Discrete adjoint method with applications to PDE networks and ramp metering

Jack Reilly

University of California Berkeley

jackdreilly@berkeley.edu

July 18, 2013

Overview

Overview

Optimization of a PDE-constrained system

Optimization problem

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

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

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

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

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

Example: linear system

Discrete linear dynamics

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

with initial condition x_0.

Let

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

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

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

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

Example: linear system

Optimization problem

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

An equivalent problem is

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

and the gradient is

Gradient

$$\nabla_u C = -C_x H_x^{-1} H_u + C_u$$

Example: linear system

Gradient

$$\nabla_u C = -C_x H_x^{-1} H_u + C_u$$

Two ways to compute the first term

Forward

$$C_x M$$
$$H_x M = -H_u$$

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

$$\lambda^T H_u$$
$$\lambda^T H_x = -C_x$$

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

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

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

Optimization of a PDE-constrained system

General problem

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

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

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

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

$$\nabla_u C = C_x M + C_u$$

$$H_x M = -H_u$$

Optimization of a PDE-constrained system

General problem

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

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

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

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

$$\nabla_u C = C_x M + C_u$$

$$H_x M = -H_u$$

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

Adjoint

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

Optimization of a PDE-constrained system

General problem

$$\text{minimize}_{u\in\mathcal{U}} \quad C(x,u)$$
$$\text{subject to} \quad H(x,u)=0$$

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

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

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

Linear system

$$\text{minimize}_{u\in\mathcal{U}} \quad C_x x + C_u u$$
$$\text{subject to} \quad H_x x + H_u u + c = 0$$

$$\nabla_u C = C_x M + C_u$$

$$H_x M = -H_u$$

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

Adjoint

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

Computing $\nabla_u C(x, u)$

Want to evaluate

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

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

Computing $\nabla_u C(x, u)$

Want to evaluate

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

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

If λ is solution to the adjoint equation

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

Computing $\nabla_u C(x, u)$

Want to evaluate

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

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

If λ is solution to the adjoint equation

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

Then

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

Adjoint solution λ

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

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)

  $\Delta u = \nabla_u C = \lambda^T \frac{\partial H}{\partial u} + \frac{\partial C}{\partial u}$ 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 $C(u + t\Delta u) - C(u) > \alpha(\nabla_u C)^T(t\Delta u)$ **do**
    $t \leftarrow \beta t$
end while

Other descent methods

- General purpose nonlinear solvers.
  - fmincon
  - ipopt
- Attempt to find **global** solutions, rather than terminating in a local minima.
- Often use quasi-Newton methods to estimate second-order information.

Constraints on control

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

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

Barrier functions

$$\tilde{C}\left(\vec{\rho}, \vec{u}, \epsilon\right) = C\left(\vec{\rho}, \vec{u}\right) - \epsilon \sum_{u \in \vec{u}} \log\left(\left(u_{\text{max}} - u\right)\left(u - u_{\text{min}}\right)\right) \tag{2}$$

Then have $\epsilon \in \mathbb{R}^{+}$ tend to zero.

Overview

Hyperbolic PDE's

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

$$\rho_t + f(\rho)_x = 0 \tag{3}$$

A Cauchy problem specifies an initial condition:

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

Riemann problem

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

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

where:

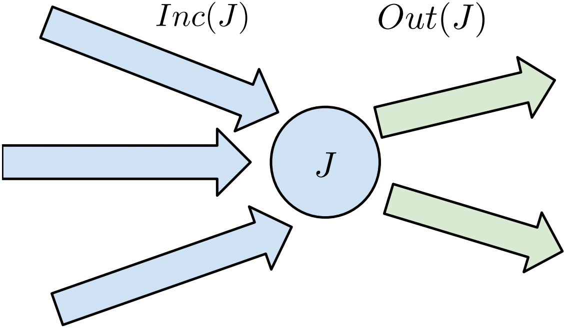
$$\bar{\rho}(x) = \begin{cases} \rho_- & x < \bar{x} \\ \rho_+ & x \geq \bar{x} \end{cases}$$

Overview

Network description

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

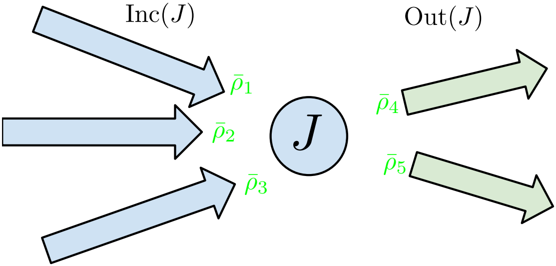
- $i \in \mathcal{I}$ a link with dynamics according to PDE.
- $J \in \mathcal{J}$ a junction with incoming links $Inc(J)$, outgoing links $Out(J)$.



Boundary conditions at junctions?

Riemann problem at junction

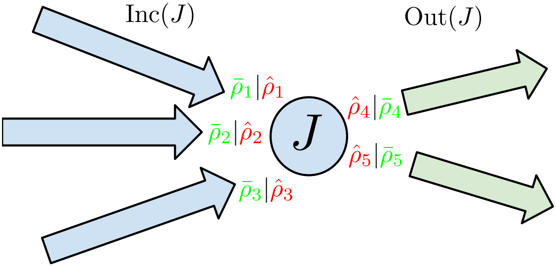
For a junction J, let each link $i \in Inc(J) \cup Out(J)$ have constant IC $\rho_i^0 \in \rho_J$.



Define a **Riemann Solver** RS:

$$RS : \quad \mathbb{R}^{m+n} \to \mathbb{R}^{m+n}$$
$$(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) \mapsto RS(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) = (\hat{\rho}_1, \ldots, \hat{\rho}_{n+m})$$

where $\hat{\rho}_i$ provides the trace for link i for $t \geq 0$.

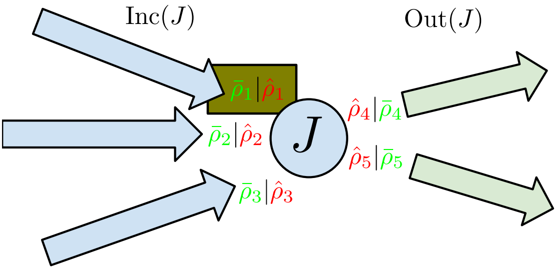


Define a **Riemann Solver** RS:

$$RS: \quad \mathbb{R}^{m+n} \to \mathbb{R}^{m+n}$$
$$(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) \mapsto RS(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) = (\hat{\rho}_1, \ldots, \hat{\rho}_{n+m})$$

where $\hat{\rho}_i$ provides the trace for link i for $t \geq 0$.

- Consider a specific link

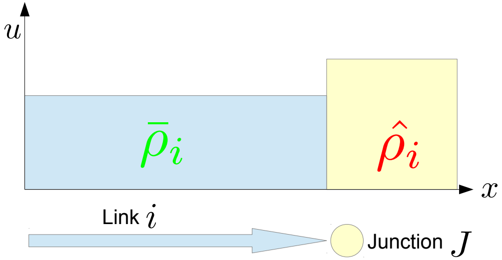


Define a **Riemann Solver** RS:

$$\begin{aligned} RS: \quad \mathbb{R}^{m+n} &\to \mathbb{R}^{m+n} \\ (\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) &\mapsto RS(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) = (\hat{\rho}_1, \ldots, \hat{\rho}_{n+m}) \end{aligned}$$

where $\hat{\rho}_i$ provides the trace for link i for $t \geq 0$.

- Consider a specific link



Conditions on Riemann solver

- Self-similar

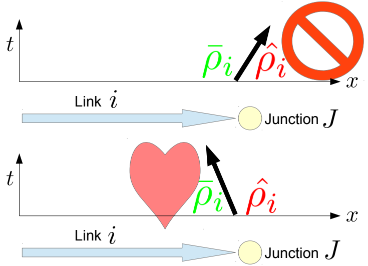
$$RS\left(RS\left(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}\right)\right) = RS\left(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}\right) = \left(\hat{\rho}_1, \ldots, \hat{\rho}_{n+m}\right)$$

Conditions on Riemann solver

- Self-similar

$$RS\left(RS\left(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}\right)\right) = RS\left(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}\right) = \left(\hat{\rho}_1, \ldots, \hat{\rho}_{n+m}\right)$$

- All shockwaves must emanate outward from junction

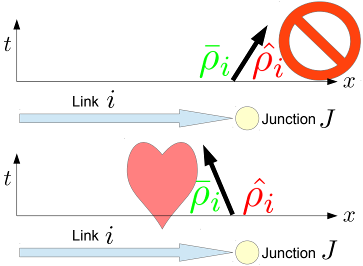


Conditions on Riemann solver

- Self-similar

$$RS\left(RS\left(\bar{\rho}_1,\ldots,\bar{\rho}_{n+m}\right)\right) = RS\left(\bar{\rho}_1,\ldots,\bar{\rho}_{n+m}\right) = \left(\hat{\rho}_1,\ldots,\hat{\rho}_{n+m}\right)$$

- All shockwaves must emanate outward from junction



- Conservation of mass

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

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 Δx.
2. Approximate IC by averaging over Δx.
3. Find exact sln of system by solving Riemann problems at discretized boundaries for Δt time.
4. Approximate new sln by averaging over Δx.
5. Set IC as new sln and go to step 3.

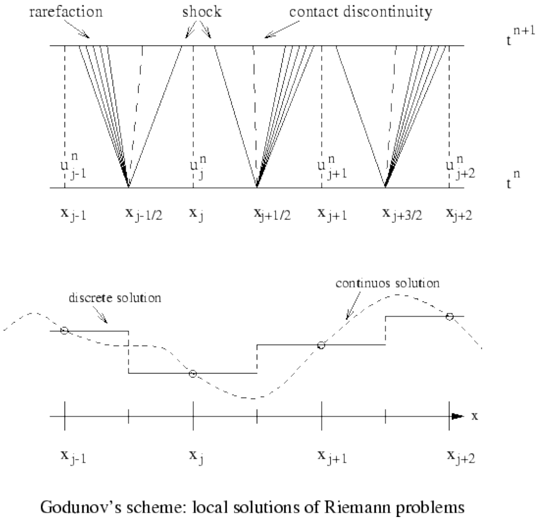


Figure: credit: http://www.uv.es/astrorela/simulacionnumerica/node34.html

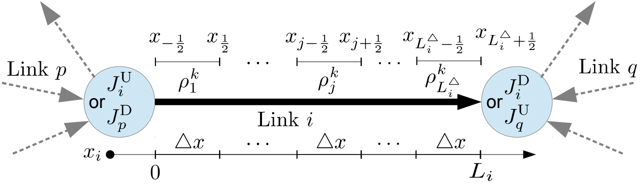
Derivation of Godunov's method

Take a discrete initial condition $\bar{\rho}^{\Delta}$. We want $\rho_j^{\Delta t}$, the average value at cell j at time Δt:

$$\rho_j^{\Delta t} \approxeq \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \rho^{\Delta}(\Delta t, x) dx. \tag{6}$$

where $\rho^{\Delta}(\Delta t, x)$ is an exact solution of the Cauchy problem with initial condition $\bar{\rho}^{\Delta}$.

This requires solution of $\rho(x, t)$ over $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [0, \Delta t]$.

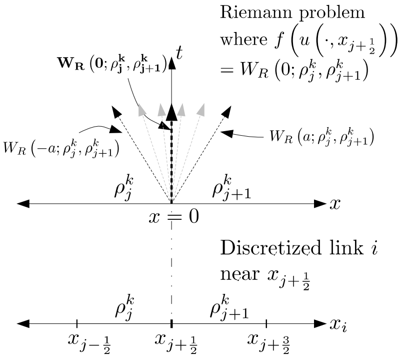


(a) Space discretization for a link $i \in \mathcal{I}$

But since Riemann problems are self-similar, fluxes across boundaries are constant:

$$f(\rho(t, x_{j+\frac{1}{2}})) = f(W_R(0; \rho_j^k, \rho_{j+1}^k)).$$

$$f(\rho(t, x_{j-\frac{1}{2}})) = f(W_R(0; \rho_{j-1}^k, \rho_j^k)).$$

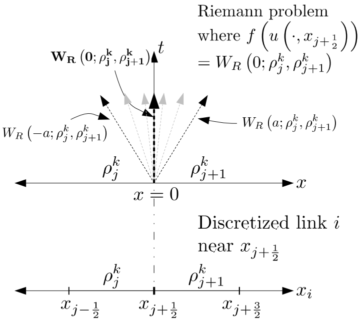


$$\rho_j^{k+1} = \rho_j^k - \frac{\Delta t}{\Delta x}\left(g^G(\rho_j^k, \rho_{j+1}^k) - g^G(\rho_{j-1}^k, \rho_j^k)\right), \tag{7}$$

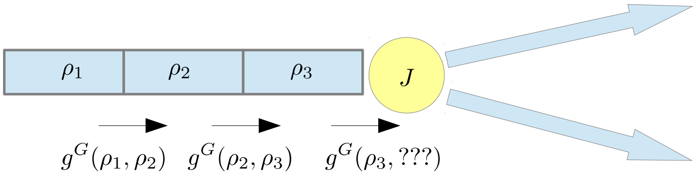
where g^G is the numerical flux:

$$g^G\left(\rho_j, \rho_{j+1}\right) = f(W_R(0; \rho_j, \rho_{j+1}))$$

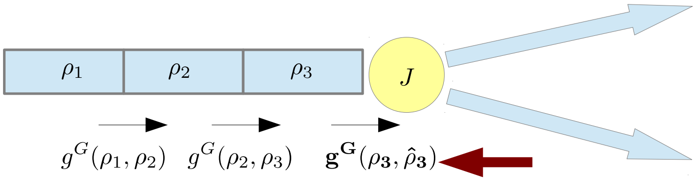
No longer depends on solution of continuous function.



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

Algorithm 3 Riemann solver update procedure

Input: initial state at time $t = k\Delta t$, $\left(\rho_i^k : i \in \mathcal{I}\right)$

Output: resulting state at time $t = (k+1))\,\Delta t$, $\left(\rho_i^{k+1} : i \in \mathcal{I}\right)$

for junction $J \in \mathcal{J}$:

    # Apply Riemann solver to J

    $\vec{\hat{\rho}}_J^k = RS\left(\vec{\rho}_J^k\right)$

for link $i \in \mathcal{I}$:

    # update density on link i with junction fluxes

    $\rho_i^{k+1} = \rho_i^k - \frac{\Delta t}{\Delta x}\left(f\left(\left(\vec{\hat{\rho}}_{J_i^{\mathrm{D}}}^k\right)_i\right) - f\left(\left(\vec{\hat{\rho}}_{J_i^{\mathrm{U}}}^k\right)_i\right)\right)$

Summary of Godunov scheme for PDE networks

Or by using the flux solution directly...

Algorithm 4 `Godunov junction flux update procedure`

`Input: initial state at time` $t = k\Delta t$, $\left(\rho_i^k : i \in \mathcal{I}\right)$

`Output: resulting state at time` $t = (k+1))\,\Delta t$, $\left(\rho_i^{k+1} : i \in \mathcal{I}\right)$

`for link` $i \in \mathcal{I}$`:`

`# update density on link` i `with direct Godonuv fluxes`

$$\rho_i^{k+1} = \rho_i^k - \frac{\Delta t}{\Delta x}\left(\left(g^G_{J_i^{\mathrm{D}}}\left(\vec{\rho}^{\,k}_{J_i^{\mathrm{D}}}\right)\right)_i - \left(g^G_{J_i^{\mathrm{U}}}\left(\vec{\rho}^{\,k}_{J_i^{\mathrm{U}}}\right)\right)_i\right)$$

Overview

PDE with control

Modify formulation to include

- state vector $\vec{\rho} \in \mathbb{R}^{NT}$
- control vector $\vec{u} \in \mathbb{R}^{MT}$
  - $\left(u^k_{j^1_J}, \ldots, u^k_{j^{M_J}_J}\right) \in \mathbb{R}^{M_J}$ modifies Riemann problem at J for time k.

$$\begin{aligned} RS_J : \quad & \mathbb{R}^{n_J+m_J} \times \mathbb{R}^{M_J} && \to \mathbb{R}^{n_J+m_J} \\ & \left(\vec{\rho}^k_J, \vec{u}^k_J\right) && \mapsto RS_J\left(\vec{\rho}^k_J, \vec{u}^k_J\right) = \vec{\hat{\rho}}^k_J \end{aligned}$$

  - M are the number of control parameters at each time-step.

Updated discrete state equations:

$$\begin{aligned} h^k_i(\vec{\rho}, \vec{u}) = \quad \rho^k_i - \rho^{k-1}_i \quad & + \frac{\Delta t}{\Delta x}\left(g^G_{J^D_i}\left(\vec{\rho}^k_{J^D_i}, \vec{u}^{k-1}_{J^D_i}\right)\right)_i \\ & - \frac{\Delta t}{\Delta x}\left(g^G_{J^U_i}\left(\vec{\rho}^k_{J^U_i}, \vec{u}^{k-1}_{J^U_i}\right)\right)_i \end{aligned} \quad (8)$$

Optimization problem

Optimization Problem

$$\begin{aligned} \min_{\vec{u}} \quad & C(\vec{\rho}, \vec{u}) \\ \text{subject to:} \quad & H(\vec{\rho}, \vec{u}) = 0 \end{aligned} \tag{9}$$

Review: adjoint method

$$d_{\vec{u}} C(\vec{\rho}', \vec{u}') = \lambda^T H_{\vec{u}} + C_{\vec{u}} \tag{10}$$

where

$$H_{\vec{\rho}}^T \lambda = -C_{\vec{\rho}}^T \tag{11}$$

Assume initial $\vec{u}$ and state $\vec{\rho}$ where $H(\vec{\rho}, \vec{u}) = 0$.

What needs to be computed for adjoint method?

- $C_{\vec{\rho}}$, $C_{\vec{u}}$: Problem specific, no sparsity assumptions.
- $H_{\vec{\rho}}$, $H_{\vec{u}}$: can analyze properties of PDE networks and Godunov scheme to:
  - derive partial derivative expressions
  - understand sparsity

Partial derivates of state equations

$H_{\vec{\rho}}$

By chain rule:

$$\frac{\partial h_i^k}{\partial \rho_j^l} = \frac{\partial \rho_i^k}{\partial \rho_j^l} - \frac{\partial \rho_i^{k-1}}{\partial \rho_j^l} +$$

$$\frac{\Delta t}{L_i}\left(\frac{\partial}{\partial \rho_j^l}\left(g^G_{J_i^{\mathbf{D}}}\left(\vec{\rho}^{k-1}_{J_i^{\mathbf{D}}}, \vec{u}^{k-1}_{J_i^{\mathbf{D}}}\right)\right)_i - \frac{\partial}{\partial \rho_j^l}\left(g^G_{J_i^{\mathbf{U}}}\left(\vec{\rho}^{k-1}_{J_i^{\mathbf{U}}}, \vec{u}^{k-1}_{J_i^{\mathbf{U}}}\right)\right)_i\right)$$

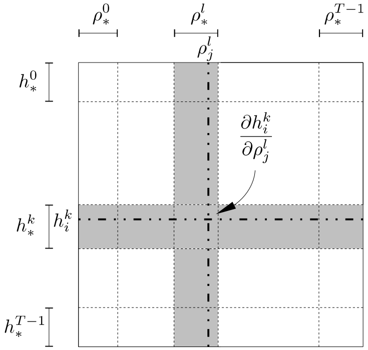
- Only require knowledge of partial derivatives on Godunov fluxes g^G.
- $\frac{\partial}{\partial \rho_j^l}\left(g^G_{J_i^{\mathbf{D}}}\left(\vec{\rho}^{k-1}_{J_i^{\mathbf{D}}}, \vec{u}^{k-1}_{J_i^{\mathbf{D}}}\right)\right)_i = 0$ unless $l = k-1$.
- $\frac{\partial}{\partial \rho_j^l}\left(g^G_{J_i^{\mathbf{D}}}\left(\vec{\rho}^{k-1}_{J_i^{\mathbf{D}}}, \vec{u}^{k-1}_{J_i^{\mathbf{D}}}\right)\right)_i = 0$ unless j is downstream of i.
- Similar results for $J_i^{\mathbf{U}}$.

Partial derivates of state equations

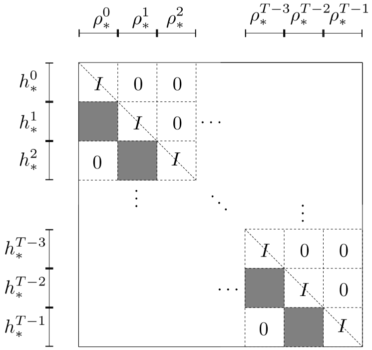
$H_{\vec{\rho}}$

- Only require knowledge of partial derivatives on Godunov fluxes g^G.
- $\frac{\partial}{\partial \rho_j^l} \left(g^G_{J_i^{\mathrm{D}}} \left(\vec{\rho}^{\,k-1}_{J_i^{\mathrm{D}}}, \vec{u}^{\,k-1}_{J_i^{\mathrm{D}}} \right) \right)_i = 0$ unless $l = k-1$.
- $\frac{\partial}{\partial \rho_j^l} \left(g^G_{J_i^{\mathrm{D}}} \left(\vec{\rho}^{\,k-1}_{J_i^{\mathrm{D}}}, \vec{u}^{\,k-1}_{J_i^{\mathrm{D}}} \right) \right)_i = 0$ unless j is downstream of i.
- Similar results for J_i^{U}.

Thus each partial term is zero unless variable is from previous time-step and adjacent to constraint link (or $i = j$ and $l = k$.



(b) Ordering of the partial derivative terms. Constraints and state variables are clustered first by time, and then by cell index.



(c) Sparsity structure of the $H_{\vec{\rho}}$ matrix. Besides the diagonal blocks, which are identity matrices, blocks where $l \neq k-1$ are zero.

Figure: Structure of the $H_{\vec{\rho}}$ matrix.

Partial derivates of state equations

$H_{\vec{u}}$

By chain rule:

$$\frac{\partial h_i^k}{\partial u_j^l} = \frac{\Delta t}{L_i}\left(\frac{\partial}{\partial u_j^l}\left(g^G_{J_i^{\mathbf{D}}}\left(\vec{\rho}^{\,k-1}_{J_i^{\mathbf{D}}}, \vec{u}^{\,k-1}_{J_i^{\mathbf{D}}}\right)\right)_i - \frac{\partial}{\partial u_j^l}\left(g^G_{J_i^{\mathbf{U}}}\left(\vec{\rho}^{\,k-1}_{J_i^{\mathbf{U}}}, \vec{u}^{\,k-1}_{J_i^{\mathbf{U}}}\right)\right)_i\right) \tag{12}$$

Similar arguments to $H_{\vec{\rho}}$ give us that each partial term above is zero unless control variable u_j^l is from same time-step and in

$$\left(u^k_{j^{\mathbf{1}}_{J_i^{\mathbf{U}}}}, \ldots, u^k_{j^{M_{J_i^{\mathbf{U}}}}_{J_i^{\mathbf{U}}}}\right) \text{ or } \left(u^k_{j^{\mathbf{1}}_{J_i^{\mathbf{D}}}}, \ldots, u^k_{j^{M_{J_i^{\mathbf{D}}}}_{J_i^{\mathbf{D}}}}\right).$$

Complexity of solving gradient

Solving adjoint system

$$H_{\vec{\rho}}^T \lambda = -C_{\vec{\rho}}^T \tag{13}$$

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

- size $TN \times TN$
- lower triangular
- *card* $H_{\vec{\rho}} = O(NTD_{\vec{\rho}})$: $D_{\vec{\rho}} = \max_{J \in \mathcal{J}} (n_J + m_J)$

Efficiently solve λ via backward-substitution in time $O(TND_{\vec{\rho}})$, or **linear in** TN.

Complexity of solving gradient

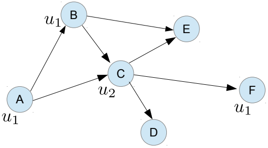
Solving ∇C

$$\nabla C = \lambda^T H_{\vec{u}} + C_{\vec{u}} \tag{14}$$

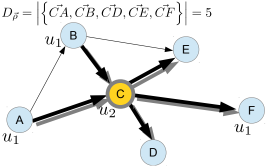
From previous result, $H_{\vec{u}}$ has following properties:

- size $TN \times TM$
- *card* $H_{\vec{u}} = O(TND_{\vec{u}})$: $D_{\vec{u}} = \max_{u \in \vec{u}} \sum_{J \in \mathcal{J} : u \in \vec{u}_J^*} (n_J + m_J)$

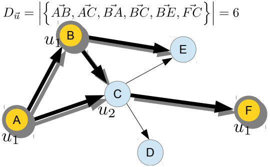
Sparse matrix multiplication has total cost $O(TMD_{\vec{u}})$.


A
B
C
D
E
F
u_1
u_2

(a)


$D_{\vec{\rho}} = \left|\left\{\vec{CA}, \vec{CB}, \vec{CD}, \vec{CE}, \vec{CF}\right\}\right| = 5$
A
B
C
D
E
F
u_1
u_2

(b)


$D_{\vec{u}} = \left|\left\{\vec{AB}, \vec{AC}, \vec{BA}, \vec{BC}, \vec{BE}, \vec{FC}\right\}\right| = 6$
A
B
C
D
E
F
u_1
u_2

(c)

Complexity of solving gradient

Total complexity of computing gradient via discrete adjoint

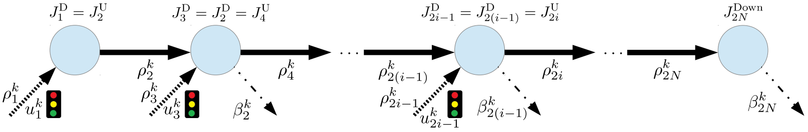
$$O(T\,(D_{\vec{\rho}}N + D_{\vec{u}}M))$$

$$\nabla C = \lambda^T H_{\vec{u}} + C_{\vec{u}} \qquad \sout{O(NT^2M)} \qquad O(TMD_{\vec{u}})$$

$$H_x^T \lambda = -J_x \qquad \sout{O((NT)^3)} \qquad O(NTD_{\vec{\rho}}) \quad \sout{O((NT)^2)}$$

Overview

Discrete adjoint method applied to ramp metering



- For a junction $J_{2i-1}^{\mathrm{D}} = J_{2(i-1)}^{\mathrm{D}} = J_{2i}^{\mathrm{U}}$ at time-step $k \in \{0, \ldots, T-1\}$.
- Upstream mainline density: $\rho_{2(i-1)}^k$.
- Downstream mainline density: ρ_{2i}^k.
- Onramp density: ρ_{2i-1}^k
- Offramp split ratio: $\beta_{2(i-1)}^k$.

Governing Equations

Mainline equations

$$h_{2i}^{k}(\vec{\rho}, \vec{u}) = \quad \rho_{2i}^{k} - \rho_{2i}^{k-1} \quad + \frac{\Delta t}{L_{2i}} \left(g_{2i,\mathrm{D}}^{k-1} - g_{2i,\mathrm{U}}^{k-1} \right) = 0$$

Onramp equations

$$h_{2i-1}^{k}(\vec{\rho}, \vec{u}) \quad = \quad \rho_{2i-1}^{k} - \rho_{2i-1}^{k-1} + \frac{\Delta t}{L_{2i-1}} \left(g_{2i-1,\mathrm{D}}^{k-1} - D_{2i-1}^{k-1} \right) = 0$$

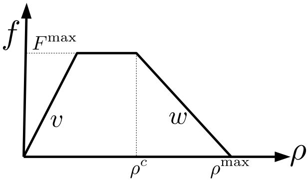
Flux solutions

$$\delta^k_{2(i-1)} = \min\left(v_{2(i-1)}\rho^k_{2(i-1)}, F^{\max}_{2(i-1)}\right) \quad (15)$$

$$\sigma^k_{2i} = \min\left(w_{2i}\left(\rho^{\max}_{2i} - \rho^k_{2i}\right), F^{\max}_{2i}\right) \quad (16)$$

$$d^k_{2i-1} = u^k_{2i-1}\min\left(\frac{L_{2i-1}}{\Delta t}\rho^k_{2i-1}, F^{\max}_{2i-1}\right) \quad (17)$$

$$g^k_{2i,\mathrm{U}} = \min\left(\beta^k_{2(i-1)}\delta^k_{2(i-1)} + d^k_{2i-1}, \sigma^k_{2i}\right) \quad (18)$$



Flux solutions

$$\delta^k_{2(i-1)} = \min\left(v_{2(i-1)}\rho^k_{2(i-1)}, F^{\max}_{2(i-1)}\right) \quad (19)$$

$$\sigma^k_{2i} = \min\left(w_{2i}\left(\rho^{\max}_{2i} - \rho^k_{2i}\right), F^{\max}_{2i}\right) \quad (20)$$

$$d^k_{2i-1} = u^k_{2i-1}\min\left(\frac{L_{2i-1}}{\Delta t}\rho^k_{2i-1}, F^{\max}_{2i-1}\right) \quad (21)$$

$$g^k_{2i,\mathrm{U}} = \min\left(\beta^k_{2(i-1)}\delta^k_{2(i-1)} + d^k_{2i-1}, \sigma^k_{2i}\right) \quad (22)$$

$$g^k_{2(i-1),\mathrm{D}} = \begin{cases} \delta^k_{2(i-1)} & \frac{p_{\mathbf{2}(i-\mathbf{1})}g^k_{\mathbf{2}i,\mathbf{U}}}{\beta^k_{\mathbf{2}(i-\mathbf{1})}\left(1+p_{\mathbf{2}(i-\mathbf{1})}\right)} \geq \delta^k_{2(i-1)} \text{[C1]} \\ \frac{g^k_{\mathbf{2}i,\mathbf{U}} - d^k_{\mathbf{2}i-\mathbf{1}}}{\beta^k_{\mathbf{2}(i-\mathbf{1})}} & \frac{g^k_{\mathbf{2}i,\mathbf{U}}}{1+p_{\mathbf{2}(i-\mathbf{1})}} \geq d^k_{2i-1} \quad \text{[C2]} \\ \frac{p_{\mathbf{2}(i-\mathbf{1})}g^k_{\mathbf{2}i,\mathbf{U}}}{\left(1+p_{\mathbf{2}(i-\mathbf{1})}\right)\beta^k_{\mathbf{2}(i-\mathbf{1})}} & \text{otherwise} \quad \text{[C3]} \end{cases} \quad (23)$$

$$g^k_{2i-1,\mathrm{D}} = g^k_{2i,\mathrm{U}} - \beta^k_{2(i-1)}g^k_{2(i-1),\mathrm{D}} \quad (24)$$

Partial derivates for adjoint method

$$\frac{\partial \delta^k_{2(i-1)}}{\partial s} = \begin{cases} v_{2(i-1)} & s = \rho^k_{2(i-1)},\ v_i \rho^k_{2(i-1)} \le F^{\max}_{2(i-1)} \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial \sigma^k_{2i}}{\partial s} = \begin{cases} -w_{2i} & s = \rho^k_{2i},\ w_{2i}\left(\rho^{\max}_{2i} - \rho^k_{2i}\right) \le F^{\max}_{2i} \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial d}{\partial s} = \begin{cases} u^k_{2i-1} & s = \rho^k_{2i-1},\ \rho^k_{2i-1} \le F^{\max}_{2i-1} \\ \min\left(\rho^k_{2i-1}, F^{\max}_{2i-1}\right) & s = u^k_{2i-1} \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial s} g^k_{2i,U} = \begin{cases} \beta^k_{2(i-1)} \frac{\partial \delta^k_{2(i-1)}}{\partial s} + \frac{\partial d^k_{2(i-1)}}{\partial s} & \beta^k_{2(i-1)} \delta^k_{2(i-1)} + d^k_{2i-1} \le \sigma^k_{2i} \\ \frac{\partial \sigma^k_{2i}}{\partial s} & \text{otherwise} \end{cases}$$

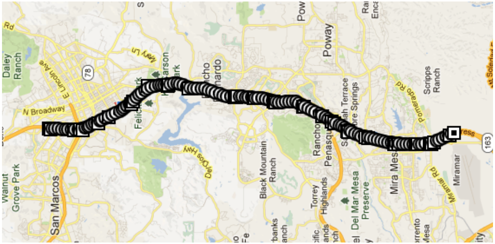
$$\frac{\partial}{\partial s} g_{2(i-1),D} = \begin{cases} \frac{\partial \delta^k_{2(i-1)}}{\partial s} & \frac{g^k_{2i,U} p_{2(i-1)}}{1+p_{2(i-1)}} \ge \frac{\delta^k_{2(i-1)}}{\beta^k_{2(i-1)}} \\ \frac{1}{\beta^k_{2(i-1)}} \left(\frac{\partial}{\partial s} g^k_{2i,U} - \frac{\partial d^k_{2i-1}}{\partial s} \right) & \frac{g^k_{2i,U}}{1+p_{2(i-1)}} \ge d^k_{2(i-1)} \\ \frac{p_{2(i-1)}}{\beta^k_{2(i-1)}\left(1+p_{2(i-1)}\right)} \frac{\partial}{\partial s} g^k_{2i,U} & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial s} g_{2i-1,D} = \frac{\partial}{\partial s} g^k_{2i,U} - \beta^k_{2(i-1)} \frac{\partial}{\partial s} g_{2(i-1),D}$$

Numerical results

Network

- I15 South in San Diego, CA, USA.
- 19.4 miles.
- 125 discrete links.
- 9 onramps.
- Scaled flow data from real loop detector data.

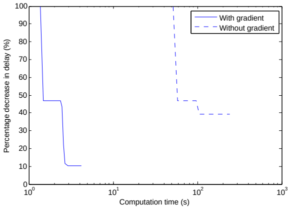


Numerical results

Comparative methods

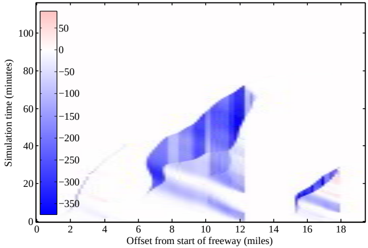
- Adjoint method
- *Finite differences (No gradient information).
- Alinea [1]
- No control

*Finite differences becomes impractical for even very small networks.

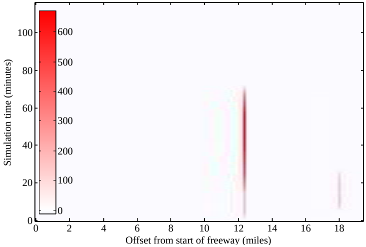


[1] Papageorgiou et al

Open loop optimal control



(d) Density difference profile in units of vehicles per mile.



(e) Queue difference profile in units of vehicles.

Model predictive control

- Assume noisy state estimation and noisy predicted boundary flow data.
- About 2% relative error in estimates.
- Every 15 minutes:
  - Get state and boundary flow estimates over next 25 minutes.
  - Produce policy for next 15 minutes.
- Repeat for entire simulation horizon.

Model predictive control

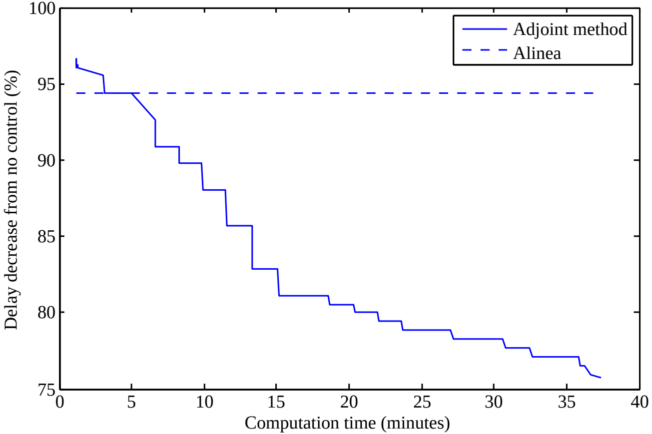
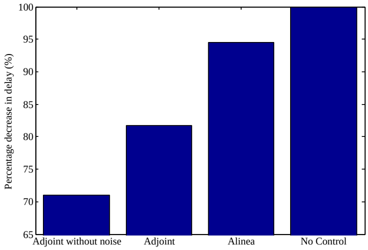


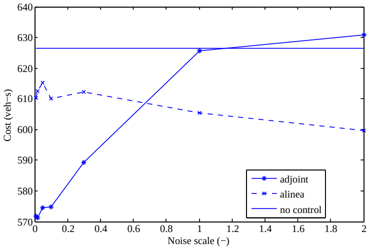
Figure: Performance versus simulation time for freeway network. The results indicate that the algorithm can run with performance better than Alinea if given an update time around 15 minutes.

Robustness to noise

- Adjoint method relies on reasonable input data estimates.
- If data is too noisy (>100% relative error), reactive methods such as Alinea will outperform.



(a) MPC performance on I15 South network.



(b) MPC performance with increasing sensor noise.

Partial control via rerouting

...