believe there could be auspicious circumstances yielding a monotonicity to the evolution of $\Omega(t)$. Precisely, if $\Omega(s) \subset \Omega(t)$ whenever $s \leq t$, any minima contained in $\Omega(0)$ are "resolved" by evolving Ω via φ.

2 **Numerical Implementation.**

Characteristics. The method of characteristics is informative for the problem (1). Specifically, consider curves $\mathbf{x} : \mathbb{R}_+ \to \mathbb{R}^2$ and $t : \mathbb{R}_+ \to \mathbb{R}_+$, and define $\psi : \mathbb{R}_+ \to \mathbb{R}$ by the formula $\psi(s) = \varphi(t(s), \mathbf{x}(s))$. Then

$$\frac{d\psi}{ds} = \dot{\varphi}\frac{dt}{ds} + \nabla\varphi \cdot \frac{d\mathbf{x}}{ds};$$

if φ solves (1),

$$\frac{dt}{ds} = 1 \qquad \text{and} \qquad \frac{d\mathbf{x}}{ds} = -\nabla f$$

then ψ is constant over s. The curves $s \mapsto (t(s), \mathbf{x}(s)) \in \mathbb{R}_+ \times \mathbb{R}^2$ so specified are the characteristics for the problem (1), and they flow along the directions of steepest descent relative to f.

"Upwind" Derivatives for a Uniform Grid. This observation unveils the appropriate numerical "upwind" derivatives of φ: taking a uniform grid on $\mathbb{R}^2$ with mesh parameter h, setting $\varphi^n_{i,j} = \varphi(t_n, ih, jh)$,

$$\xi_{i,j} = (\nabla f(\mathbf{x}_{i,j}))_x, \qquad \eta_{i,j} = (\nabla f(\mathbf{x}_{i,j}))_y,$$

and

$$D_x^+ \varphi^n_{i,j} = \frac{\varphi^n_{i+1,j} - \varphi^n_{i,j}}{h} \qquad D_y^+ \varphi^n_{i,j} = \frac{\varphi^n_{i,j+1} - \varphi^n_{i,j}}{h}$$
$$D_x^- \varphi^n_{i,j} = \frac{\varphi^n_{i,j} - \varphi^n_{i-1,j}}{h} \qquad D_y^- \varphi^n_{i,j} = \frac{\varphi^n_{i,j} - \varphi^n_{i,j-1}}{h},$$

we have

$$D_x \varphi^n_{i,j} = \begin{cases} D_x^+ \varphi^n_{i,j} & \xi_{i,j} \geq 0 \\ D_x^- \varphi^n_{i,j} & \xi_{i,j} < 0 \end{cases} \quad \text{and} \quad D_y \varphi^n_{i,j} = \begin{cases} D_y^+ \varphi^n_{i,j} & \eta_{i,j} \geq 0 \\ D_y^- \varphi^n_{i,j} & \eta_{i,j} < 0 \end{cases}$$

as "upwind" derivatives for the explicit update

$$\varphi^{n+1}_{i,j} = \varphi^n_{i,j} + k\left[\xi_{i,j} D_x \varphi^n_{i,j} + \eta_{i,j} D_y \varphi^n_{i,j}\right]. \tag{2}$$

[1] For a conceptual proof, simply note that $\dot{\varphi}(t, \mathbf{x}_*) = 0$ for all $t \geq 0$ since $\nabla f(\mathbf{x}_*) = \mathbf{0}$.

Remark: Stability. This method is conditionally stable, requiring time steps satisfying a bound determined by the mesh size and the gradients of f over the computational domain. In general, the time step parameter k must satisfy a bound of the form

$$k \leq \frac{h}{\lambda} \qquad \text{where} \qquad \lambda = \max\{||\nabla f(\mathbf{x})|| : \mathbf{x} \in C\}.$$

For example, supposing that f is Lipschitz-continous on $C \subset \mathbb{R}^2$ with Lipschitz constant κ yields the simplified bound $k \leq h/\kappa$. We use the heuristic bound

$$k \leq \frac{h}{\zeta} \qquad \text{where} \qquad \zeta = \max\{||\nabla f(\mathbf{x}_{i,j})|| : 1 \leq i, j \leq N+1\},$$

where in the numerical examples provided below we apply $k = h/(2\zeta)$.

Remark: Boundaries. On the boundary of the computational domain, i.e. when $i, j \in \{1, N+1\}$, one of the two directional differences D^+, D^- is not defined. We implement the following boundary scheme, equivalent to assuming that f is constant outside of the computational domain. Consider the case $i = 1$, where $D_x^+ \varphi_{1,j}^n$ is defined and $D_x^- \varphi_{1,j}^n$ is not. The formula given above breaks down when $\xi_{1,j} < 0$, due to fact that the "upwind" differencing asks for information outside of the computational domain. In this case we simply set $D_x \varphi_{1,j}^n = 0$.

Extrapolating this case, we generate the boundary formulae

$$D_x \varphi_{1,j}^n = \begin{cases} D_x^+ \varphi_{1,j}^n & \xi_{1,j} \geq 0 \\ 0 & \xi_{1,j} < 0 \end{cases} \quad \text{and} \quad D_x \varphi_{N+1,j}^n = \begin{cases} 0 & \xi_{N+1,j} \geq 0 \\ D_x^- \varphi_{N+1,j}^n & \xi_{N+1,j} < 0 \end{cases}$$

and

$$D_y \varphi_{i,1}^n = \begin{cases} D_y^+ \varphi_{i,1}^n & \eta_{i,1} \geq 0 \\ 0 & \eta_{i,1} < 0 \end{cases} \quad \text{and} \quad D_y \varphi_{i,N+1}^n = \begin{cases} 0 & \eta_{i,N+1} \geq 0 \\ D_y^- \varphi_{i,N+1}^n & \eta_{i,N+1} < 0 \end{cases}$$

For example, suppose that $\xi_{1,1} < 0$ and $\eta_{1,1} < 0$; i.e. the gradient of f at the top-left corner of the computational domain points outside of the computational domain (i.e. the characteristics of (1) are entering the computational domain), and in the particular way illustrated in FIGURE. The boundary differences defined above yield $\varphi_{1,1}^{n+1} = \varphi_{1,1}^n$ for all $n \geq 0$, which is essentially a signal that the computational domain should be either increased in size or translated.

Example 1. Our first example involves a quadratic form. We let $f(x,y) = (x-1)^2 + (y-1)^2$. We take the initial level-set function $\varphi_0(x,y) = (x-1.75)^2 + (y-1.75)^2$, and observe the evolution shown in Fig. 1. The zero-level set of φ does indeed approach the unique minimizer $(1,1)$. However, autonomous shrinking of the level-set yields an innaccurate result, as shown in Fig. 2.

Example 2. Our second example involves a quartic, everywhere nonnegative, with minimizers $(-1,-1)$ and $(1,1)$, and a saddle point at $(0,0)$:

$$f(x,y) = 12 - 3x^2 - 18xy - 3y^2 + 2x^4 + 3x^3y + 2x^2y^2 + 3xy^3 + 2y^4.$$

Taking the initial level-set function $\varphi_0(x,y) = (x-1.75)^2 + (y-1.75)^2$, we observe the evolution shown in Fig. 3. Again, autonomous shrinking of the level-set yields an innaccurate result, as shown in Fig. 4.

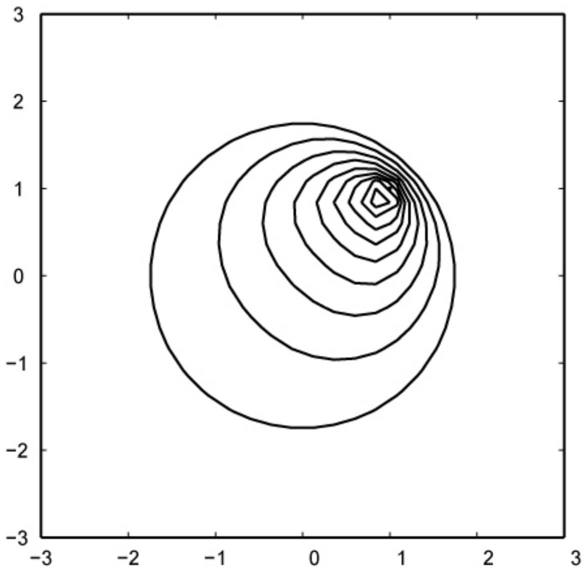

Figure 1: Example 1 with $N = 25$; $T = 500$; contours plotted at $t = 0, 10, 20, \ldots, 500$.

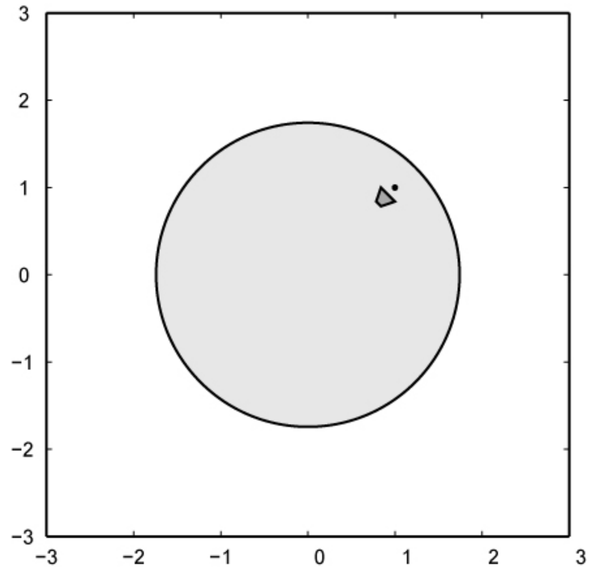

Figure 2: Initial (light grey) and final (dark grey) lower sections for example 1.

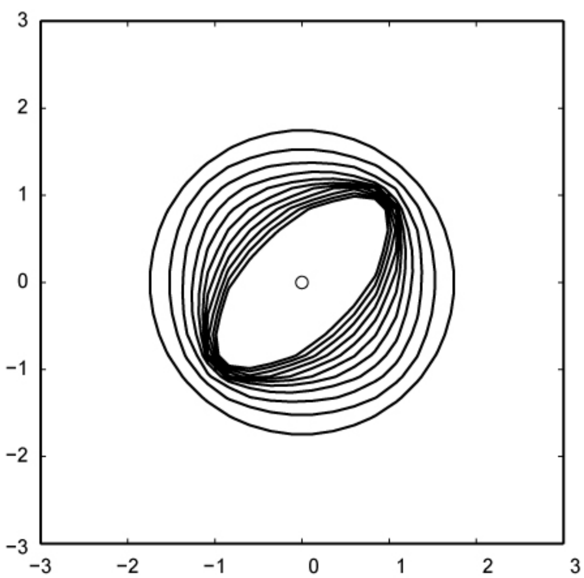

Figure 3: Example 2 with $N = 25$; $T = 500$; contours plotted at $t = 0, 50, 100, \ldots, 500$.

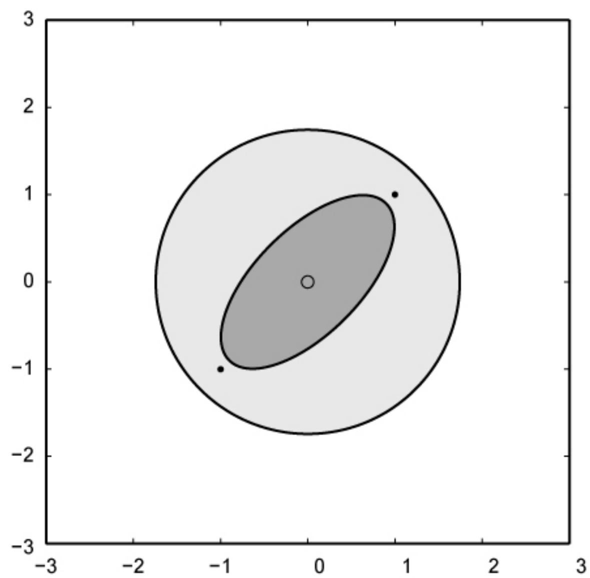

Figure 4: Initial (light grey) and final (dark grey) lower sections for example 2.

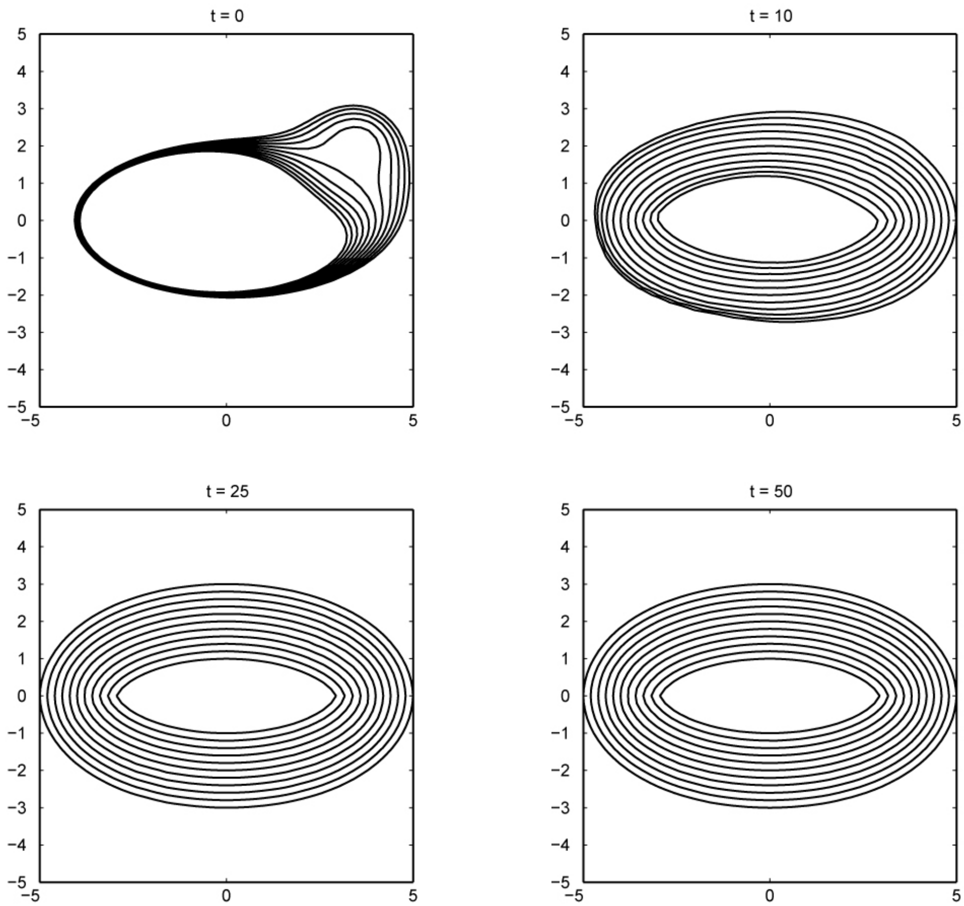

Figure 5: Reinitialization test.

Reinitialization. Periodically, φ_t should be "reset" to be the signed distance function relative to $\varphi_t^{-1}(0) = \Gamma(t)$. As described by Russo & Smereka (2000), this can be accomplished by solving

$$\dot{\psi} + \operatorname{sgn}(\varphi_t)\,||\nabla\psi|| = \operatorname{sgn}(\varphi_t) \qquad\qquad \psi(0, \mathbf{x}) = \varphi_t(\mathbf{x})$$

where

$$\operatorname{sgn}(\lambda) = \begin{cases} 1 & \lambda > 0 \\ 0 & \lambda = 0 \\ -1 & \lambda < 0 \end{cases}.$$

We apply the first-order numerical method described by Russo & Smereka (2000) to solve this equation. Their test