

A Hamilton-Jacobi-based Proximal Operator

Stanley Osher¹, Howard Heaton², Samy Wu Fung³

¹UCLA, ²Typal Academy, ³Colorado School of Mines

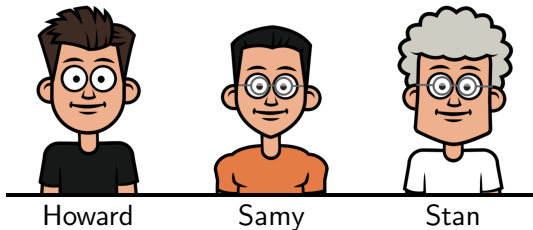


Scope: Optimization with proximals when we do not have exact formulas

Problem: For a continuous and weakly convex function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and $t > 0$, estimate

$$\text{prox}_{tf}(x) = \underset{y}{\operatorname{argmin}} f(y) + \frac{1}{2t} \|y - x\|^2,$$

only using evaluations of f .





- Moreau Envelope
- Hamilton-Jacobi PDEs
- Cole-Hopf Transformation
- Monte Carlo Sampling



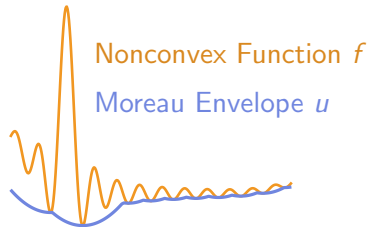
Given a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and time $t > 0$, the Moreau envelope u is

$$u(x, t) \triangleq \min_y f(y) + \frac{1}{2t} \|y - x\|^2.$$

When it exists, the gradient is

$$\nabla u(x, t) = \frac{1}{t} (x - \text{prox}_{tf}(x))$$

$$\implies \text{prox}_{tf}(x) = x - t \nabla u(x, t).$$





The envelope u solves

$$\begin{cases} u_t + \frac{1}{2} \|\nabla u\|^2 = 0 & \text{in } \mathbb{R}^n \times (0, \infty) \\ u = f & \text{on } \mathbb{R}^n \times \{t = 0\}. \end{cases}$$

For small $\delta > 0$, we can approximate u using the viscous PDE

$$\begin{cases} u_t^\delta + \frac{1}{2} \|\nabla u^\delta\|^2 = \frac{\delta}{2} \Delta u^\delta & \text{in } \mathbb{R}^n \times (0, \infty) \\ u^\delta = f & \text{on } \mathbb{R}^n \times \{t = 0\}. \end{cases}$$



Using an idea from Cole and Hopf, we use the change of variables

$$v^\delta = \exp(-u^\delta/\delta),$$

for which v^δ solves the heat equation

$$\begin{cases} v_t^\delta - \frac{\delta}{2} \Delta v^\delta = 0 & \text{in } \mathbb{R}^n \times (0, \infty) \\ v^\delta = \exp(-f/\delta) & \text{on } \mathbb{R}^n \times \{t = 0\}. \end{cases}$$



For a heat kernel $\Phi_{\delta t}$, the heat equation solution v^δ can be written as

$$v^\delta(x, t) = \left(\Phi_{\delta t} * \exp(-f/\delta) \right)(x) = \mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [\exp(-f(y)/\delta)],$$

with the expectation over a normal distribution with variance δt and mean x . Then

$$\nabla v^\delta(x, t) = -\frac{1}{\delta t} \cdot \mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [(x - y) \exp(-f(y)/\delta)],$$

which implies

$$\nabla u^\delta(x, t) = \frac{1}{t} \cdot \left(x - \frac{\mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [y \cdot \exp(-f(y)/\delta)]}{\mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [\exp(-f(y)/\delta)]} \right).$$



Using the gradient of the Moreau envelope and our approximation,

$$\begin{aligned}\text{prox}_{tf}(x) &= x - t \nabla u(x, t) \\ &\approx x - t \nabla u^\delta(x, t) \\ &= \frac{\mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [y \cdot \exp(-f(y)/\delta)]}{\mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [\exp(-f(y)/\delta)]}.\end{aligned}$$

(Informal) Theorem: If t is sufficiently small and $u(x, t) \geq 0$, then

$$\lim_{\delta \rightarrow 0^+} \frac{\mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [y \cdot \exp(-f(y)/\delta)]}{\mathbb{E}_{y \sim \mathcal{N}(x, \delta t)} [\exp(-f(y)/\delta)]} = \text{prox}_{tf}(x).$$

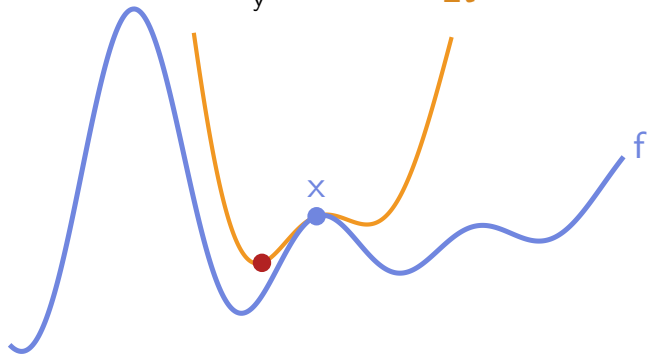


Algorithm 1 HJ-Prox – Approximation of Proximal Operator

```
1: HJ-Prox( $x, t; f, \delta, N, \alpha, \varepsilon$ ):  
2:   for  $i \in [N]$ :  
3:     Sample  $y^i \sim \mathcal{N}(x, \delta t / \alpha)$   
4:      $z_i \leftarrow f(y^i)$   
5:   if  $\exp(-\alpha z_i / \delta) \leq \varepsilon$  for large proportion of samples:  
6:     return HJ-Prox( $x, t; f, \delta, N, \alpha/2, \varepsilon$ )  
7:    $\text{prox} \leftarrow \text{softmax}(-\alpha z / \delta)^\top [y^1 \cdots y^N]$   
8:   return prox
```



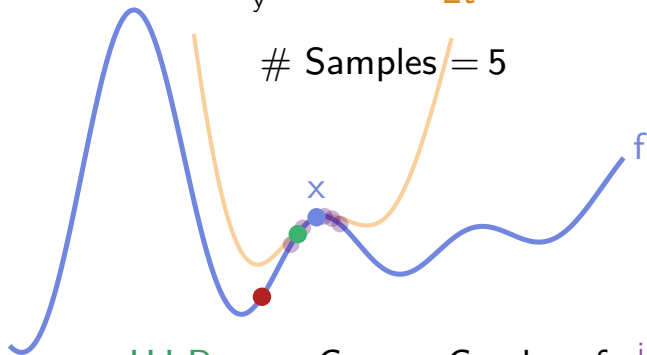
$$\text{prox}_{\text{tf}}(\textcolor{blue}{x}) = \underset{y}{\operatorname{argmin}} \textcolor{brown}{f}(y) + \frac{1}{2t} \|y - \textcolor{blue}{x}\|^2$$





$$\text{prox}_{\text{tf}}(\textcolor{blue}{x}) = \underset{y}{\operatorname{argmin}} \textcolor{brown}{f}(y) + \frac{1}{2t} \|y - \textcolor{blue}{x}\|^2$$

Samples = 5

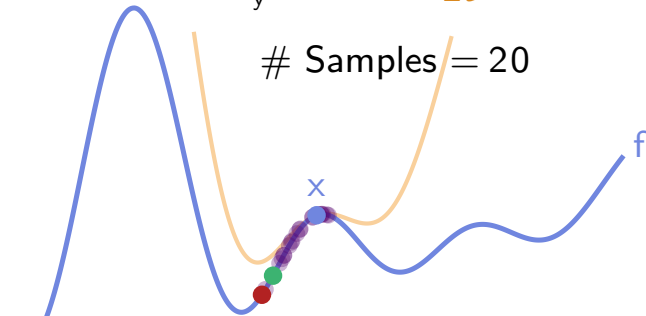


HJ-Prox = Convex Combo of y^i
 y^i weight = $\text{softmax}(-f(y)/\delta)$



$$\text{prox}_{\text{tf}}(\textcolor{blue}{x}) = \underset{y}{\operatorname{argmin}} \textcolor{brown}{f}(y) + \frac{1}{2t} \|y - \textcolor{blue}{x}\|^2$$

Samples = 20

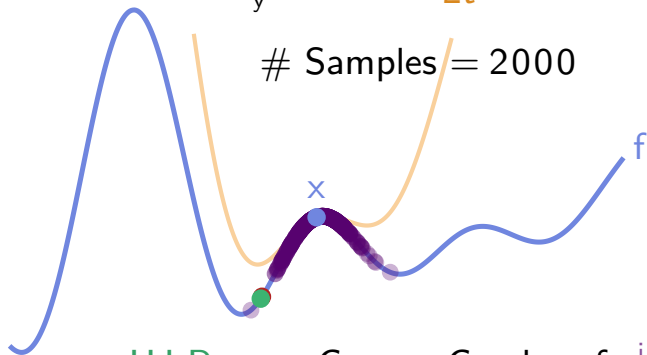


HJ-Prox = Convex Combo of y^i
 y^i weight = $\text{softmax}(-f(y)/\delta)$



$$\text{prox}_{\text{tf}}(\textcolor{blue}{x}) = \underset{y}{\operatorname{argmin}} \textcolor{brown}{f}(y) + \frac{1}{2t} \|y - \textcolor{blue}{x}\|^2$$

Samples = 2000



HJ-Prox = Convex Combo of y^i
 y^i weight = $\text{softmax}(-f(y)/\delta)$



Consider a constrained minimization problem where objective values f can only be accessed via a noisy oracle \mathcal{O} , *i.e.*

$$\min_{x \in \mathbb{R}^{1000}} \mathbb{E}[\mathcal{O}(x)] \quad \text{s.t.} \quad Ax = b,$$

where the expectation is over the noise. We assume A is large and under-determined.

Using HJ-Prox, we solve with the linearized method of multipliers via updates

$$\begin{aligned} x^{k+1} &= \text{prox}_{t\mathcal{O}} \left(x^k - tA^\top (u^k + \lambda(Ax^k - b)) \right) \\ u^{k+1} &= u^k + \lambda(Ax^{k+1} - b). \end{aligned}$$

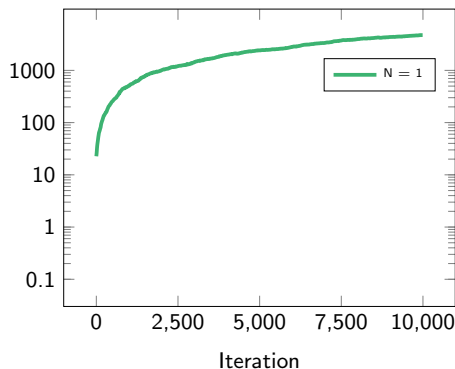
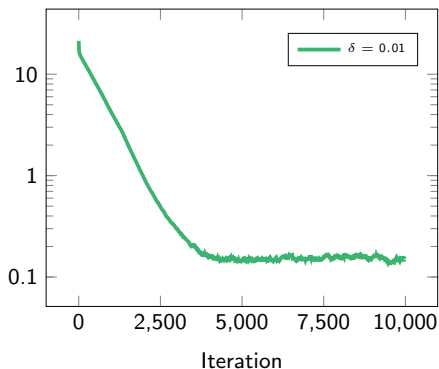


Figure 1: Plots of linearized method of multipliers using HJ-Prox estimates for each proximal. Gradient descent on constraint violation is provided for reference and does not use the oracle \mathcal{O} .

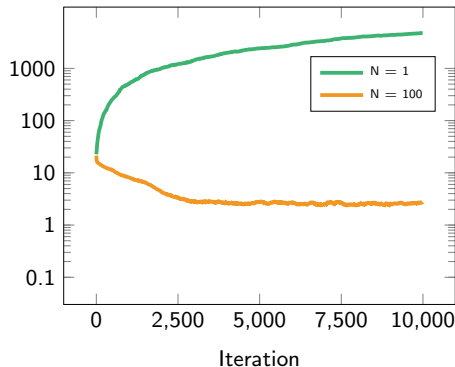
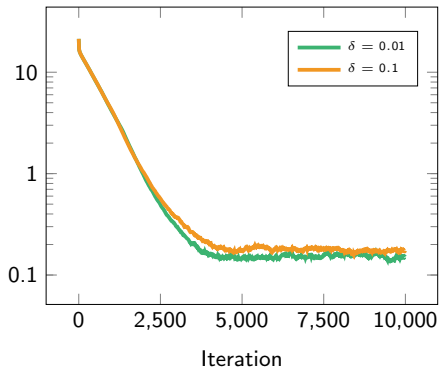


Figure 2: Plots of linearized method of multipliers using HJ-Prox estimates for each proximal. Gradient descent on constraint violation is provided for reference and does not use the oracle \mathcal{O} .

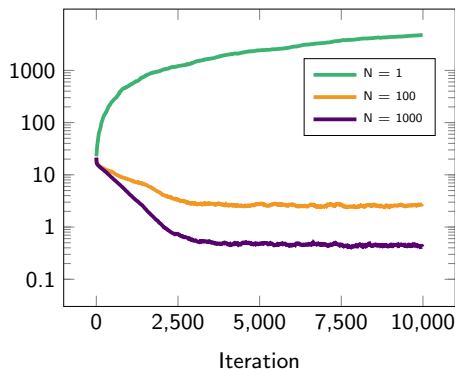
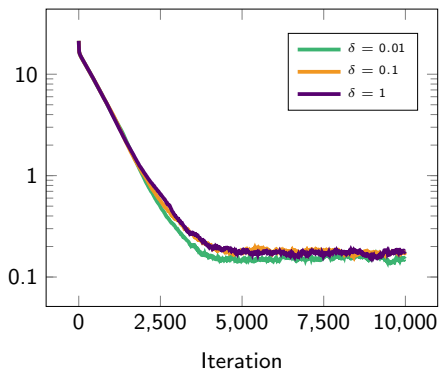


Figure 3: Plots of linearized method of multipliers using HJ-Prox estimates for each proximal. Gradient descent on constraint violation is provided for reference and does not use the oracle \mathcal{O} .

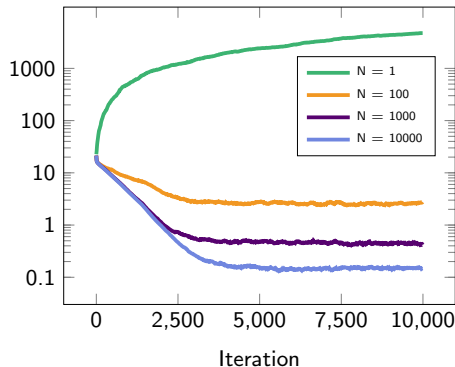
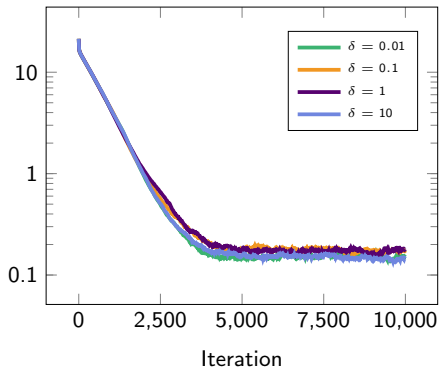


Figure 4: Plots of linearized method of multipliers using HJ-Prox estimates for each proximal. Gradient descent on constraint violation is provided for reference and does not use the oracle \mathcal{O} .

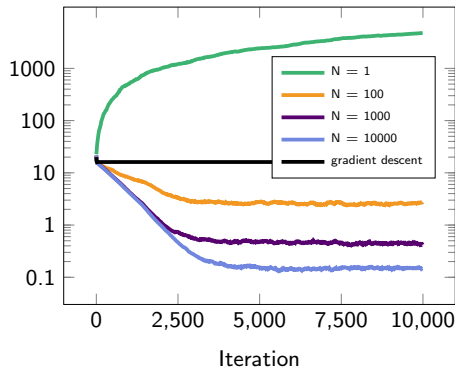
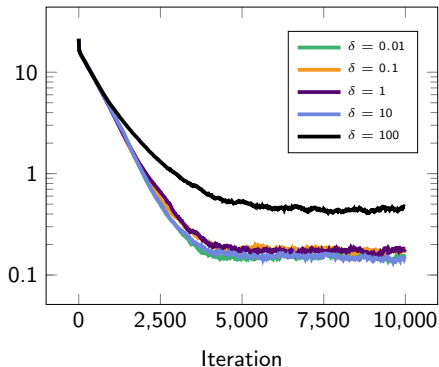


Figure 5: Plots of linearized method of multipliers using HJ-Prox estimates for each proximal. Gradient descent on constraint violation is provided for reference and does not use the oracle \mathcal{O} .



- HJ-Prox gives a simple zeroth-order approximation to proximals
- The parameter δ smooths approximations (potentially helpful for denoising)
- HJ-Prox can be embedded inside optimization algorithms
(e.g. proximal gradient, Douglas Rachford, ADMM, PDHG)

Preprint: arxiv.org/abs/2211.12997

Reprint: <https://www.pnas.org/doi/10.1073/pnas.2220469120>