Discrete adjoint method with applications to scalar hyperbolic PDE networks

Jack Reilly

University of California Berkeley

jackdreilly@berkeley.edu

May 21, 2013

Overview

Overview

Optimization of a PDE-constrained system

Optimization problem

$$\begin{aligned} \text{minimize}_{u \in \mathcal{U}} \quad & J(x, u) \\ \text{subject to} \quad & H(x, u) = 0 \end{aligned}$$

- $x \in \mathcal{X} \subseteq \mathbb{R}^n$: state variables
- $u \in \mathcal{U} \subseteq \mathbb{R}^m$: control variables

$$\begin{aligned} J : \mathcal{X} \times \mathcal{U} &\to \mathbb{R} \\ (x, u) &\mapsto J(x, u) \end{aligned}$$

$$\begin{aligned} H : \mathcal{X} \times \mathcal{U} &\to \mathbb{R}^{n_H} \\ (x, u) &\mapsto H(x, u) \end{aligned}$$

Want to do gradient descent. How to compute the gradient?

Example: linear system

Discrete linear dynamics

$$x_{t+1} = Ax_t + Bu_t,\ \ t \in \{0, \dots, T-1\}$$

with initial condition x_0.

Let

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_T \end{bmatrix} \qquad\qquad u = \begin{bmatrix} u_0 \\ \vdots \\ u_{T-1} \end{bmatrix}$$

Example: linear system

$$x = \begin{bmatrix} Ax_0 + Bu_0 \\ Ax_1 + Bu_1 \\ \vdots \\ Ax_{T-1} + Bu_{T-1} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & & & \\ A & \ddots & & \\ & \ddots & \ddots & \\ & & A & 0 \end{bmatrix} x + \begin{bmatrix} B & & \\ & \ddots & \\ & & \ddots & \\ & & & B \end{bmatrix} u + \begin{bmatrix} Ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Can be written as

$$(\tilde{A} - I)x + \tilde{B}u + c = 0$$

Note: $(\tilde{A} - I)$ is invertible (lower triangular, with -1 on diagonal). Good: system is deterministic!

Example: linear system

Linear system

$$H_x x + H_u u + c = 0$$

- $x \in \mathbb{R}^n$ state
- $u \in \mathbb{R}^m$ control, with $m \leq n$
- $H_x \in \mathbb{R}^{n \times n}$, assume invertible
- $H_u \in \mathbb{R}^{n \times m}$
- $c \in \mathbb{R}^n$

want to minimize linear cost function

$$\begin{array}{ll} \text{minimize}_{u \in \mathcal{U}} & J_x x + J_u u \\ \text{subject to} & H_x x + H_u u + c = 0 \end{array}$$

$J_x \in \mathbb{R}^{1 \times n}$ and $J_u \in \mathbb{R}^{1 \times m}$ are given row vectors.

Example: linear system

Optimization problem

$$\begin{aligned} &\text{minimize}_{u\in\mathcal{U}} && J_x x + J_u u \\ &\text{subject to} && H_x x + H_u u + c = 0 \end{aligned}$$

An equivalent problem is

$$\text{minimize}_{u\in\mathcal{U}} - J_x H_x^{-1}(H_u u + c) + J_u u$$

and the gradient is

Gradient

$$\nabla_u J = -J_x H_x^{-1} H_u + J_u$$

Example: linear system

Gradient

$$\nabla_u J = -J_x H_x^{-1} H_u + J_u$$

Two ways to compute the first term

Forward

$$J_x M$$

$$H_x M = -H_u$$

Solve for $M \in \mathbb{R}^{n \times m}$: m inversions

$$H_x \left[\begin{array}{c|c|c} M_1 & \dots & M_m \end{array} \right] = \left[\begin{array}{c|c|c} H_{u_1} & \dots & H_{u_m} \end{array} \right]$$

Cost $O(mn^2)$.
Then product $1 \times n$ times $n \times m$: $O(nm)$

Adjoint

$$\lambda^T H_u$$

$$\lambda^T H_x = -J_x$$

Solve for $\lambda \in \mathbb{R}^n$: 1 inversion

$$H_x^T \lambda = J_x^T$$

Cost $O(n^2)$.
Then product $1 \times n$ times $n \times m$: $O(nm)$

Optimization of a PDE-constrained system

General problem

$$\begin{aligned} \text{minimize}_{u \in \mathcal{U}} \quad & J(x, u) \\ \text{subject to} \quad & H(x, u) = 0 \end{aligned}$$

$$\nabla_u J = \frac{\partial J}{\partial x} \nabla_u x + \frac{\partial J}{\partial u}$$

On trajectories, $H(x, u) = 0$ constant, thus $\nabla_u H = 0$

$$\frac{\partial H}{\partial x} \nabla_u x + \frac{\partial H}{\partial u} = 0$$

Linear system

$$\begin{aligned} \text{minimize}_{u \in \mathcal{U}} \quad & J_x x + J_u u \\ \text{subject to} \quad & H_x x + H_u u + c = 0 \end{aligned}$$

$$\nabla_u J = J_x M + J_u$$

$$H_x M = -H_u$$

Optimization of a PDE-constrained system

General problem

$$\begin{array}{ll} \text{minimize}_{u \in \mathcal{U}} & J(x,u) \\ \text{subject to} & H(x,u) = 0 \end{array}$$

$$\nabla_u J = \frac{\partial J}{\partial x} \nabla_u x + \frac{\partial J}{\partial u}$$

On trajectories, $H(x,u) = 0$ constant, thus $\nabla_u H = 0$

$$\frac{\partial H}{\partial x} \nabla_u x + \frac{\partial H}{\partial u} = 0$$

Linear system

$$\begin{array}{ll} \text{minimize}_{u \in \mathcal{U}} & J_x x + J_u u \\ \text{subject to} & H_x x + H_u u + c = 0 \end{array}$$

$$\nabla_u J = J_x M + J_u$$

$$H_x M = -H_u$$

Instead, solve for $\lambda \in \mathbb{R}^n$

Adjoint

$$H_x^T \lambda = J_x^T$$

Optimization of a PDE-constrained system

General problem

$$\begin{aligned} \text{minimize}_{u \in \mathcal{U}} \quad & J(x,u) \\ \text{subject to} \quad & H(x,u) = 0 \end{aligned}$$

$$\nabla_u J = \frac{\partial J}{\partial x} \nabla_u x + \frac{\partial J}{\partial u}$$

On trajectories, $H(x,u) = 0$ constant, thus $\nabla_u H = 0$

$$\frac{\partial H}{\partial x} \nabla_u x + \frac{\partial H}{\partial u} = 0$$

Adjoint

$$\frac{\partial H}{\partial x}^T \lambda = \frac{\partial J}{\partial x}$$

Linear system

$$\begin{aligned} \text{minimize}_{u \in \mathcal{U}} \quad & J_x x + J_u u \\ \text{subject to} \quad & H_x x + H_u u + c = 0 \end{aligned}$$

$$\nabla_u J = J_x M + J_u$$

$$H_x M = -H_u$$

Instead, solve for $\lambda \in \mathbb{R}^n$

Adjoint

$$H_x^T \lambda = J_x^T$$

Computing $\nabla_u J(x, u)$

Want to evaluate

$$\frac{\partial J}{\partial x} \nabla_u x$$

$$\text{where } \frac{\partial H}{\partial x} \nabla_u x + \frac{\partial H}{\partial u} = 0$$

Computing $\nabla_u J(x, u)$

Want to evaluate

$$\frac{\partial J}{\partial x} \nabla_u x$$
$$\text{where } \frac{\partial H}{\partial x} \nabla_u x + \frac{\partial H}{\partial u} = 0$$

If λ is solution to the adjoint equation

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0$$

Computing $\nabla_u J(x, u)$

Want to evaluate

$$\frac{\partial J}{\partial x} \nabla_u x$$

$$\text{where } \frac{\partial H}{\partial x} \nabla_u x + \frac{\partial H}{\partial u} = 0$$

If λ is solution to the adjoint equation

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0$$

Then

$$\frac{\partial J}{\partial x} \nabla_u x = -\lambda^T \frac{\partial H}{\partial x} \nabla_u x = \lambda^T \frac{\partial H}{\partial u}$$

Adjoint solution λ

$$\nabla_u J = \lambda^T \frac{\partial H}{\partial u} + \frac{\partial J}{\partial u}$$

Also useful for sensitivity analysis.

Sensitivity analysis

λ_k is the price of changing H_k

Optimization algorithm

Algorithm 1 Gradient descent loop

Pick initial control u^{init}

while not converged **do**

  $x = forwardSim(u, IC, BC)$ solve for state trajectory (forward system)

  $\lambda = adjointSln(x, u)$ solve for adjoint parameters (adjoint system)

  $\Delta u = \nabla_u J = \lambda^T \frac{\partial H}{\partial u} + \frac{\partial J}{\partial u}$ Compute the gradient (search direction)

  $u \leftarrow u + t\Delta u$ update u using line search along Δu

end while

Line search

Example 1: decreasing step size

$$t^{(k)} = t^{(1)}/k$$

Example 2: backtracking line-search

- fix parameters $0 < \alpha < 0.1$ and $0 < \beta < 1$
- given search direction Δu

Algorithm 2 Backtracking line search

while $J(u + t\Delta u) - J(u) > \alpha(\nabla_u J)^T(t\Delta u)$ **do**
    $t \leftarrow \beta t$
end while

Constraints on control

What if there are physical constraints on the permissible control values u?

$$u_{\min} \leq u \leq u_{\max} \tag{1}$$

Barrier functions

Lecture 9 slides....

Overview

Hyperbolic PDE's

A conservation law in one space dimension can written in the form:

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial f(\rho(t,x))}{\partial x} = 0 \tag{2}$$

Example

We will constantly refer to the example with linear flux function:

$$\rho_t + a\rho_x = 0 \tag{3}$$

A Cauchy problem wants solution to:

$$\begin{cases} \rho_t + f(\rho)_x & = 0 \\ \rho(0,x) & = \rho_0(x) \end{cases} \tag{4}$$

Riemann problem

Define a **Riemann problem** as a Cauchy problem:

$$\begin{cases} \rho_t + f(\rho)_x & = 0 \\ \rho(0, x) & = \rho_0(x) \end{cases} \tag{5}$$

where:

$$\rho_0(x) = \begin{cases} \rho_l & x < 0 \\ \rho_r & x \geq 0 \end{cases} \tag{6}$$

Example

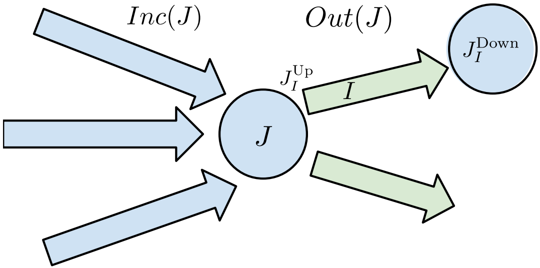
On board...

Overview

Network description

Consider a network of hyperbolic PDE's $(\mathcal{I}, \mathcal{J})$

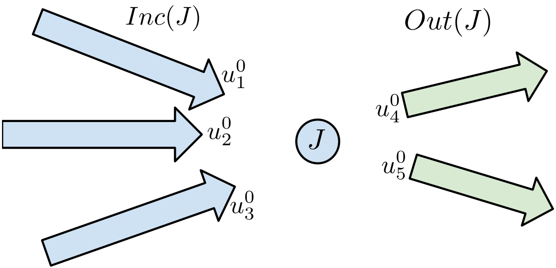
- $i \in \mathcal{I}$ a link with dynamics according to PDE.
- $J \in \mathcal{J}$ a junction with incoming links $Inc(J)$, outgoing links $Out(J)$.



Boundary conditions at junctions?

Riemann problem at junction

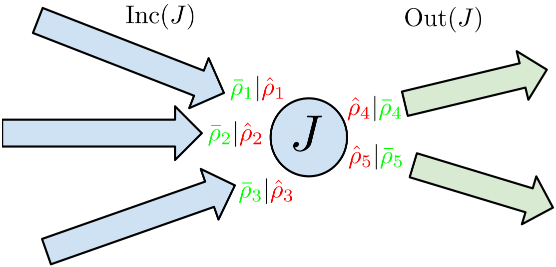
For a junction J, let each link $i \in Inc(J) \cup Out(J)$ have constant IC $\rho_i^0 \in \rho_J$.



Define a **Riemann Solver** *RS*:

$$
\begin{aligned}
RS : \mathbb{R}^{m+n} &\to \mathbb{R}^{m+n} && (7)\\
\rho_J^0 &\mapsto RS\left(\rho_J^0\right) = \hat{\rho}_J && (8)
\end{aligned}
$$

where $\hat{\rho}_J^i \in \hat{\rho}_J$ is the boundary condition at the junction interface for link i.



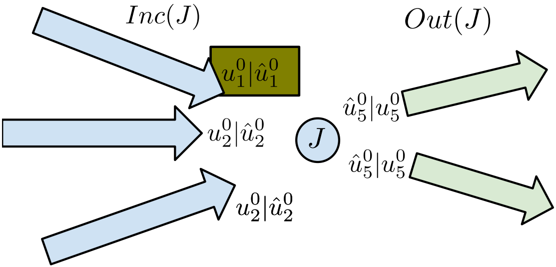
Define a **Riemann Solver** RS:

$$RS : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^{m+n} \quad (7)$$

$$\rho_J^0 \mapsto RS\left(\rho_J^0\right) = \hat{\rho}_J \quad (8)$$

where $\hat{\rho}_J^i \in \hat{\rho}_J$ is the boundary condition at the junction interface for link i.

- Consider a specific link



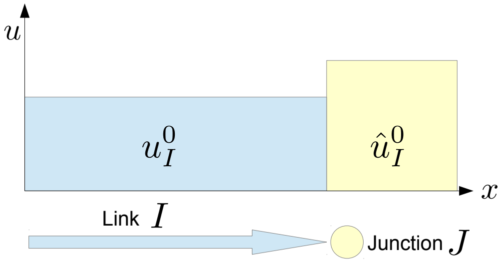
Define a **Riemann Solver** *RS*:

$$RS : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^{m+n} \quad (7)$$

$$\rho_J^0 \mapsto RS\left(\rho_J^0\right) = \hat{\rho}_J \quad (8)$$

where $\hat{\rho}_J^i \in \hat{\rho}_J$ is the boundary condition at the junction interface for link i.

- Consider a specific link



Conditions on Riemann solver

- Self-similar

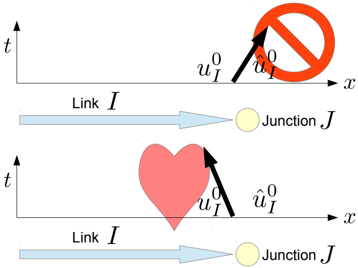
$$RS(\rho_J) = RS(\hat{\rho}_J) = \hat{\rho}_J \tag{9}$$

Conditions on Riemann solver

- Self-similar

$$RS(\rho_J) = RS(\hat{\rho}_J) = \hat{\rho}_J \tag{9}$$

- All shockwaves must emanate outward from junction

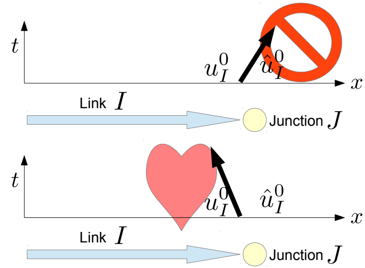


Conditions on Riemann solver

- Self-similar

$$RS(\rho_J) = RS(\hat{\rho}_J) = \hat{\rho}_J \tag{9}$$

- All shockwaves must emanate outward from junction



- Conservation of mass

$$\sum_{i \in Inc(J)} f\left(\{\hat{\rho}_J\}_i\right) = \sum_{j \in Out(J)} f\left(\{\hat{\rho}_J\}_j\right) \tag{10}$$

Riemann solver for linear flux function

Example

On board...

Overview

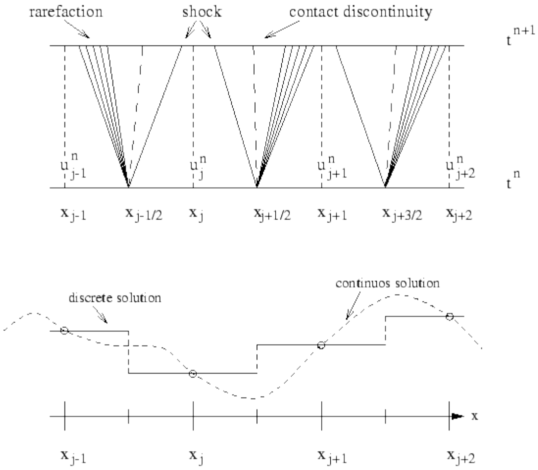
Discretizing via Godunov method

- Cannot represent (or not practical to represent) continuous function on computer.
- Approximate solution by discretizing space and time.
- Solve for vector of discrete variables.

Godunov's scheme (high level)

1. Split system in discrete chunks of size $\triangle x$.
2. Approximate IC by averaging over $\triangle x$.
3. Find exact sln of system by solving Riemann problems at discretized boundaries for $\triangle t$ time.
4. Approximate new sln by averaging over $\triangle x$.
5. Set IC as new sln and go to step 3.

Discretizing single system



Godunov's scheme: local solutions of Riemann problems

Figure: credit: http://www.uv.es/astrorela/simulacionnumerica/node34.html

Derivation of Godunov's method

Take discrete initial condition ρ_i at cell i. We want $\bar{\rho})_i$, the average value at cell i at time $\triangle t$:

$$\bar{\rho}_i = \rho_i - \frac{1}{x_{i+1} - x_i} \int_0^{\triangle t} \left(f\left(u\left(t, x_{i+1}\right)\right) - f\left(u\left(t, x_i\right)\right)\right) dt \qquad (11)$$

This requires solution of $\rho(x, t)$ over $[0, \triangle x] \times [0, \triangle t]$.
But since Riemann problems are self-similar, fluxes across boundaries are constant:

$$\int_0^{\triangle t} f\left(u\left(t, x_i\right)\right) dt \approx \triangle t g^G\left(\rho_i, \rho_{i+1}\right) \qquad (12)$$

where g^G is the flux across cell boundaries obtained via sln of Riemann problem.
Now only function of discrete values:

$$\bar{\rho}_i = \rho_i - \frac{\triangle t}{x_{i+1} - x_i} \left(g^G\left(\rho_i, \rho_{i+1}\right) - g^G\left(\rho_{i-1}, \rho_i\right)\right) \qquad (13)$$

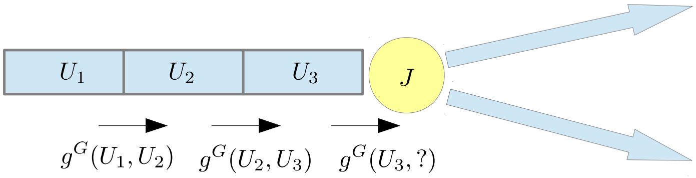
CFL condition

In previous derivation, it is assumed no solution from one Riemann problem influences another at the discrete boundaries.
This limits how large the time-step to guarantee convergence of Godunov scheme to continuous solution.

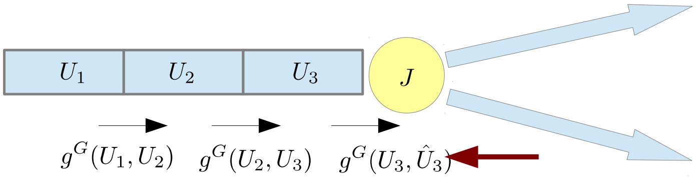
The Courant Friedrichs Lewy (CFL) condition

$$\lambda^{\max} \leq \frac{\triangle x}{\triangle t} \tag{14}$$

Solving for Godunov flux easy for 1-to-1 junctions. What about *n*-to-*m*?



Solving for Godunov flux easy for 1-to-1 junctions. What about *n*-to-*m*?



- Apply Riemann solver at junction
- Use Riemann solution as boundary condition for g^G at junction.

Summary of Godunov scheme for PDE networks

1. Begin with initial condition $(t = 0)$ $\{\rho_i : i \in \mathcal{I}\}$.

Summary of Godunov scheme for PDE networks

1. Begin with initial condition $(t = 0)$ $\{\rho_i : i \in \mathcal{I}\}$.
2. For every junction $J \in \mathcal{J}$:
   2.1 Apply the Riemann solver to Riemann data to obtain the boundary condition $\hat{\rho}_J = RS(\rho_J)$.

Summary of Godunov scheme for PDE networks

1. Begin with initial condition $(t = 0)$ $\{\rho_i : i \in \mathcal{I}\}$.
2. For every junction $J \in \mathcal{J}$:
   2.1 Apply the Riemann solver to Riemann data to obtain the boundary condition $\hat{\rho}_J = RS(\rho_J)$.
3. For every link $i \in \mathcal{I}$:
   3.1 Letting $J_i^{\text{Up}} = J \in \mathcal{J} : i \in Out(J)$ and $J_i^{\text{Down}} = J \in \mathcal{J} : i \in In(J)$, the discrete value over link i at time $\triangle t$, $\bar{\rho}_i$, is given by:

$$\bar{\rho}_i = \rho_i - \frac{\triangle t}{L_i}\left(f\left(\left\{\hat{\rho}_{J_i^{\text{Down}}}\right\}_i\right) - f\left(\left\{\hat{\rho}_{J_i^{\text{Up}}}\right\}_i\right)\right)$$

Summary of Godunov scheme for PDE networks

1. Begin with initial condition $(t = 0)$ $\{\rho_i : i \in \mathcal{I}\}$.
2. For every junction $J \in \mathcal{J}$:
   2.1 Apply the Riemann solver to Riemann data to obtain the boundary condition $\hat{\rho}_J = RS\,(\rho_J)$.
3. For every link $i \in \mathcal{I}$:
   3.1 Letting $J_i^{\text{Up}} = J \in \mathcal{J} : i \in Out\,(J)$ and $J_i^{\text{Down}} = J \in \mathcal{J} : i \in In\,(J)$, the discrete value over link i at time $\triangle t$, $\bar{\rho}_i$, is given by:

$$\bar{\rho}_i = \rho_i - \frac{\triangle t}{L_i}\left(f\left(\left\{\hat{\rho}_{J_i^{\text{Down}}}\right\}_i\right) - f\left(\left\{\hat{\rho}_{J_i^{\text{Up}}}\right\}_i\right)\right)$$

Example

On board...

Overview

PDE with control

Modify formulation to include

- state vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{I}|T}$
- control vector $\tilde{\mathbf{u}} \in \mathbb{R}^{N_{\vec{u}}T}$
  - $\tilde{\mathbf{u}}_{J,k} \subseteq \tilde{\mathbf{u}}_k$ modifies Riemann problem at J for time k.

$$RS : \mathbb{R}^{m+n} \times \mathbb{R}^{|\tilde{\mathbf{u}}_{J,k}|} \to \mathbb{R}^{m+n} \tag{15}$$

$$(\mathbf{x}_{J,k}, \tilde{\mathbf{u}}_{J,k}) \mapsto RS(\mathbf{x}_{J,k}, \tilde{\mathbf{u}}_{J,k}) = \hat{\rho}_{J,k} \tag{16}$$

  - $N_{\vec{u}}$ are the number of control parameters at each time-step.

Updated discrete state equations:

$$h_{i,1}(\mathbf{x}, \mathbf{v}) = x_{i,1} - \rho_i = 0 \tag{17}$$

$$h_{i,k}(\mathbf{x}, \mathbf{v}) = x_{i,k} - x_{i,k-1} + \tag{18}$$

$$\frac{\triangle t}{L_i}(\hat{F}_i(\mathbf{x}_{J_i^{\mathbf{Down}},k-1}, \tilde{\mathbf{u}}_{J_i^{\mathbf{Down}},k-1}) - \hat{F}_i(\mathbf{x}_{J_i^{\mathbf{Up}},k-1}, \tilde{\mathbf{u}}_{J_i^{\mathbf{Up}},k-1})) \tag{19}$$

$$= 0 \forall k \in [2, \ldots, T] \tag{20}$$

Optimization problem

Optimization Problem

$$\min_{\tilde{\mathbf{u}}} J(\mathbf{x}, \tilde{\mathbf{u}}) \tag{21}$$

$$\text{subject to:} H(\mathbf{x}, \tilde{\mathbf{u}}) = 0 \tag{22}$$

Review: adjoint method

$$\nabla J = \lambda^T H_{\vec{u}} + J_{\vec{u}} \tag{23}$$

$$H_x^T \lambda = -J_x \tag{24}$$

Assume initial $\tilde{\mathbf{u}}_0$ and state $H(\mathbf{x}_0, \tilde{\mathbf{u}}_0) = 0$.

What needs to be computed for adjoint method?

- $\frac{\partial J(\mathbf{x_0},\tilde{\mathbf{u}}_\mathbf{0})}{\partial \tilde{\mathbf{u}}}, \frac{\partial J(\mathbf{x_0},\tilde{\mathbf{u}}_\mathbf{0})}{\partial \mathbf{x}}$: Problem specific, no sparsity assumptions.
- $\frac{\partial H(\mathbf{x_0},\tilde{\mathbf{u}}_\mathbf{0})}{\partial \tilde{\mathbf{u}}}, \frac{\partial H(\mathbf{x_0},\tilde{\mathbf{u}}_\mathbf{0})}{\partial \mathbf{x}}$: can analyze properties of PDE networks and Godunov scheme to:
  - derive partial derivative expressions
  - understand sparsity

Partial derivates of state equations

H_x

By chain rule:

$$\frac{\partial h_{i,k}}{\partial x} = -\frac{\partial x_{i,k-1}}{\partial x} + \frac{\triangle t}{L_i}\left(\frac{\partial f}{\partial \left\{\hat{\rho}_{J_i^{\mathbf{Down}},k}\right\}_i}\frac{\partial \left\{\hat{\rho}_{J_i^{\mathbf{Down}},k}\right\}_i}{\partial x} - \frac{\partial f}{\partial \left\{\hat{\rho}_{J_i^{\mathbf{Up}},k}\right\}_i}\frac{\partial \left\{\hat{\rho}_{J_i^{\mathbf{Up}},k}\right\}_i}{\partial x}\right)$$

- Only require f' and partial derivatives on Riemann solvers.
- $\frac{\partial h_{i,k}}{\partial x_{j,l}} = 0$ unless $l = k - 1$.
- $\frac{\partial x_{i,k-1}}{\partial x_{j,l}} = 0$ unless $i = j$.
- $\frac{\partial \left\{\hat{\rho}_{J_i^{\mathbf{Down}},k}\right\}_i}{\partial x_{j,l}} = 0$ unless $j \in J_i^{\text{Down}}$ (same for J_i^{Up}).

Partial derivates of state equations

H_x

- Only require f' and partial derivatives on Riemann solvers.
- $\frac{\partial h_{i,k}}{\partial x_{j,l}} = 0$ unless $l = k - 1$.
- $\frac{\partial x_{i,k-1}}{\partial x_{j,l}} = 0$ unless $i = j$.
- $\frac{\partial \left\{ \hat{\rho}_{J_i^{\mathbf{Down}},k} \right\}_i}{\partial x_{j,l}} = 0$ unless $j \in J_i^{\mathrm{Down}}$ (same for J_i^{Up}).

Thus each partial term is zero unless variable is from previous time-step and adjacent to constraint link.

Partial derivates of state equations

H_v

By chain rule:

$$\frac{\partial h_{i,k}}{\partial v} = \frac{\triangle t}{L_i}\left(\frac{\partial f}{\partial\left\{\hat{\rho}_{J_i^{\textbf{Down}},k}\right\}_i}\frac{\partial\left\{\hat{\rho}_{J_i^{\textbf{Down}},k}\right\}_i}{\partial v} - \frac{\partial f}{\partial\left\{\hat{\rho}_{J_i^{\textbf{Up}},k}\right\}_i}\frac{\partial\left\{\hat{\rho}_{J_i^{\textbf{Up}},k}\right\}_i}{\partial v}\right)$$

Similar arguments to H_x give us that each partial term is zero unless variable is from same time-step and $v \in \mathbf{v}_J^{\text{Down}} \cup \mathbf{v}_J^{\text{Up}}$.

Example for linear flux function

Example

On board...

Complexity of solving gradient

Solving adjoint system

$$H_x^T \lambda = -J_x \tag{25}$$

From previous result, H_x has following properties:

- size $|\mathcal{I}|T \times |\mathcal{I}|T$
- lower triangular
- $card\ H_x = O(|\mathcal{I}|TD_x)$: $D_x = \max_{J \in \mathcal{J}} |Inc\,(J) \cup Out\,(J)\,|$

Efficiently solve λ via backward-substitution in time $O(card\ H_x) = O(|\mathcal{I}|TD_x)$, or **linear in** $|\mathcal{I}T|$.

Complexity of solving gradient

Solving ∇J

$$\nabla J = \lambda^T H_{\vec{u}} + J_{\vec{u}} \tag{26}$$

From previous result, $H_{\vec{u}}$ has following properties:

- size $|\mathcal{I}| T \times N_{\vec{u}} T$
- *card* $H_{\vec{u}} = O(|\mathcal{I}| T D_{\vec{u}})$:
  $D_{\vec{u}} = \max_{v_{i,k} \in \tilde{\mathbf{u}}} \left| \bigcup Inc\,(J) \cup Out\,(J) : \vec{u}_{i,k} \in \tilde{\mathbf{u}}_{J,k} \right|$

Sparse matrix multiplication has total cost $O(D_{\vec{u}} N_{\vec{u}} T)$, typically of smaller order than solving $H_{\vec{u}}$.

Complexity of solving gradient

Total complexity of computing gradient via discrete adjoint

$$O(|\mathbf{x}|D_x + |\tilde{\mathbf{u}}|D_{\vec{u}}) \quad (27)$$

$$\nabla J = \lambda^T H_v + J_v \qquad \cancel{O(|x||v|)}$$

$$O(|v|D_v)$$

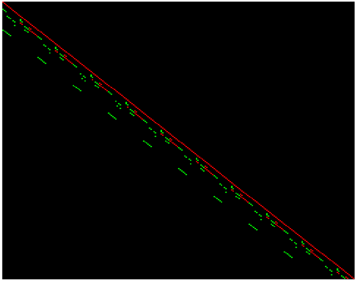
$$H_x^T \lambda = -J_x$$

$$\cancel{O(|x|^3)}$$

$$O(|x|D_x)$$

Overview

Discrete adjoint method applied to ramp metering



$\rho ik - 1 + \frac{\triangle t}{\triangle x}\left(f_i^{\mathbf{in}}(k-1) - f_i^{\mathbf{out}}(k-1)\right)$

$l_i(k-1) + \triangle t\,(D_i(k-1) - r_i(k-1))$

$\min(F_i, v_i \rho ik)$

$\min\left(F_i, w_i\left(\rho_i^{\mathbf{jam}} - \rho ik\right)\right)$

$\min(l_i(k), u_i(k))$

$\min(\delta_{i-\mathbf{1}}(k)(1-\beta_{i-\mathbf{1}}(k)) + d_{i-\mathbf{1}}(k), \sigma_i(k))$

$$f_i^{\mathbf{out}}(k) = \begin{cases} \delta_i(k) & \text{if } P_i f_{i+\mathbf{1}}^{\mathbf{in}}(k) > (1-\beta_i(k))\,\delta_i(k) \\ \frac{f_{i+\mathbf{1}}^{\mathbf{in}}(k) - d_i(k)}{\mathbf{1}-\beta_i(k)} & \text{if } : (1-P_i)\, f_{i+\mathbf{1}}^{\mathbf{in}}(k) > d_i(k) \\ \frac{P_i f_{i+\mathbf{1}}^{\mathbf{in}}(k)}{\mathbf{1}-\beta_i(k)} & \text{otherwise} \end{cases}$$

- H is piecewise affine
- Solving the forward system gives H(x) as a linear system
- $\left.\frac{dH}{dX}\right|_{X=x}$ is computed simultaneously

$$r_i(k) = f_{i+\mathbf{1}}^{\mathbf{in}}(k) - f_i^{\mathbf{out}}(k)(1-\beta_i(k))$$