

JAX-DIPS: Neural bootstrapping of finite discretization methods and application to elliptic problems with discontinuities

Pouria A. Mistani^{* †,a}, Samira Pakravan^{†,b}, Frederic Gibou^b

^aNVIDIA Corporation, Santa Clara, CA 95051, USA

^bUniversity of California, Santa Barbara, CA 93106-5070, USA

Abstract

...

Keywords: level-set method, free boundary problems, surrogate models, jump conditions, differentiable programming, neural networks

1. Level-set method

The level-set method for solving free boundary problems was introduced by Osher & Sethian (1988) [?]. The free boundary is described as the zero contour of a signed-distance function, ϕ , whose evolution is given by the advection equation according to some velocity field dictated by the physics of the problem, \mathbf{v} , that is defined over the moving boundary

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \quad (1)$$

This *implicit* representation of the moving boundaries resolves the need for the challenging task of adapting the underlying grid to the yet-unknown discontinuities in the solution field. The computational simplicity of using Cartesian grids for solving free boundary problems with irregular geometries, as well as the ability to simulate freely moving discontinuities in a *sharp*-manner, that is required by the physics of these problems, are the two main offerings of the level-set method for this class of PDE problems.

The signed-distance property of the level-set function, *i.e.* $|\nabla \phi| = 1$, deteriorates with timestepping the discretized advection equation. This is resolved by solving the Sussman reinitialization equation [?] in fictitious time (τ) every few iterations of the physical time (t),

$$\frac{\partial \phi}{\partial \tau} + \text{sgn}(\phi_0)(|\nabla \phi| - 1) = 0 \quad (2)$$

Note the asymptotic solution of the Sussman equation is the signed-distance property of the level-set function. Here $\text{sgn}(\phi_0)$ is the sign function evaluated on the level-set function before reinitialization in order to preserve the sign of grid points during the fictitious updates.

Besides implicit representation of the free boundaries, the level-set function can be used to compute normal vectors to the interface

$$\mathbf{n} = \nabla \phi / |\nabla \phi|$$

as well as the curvature of the interface

$$\kappa = \nabla \cdot \mathbf{n}$$

Below we describe numerical methods implemented in JAX-DIPS to solve the level-set equations.

^{*}Corresponding author: pmistani@nvidia.com

[†]These authors contributed equally to this work

1.1. Second order accurate semi-Lagrangian advection scheme

Semi-Lagrangian methods are unconditionally stable for solving advection equations, therefore avoiding the restrictive CFL condition on the timestep size from time t^n to t^{n+1} . The general procedure to solve for $\phi^{n+1}(\mathbf{x}^{n+1})$ is to first evaluate for each grid point, \mathbf{x}^{n+1} , the departure point, \mathbf{x}_d , in the upwind direction along the characteristic curve; then use interpolation methods to recover value of the solution field at the departure point; and finally updating the solution field by setting $\phi^{n+1}(\mathbf{x}^{n+1}) = \phi^n(\mathbf{x}_d)$.

The task of evaluating departure points is treated by the second-order mid-point method [?]

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{x}^{n+1} - \frac{\Delta t}{2} \cdot \mathbf{v}^n(\mathbf{x}^{n+1}) \\ \mathbf{x}_d &= \mathbf{x}^{n+1} - \Delta t \cdot \mathbf{v}^{n+\frac{1}{2}}(\hat{\mathbf{x}})\end{aligned}$$

The first step is trivial, but the second step demands velocity at the mid-timestep $t^{n+\frac{1}{2}}$ which can be evaluated from the velocity field at the previous two timesteps, t^n and t^{n-1} , according to

$$\mathbf{v}^{n+\frac{1}{2}} = \frac{3}{2}\mathbf{v}^n - \frac{1}{2}\mathbf{v}^{n-1}$$

this intermediate velocity field is then fed into a trilinear interpolation scheme to sample the velocities on the intermediate points $\mathbf{v}^{n+\frac{1}{2}}(\hat{\mathbf{x}})$. At this point departure points can be computed, and the advected solution is updated by sampling a nonoscillatory interpolant of the level-set function at timestep t^n over the computed departure points $\phi^n(\mathbf{x}_d)$.

1.2. Godunov Hamiltonian for reinitialization

The Sussman equation 2 is generically discretized over its spatial dimensions to obtain

$$\frac{d\phi}{d\tau} = -\text{sgn}(\phi^0) \left[H_G(D_x^+ \phi, D_x^- \phi, D_y^+ \phi, D_y^- \phi, D_z^+ \phi, D_z^- \phi) - 1 \right] \quad (3)$$

where H_G is the so-called Godunov Hamiltonian defined as

$$\begin{aligned}H_G(a, b, c, d, e, f) &= \sqrt{\sum_{(i,j) \in (a,b), (c,d), (e,f)} \max \left(|\min(s \cdot i, 0)|^2, |\max(s \cdot j, 0)|^2 \right)} \\ s &= \text{sgn}(\phi^0)\end{aligned}$$

where the one-sided derivatives are computed using second order accurate discretizations in the bulk far from interfaces

$$\begin{aligned}(D_x^+ \phi)_{i,j,k} &= \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x} - \frac{\Delta x}{2} \text{minmod}((D_{xx}\phi)_{i,j,k}, (D_{xx}\phi)_{i+1,j,k}) \\ (D_x^- \phi)_{i,j,k} &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x} + \frac{\Delta x}{2} \text{minmod}((D_{xx}\phi)_{i,j,k}, (D_{xx}\phi)_{i-1,j,k})\end{aligned}$$

where we used $\text{minmod}(a, b) \triangleq \text{median}(a, 0, b)$ slope-limiter [?]; and in the vicinity of interfaces these derivatives take into account distance to the interface using the level-set function. With similar results along y and z axes, the derivative along the positive x -axis is given by

$$\begin{aligned}(D_x^+ \phi)_{i,j,k} &= \frac{0 - \phi_{i,j,k}}{s_I} - \frac{s_I}{2} c_2^+ \\ s_I &= \frac{\Delta x}{2} + \begin{cases} -c_0^+ / c_1^+ & \text{if } |c_2^+| < \epsilon \\ \left(-c_1^+ - \text{sgn}(\phi_{i,j,k}^0) \sqrt{(c_1^+)^2 - 4c_2^+ c_0^+} \right) / (2c_2^+) & \text{if } |c_2^+| \geq \epsilon \end{cases}\end{aligned}$$

where

$$\begin{aligned} c_2^+ &= \minmod((D_{xx}\phi)_{i,j,k}, (D_{xx}\phi)_{i+1,j,k}) \\ c_1^+ &= \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x} \\ c_0^+ &= \frac{\phi_{i+1,j,k} + \phi_{i,j,k}}{2} - \frac{c_2^+(\Delta x)^2}{4} \end{aligned}$$

And along the negative x -axis we obtain

$$\begin{aligned} (D_x^- \phi)_{i,j,k} &= \frac{\phi_{i,j,k} - 0}{s_I} + \frac{s_I}{2} c_2^- \\ s_I &= \frac{\Delta x}{2} + \begin{cases} c_0^-/c_1^- & \text{if } |c_2^-| < \epsilon \\ \left(c_1^- - \text{sgn}(\phi_{i,j,k}^0) \sqrt{(c_1^-)^2 - 4c_2^- c_0^-} \right) / (2c_2^-) & \text{if } |c_2^-| \geq \epsilon \end{cases} \end{aligned}$$

where

$$\begin{aligned} c_2^- &= \minmod((D_{xx}\phi)_{i,j,k}, (D_{xx}\phi)_{i-1,j,k}) \\ c_1^- &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x} \\ c_0^- &= \frac{\phi_{i-1,j,k} + \phi_{i,j,k}}{2} - \frac{c_2^-(\Delta x)^2}{4} \end{aligned}$$

Having the right-hand-side, the TVD-RK2 method of Shu & Osher [?] is applied to the update the level-set function in fictitious time

$$\begin{aligned} \tilde{\phi}^{n+1} &= \phi^n - \Delta\tau \cdot \text{sgn}(\phi^0)[H_G^n - 1] \\ \tilde{\phi}^{n+2} &= \tilde{\phi}^{n+1} - \Delta\tau \cdot \text{sgn}(\phi^0)[\tilde{H}_G^{n+1} - 1] \\ \phi^{n+1} &= \frac{\phi^n + \tilde{\phi}^{n+2}}{2} \end{aligned}$$

with a timestep chosen by $\Delta\tau = \min(s_I, \Delta x, \Delta y, \Delta z)/3$.

Acknowledgement