Optimization-Based Modeling: Part 3 – Monotone Methods for Transport without Limiters

SAND2011-3864C

Joseph Young Denis Ridzal Pavel Bochev

Sandia National Laboratories

8th International Conference on Large-Scale Scientific Computations, 06-07-2011

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Outline

Review of the Remap Subproblem

Dual Reformulation

Optimization Algorithm

Application to Transport

Remap Subproblem

From our second talk, optimization based remap requires the solution to the remap subproblem

$$\min_{\widetilde{F}_{ij}^h} \sum_{i=1}^{K} \sum_{\widetilde{\kappa}_j \in \widetilde{N}_i} (\widetilde{F}_{ij}^h - \widetilde{F}_{ij}^T)^2 \quad \text{subject to}$$

$$\begin{cases} \widetilde{F}_{ij}^h = -\widetilde{F}_{ji}^h \\ m_i^{\min} \le \widetilde{m}_i + \sum_{\widetilde{\kappa}_j \in \widetilde{N}_i} \widetilde{F}_{ij}^h \le m_i^{\max} \end{cases} .$$

When we enforce the antisymmetric constraint $\tilde{F}_{ij}^h = -\tilde{F}_{ji}^h$ explicitely, we can reformulate the problem into

$$\begin{aligned} &\min_{\vec{F} \in \mathbb{R}^M} \; \frac{1}{2}(\vec{F} - \vec{F}^H)^{\mathsf{T}}(\vec{F} - \vec{F}^H) \quad \text{subject to} \\ &\vec{b}_{\min} \le \mathbf{A}\vec{F} \le \vec{b}_{\max} \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{K \times M}$ denotes a sparse matrix containing only -1, 0, and 1 entries.

Dual Reformulation

Rather than solve

$$\min_{\vec{F}\in\mathbb{R}^M} \frac{1}{2}(\vec{F}-\vec{F}^H)^{\mathsf{T}}(\vec{F}-\vec{F}^H) \quad \text{subject to}$$
$$\vec{b}_{\min} \leq \mathbf{A}\vec{F} \leq \vec{b}_{\max}\,, \tag{2}$$

directly, we solve it's dual formulation,

$$\min_{(\vec{\lambda},\vec{\mu})\in\mathbb{R}_+^{2K}} \frac{1}{2}\|\mathbf{A}^T\vec{\lambda}-\mathbf{A}^T\vec{\mu}\|_2^2 - \langle\vec{\lambda},\vec{b}_{\min}-\mathbf{A}\vec{F}^H\rangle - \langle\vec{\mu},-\vec{b}_{\max}+\mathbf{A}\vec{F}^H\rangle \quad .$$

This allows us to reformulate the remap subproblem into a simpler, bound-constrained optimization problem.

Dual Reformulation

Theorem (Y., R., B. 2011)
Let us define $J_p : \mathbb{R}^M \to \mathbb{R}$ *and* $J_d : \mathbb{R}^{2K} \to \mathbb{R}$ *as*

$$J_p(\vec{F}) = \frac{1}{2}\|\vec{F} - \vec{F}^H\|_2^2$$

$$J_d(\vec{\lambda}, \vec{\mu}) = \frac{1}{2}\|\mathbf{A}^\mathsf{T}\vec{\lambda} - \mathbf{A}^\mathsf{T}\vec{\mu}\|_2^2 - \langle \vec{\lambda}, \vec{b}_{\min} - \mathbf{A}\vec{F}^H \rangle - \langle \vec{\mu}, -\vec{b}_{\max} + \mathbf{A}\vec{F}^H \rangle.$$

Then, we have that

$$\min_{F \in \mathbb{R}^M} \left\{ J_p(\vec{F}) : \vec{b}_{\min} \le \mathbf{A}\vec{F} \le \vec{b}_{\max} \right\} = \min_{(\vec{\lambda},\vec{\mu}) \in \mathbb{R}_+^{2K}} \left\{ J_d(\vec{\lambda}, \vec{\mu}) \right\}$$

Furthermore,

$$\{\vec{F}^H + \mathbf{A}^\mathsf{T}(\vec{\lambda}^* - \vec{\mu}^*)\} = \arg\min_{F \in \mathbb{R}^M} \left\{ J_p(\vec{F}) : \vec{b}_{\min} \le \mathbf{A}\vec{F} \le \vec{b}_{\max} \right\}$$

whenever

$$(\vec{\lambda}^*, \vec{\mu}^*) \in \arg\min_{(\vec{\lambda},\vec{\mu}) \in \mathbb{R}_+^{2K}} \left\{ J_d(\vec{\lambda}, \vec{\mu}) \right\}.$$

Comments about the Dual Reformulation

- The point $(\vec{0}, \vec{0})$ is both dual feasible and near the optimal solution.
- Since $(\vec{0}, \vec{0})$ lies close to the optimal solution, Newton's method should converge quadratically (assuming we address the inequality constraints).
- Unfortunately, addressing the inequality constraints is difficult
  - The active set changes between iterations.
  - The initial guess lies on the edge of the feasible set.
  - As a result, neither active-set methods or barrier based interior point methods are appropriate for this problem.
- In order to address these difficulties, we apply a simplified version of the Coleman-Li reflective Newton algorithm.

Reformulation of the Optimality Conditions

Definition

We define the diagonal operator, $\text{Diag} : \mathbb{R}^m \to \mathbb{R}^{m\times m}$, as

$$[\text{Diag}(\vec{x})]_{ij} = \begin{cases} \vec{x}_i & \text{when} \quad i = j \\ 0 & \text{"} \quad i \neq j \end{cases} .$$

Definition

For some symmetric, positive semidefinite $\mathbf{H} \in \mathbb{R}^{m\times m}$ and some $\vec{b} \in \mathbb{R}^m$, we define the operator $v_{\mathbf{H},\vec{b}} : \mathbb{R}^m \to \mathbb{R}^m$ as

$$v_{\mathbf{H},\vec{b}}(\vec{x}) = \begin{cases} \vec{x}_i & \text{when} \quad [\mathbf{H}\vec{x} + \vec{b}]_i \geq 0 \\ 1 & \text{"} \quad [\mathbf{H}\vec{x} + \vec{b}]_i < 0 \end{cases} .$$

When both $\mathbf{H}$ and $\vec{b}$ are clear from the context, we abbreviate this function as v.

Reformulation of the Optimality Conditions

Lemma

Let $\mathbf{H} \in \mathbb{R}^{m\times m}$ *be symmetric, positive semidefinite and let* $\vec{b} \in \mathbb{R}^m$*.*
Then, for some $\vec{x}^* \geq 0$*, we have that*

$$\vec{x}^* \in \arg\min_{x\in\mathbb{R}^m_+} \left\{ \frac{1}{2}\langle \mathbf{H}\vec{x}, \vec{x}\rangle + \langle \vec{b}, \vec{x}\rangle \right\} \Longleftrightarrow \mathrm{Diag}(v(\vec{x}^*))(\mathbf{H}\vec{x}^* + \vec{b}) = 0.$$

This lemma allows us to reformulate a bound-constrained, quadratic program as a **piecewise-differentiable system of equations**.

Newton's Method Applied to the Reformulation

In order to apply Newton's method, we require the derivative of the reformulated function

Definition

For some symmetric, positive semidefinite $\mathbf{H} \in \mathbb{R}^{m \times m}$ and some $\vec{b} \in \mathbb{R}^m$, we define the operator $K_{\mathbf{H},\vec{b}} : \mathbb{R}^m \to \mathbb{R}^{m \times m}$ as

$$[K_{\mathbf{H},\vec{b}}(\vec{x})]_{ii} = \begin{cases} 1 & \text{when} \quad [\mathbf{H}\vec{x} + \vec{b}]_i \geq 0 \\ 0 & \text{"} \quad [\mathbf{H}\vec{x} + \vec{b}]_i < 0 \end{cases} .$$

When $\mathbf{H}$ and $\vec{b}$ are clear from context, we abbreviate this as K.

Lemma (Y., R., B. 2011)

Let $\mathbf{H} \in \mathbb{R}^{m \times m}$ be symmetric, positive definite, $\vec{b} \in \mathbb{R}^m$, and define J as

$$J(\vec{x}) = \text{Diag}(v(\vec{x}))(\mathbf{H}\vec{x} + \vec{b}).$$

Then, we have that

$$J'(\vec{x}) = K(\vec{x})\text{Diag}(\mathbf{H}\vec{x} + \vec{b}) + \text{Diag}(v(x))H.$$

Newton's Method Applied to the Reformulation

We may now apply Newton's method to the nonlinear system by solving

$$(K(\vec{x})\text{Diag}(\mathbf{H}\vec{x}+\vec{b})+\text{Diag}(v(\vec{x}))\mathbf{H})\vec{p}=-\text{Diag}(v(\vec{x}))(\mathbf{H}\vec{x}+\vec{b})$$

for p. In our case, we accomplish this by a **sparse-LU factorization**. Nonetheless, in some cases we can be slightly more efficient.

Reformulation of the Newton System

Definition

For some symmetric, positive semidefinite $\mathbf{H} \in \mathbb{R}^{m \times m}$ and some $\vec{b} \in \mathbb{R}^m$, we define the operator $D_{\mathbf{H},\vec{b}} : \mathbb{R}^m_+ \rightarrow \mathbb{R}^{m \times m}$ as

$$D_{\mathbf{H},\vec{b}}(\vec{x}) = \text{Diag}(v_{\mathbf{H},\vec{b}}(\vec{x}))^{1/2}.$$

When both $\mathbf{H}$ and $\vec{b}$ are clear from the context, we abbreviate this operator as D.

Lemma (Y., R., B. 2011)

Let $\mathbf{H} \in \mathbb{R}^{m \times m}$ be symmetric, positive semidefinite and let $\vec{b} \in \mathbb{R}^m$. Then, we have that

$$(K(\vec{x})\text{Diag}(\mathbf{H}\vec{x} + \vec{b}) + \text{Diag}(v(\vec{x}))\mathbf{H})\vec{p} = -\text{Diag}(v(\vec{x}))(\mathbf{H}\vec{x} + \vec{b})$$
$$\iff (K(\vec{x})\text{Diag}(\mathbf{H}\vec{x} + \vec{b}) + D(\vec{x})\mathbf{H}D(\vec{x}))\vec{q} = -D(\vec{x})(\mathbf{H}\vec{x} + \vec{b})$$

where $\vec{p} = D(x)\vec{q}$.

We can solve the second system with a **sparse-Choleski factorization**.

Choleski v. LU

- As long as we remain strictly feasible, we can reformulate the problem into a symmetric, positive definite system solvable by Choleski.
- However, in order to remain strictly feasible, we require a reflective linesearch detailed by Coleman and Li.
- Since we start so close to optimality, we do not enforce strict feasibility. Rather, when we become infeasible, we solve the nonlinear system with a LU factorization.

Dual Algorithm for the Remap Subproblem

1. Define $H \in \mathbb{R}^{2K \times 2K}$ and $b \in \mathbb{R}^{2K}$ as

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}\mathbf{A}^T & -\mathbf{A}\mathbf{A}^T \\ -\mathbf{A}\mathbf{A}^T & \mathbf{A}\mathbf{A}^T \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} \mathbf{A}\vec{F}^H - \vec{b}_{\text{min}} \\ -\mathbf{A}\vec{F}^H + \vec{b}_{\text{max}} \end{bmatrix}.$$

2. Initialize $\vec{x} = \vec{0}$.
3. Until $\|\text{Diag}(v(\vec{x}))(\mathbf{H}\vec{x} + \vec{b})\|$ becomes small or we exceed a fixed number of iterations.
    1. When feasible, solve

    $$(K(\vec{x})\text{Diag}(\mathbf{H}\vec{x} + \vec{b}) + D(\vec{x})\mathbf{H}D(\vec{x}))\vec{q} = -D(\vec{x})(\mathbf{H}\vec{x} + \vec{b})$$

    and set $\vec{p} = D(x)\vec{q}$. Otherwise, solve

    $$(K(\vec{x})\text{Diag}(\mathbf{H}\vec{x} + \vec{b}) + \text{Diag}(v(\vec{x}))\mathbf{H})\vec{p} = -\text{Diag}(v(\vec{x}))(\mathbf{H}\vec{x} + \vec{b}).$$

    2. Set $\vec{x} = \vec{x} + \vec{p}$.

Transport Formulation

Consider the transport equations

$$\partial_t \rho + \nabla \cdot \rho \mathbf{v} = 0 \text{ on } \Omega \times [0, T] \quad \text{and} \quad \rho(\mathbf{x}, 0) = \rho^0(\mathbf{x}).$$

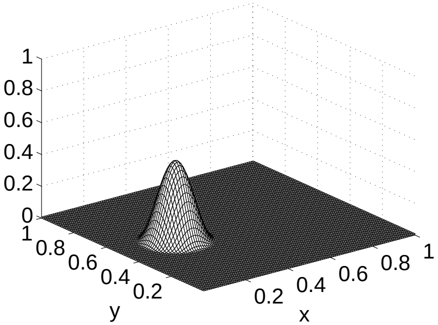
In order to compute one forward step of incremental remapping, we apply the following subalgorithm

Input Density approximation $\vec{\rho}^{\,n} = (\rho_1^n, \ldots, \rho_K^n)$ at time t_n, time step Δt_n
Output Density approximation $\vec{\rho}^{\,n+1} = (\rho_1^{n+1}, \ldots, \rho_K^{n+1})$ at time t_{n+1}

1. Project grid: $K_h(\Omega) \ni \mathbf{x}_p \mapsto \mathbf{x}_p + \Delta t_n \mathbf{v} = \widetilde{\mathbf{x}}_p \in \widetilde{K}_h(\widetilde{\Omega})$
2. Transport m and ρ: $\forall \widetilde{\kappa}_i \in \widetilde{K}_h(\widetilde{\Omega})$ set $\widetilde{m}_i = m_i^n$ and $\widetilde{\rho}_i = \widetilde{m}_i / \text{vol}(\widetilde{\kappa}_i)$
3. Remap density: $\vec{\rho}^{\,n+1} = \mathcal{R}(\{\widetilde{\rho}_1, \ldots, \widetilde{\rho}_K\})$

Rotating Hump

In our first example, we rotate a smooth hump in a circular flow.



Rotating Hump: Accuracy

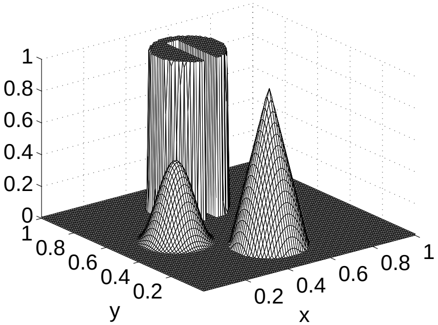
LVLT					
#cells	#remaps	L_2 err	L_1 err	L_2 rate	L_1 rate
100×100	5026	4.00e-03	8.88e-04	1.70	1.67
120×120	6031	2.94e-03	6.59e-04	1.69	1.65
140×140	7037	2.35e-03	5.30e-04	1.64	1.60

FCRT					
#cells	#remaps	L_2 err	L_1 err	L_2 rate	L_1 rate
100×100	5026	3.89e-03	8.63e-04	1.68	1.62
120×120	6031	2.85e-03	6.45e-04	1.69	1.61
140×140	7037	2.29e-03	5.21e-04	1.63	1.56

OBT					
#cells	#remaps	L_2 err	L_1 err	L_2 rate	L_1 rate
100×100	5026	4.11e-03	9.38e-04	1.81	1.81
120×120	6031	2.95e-03	6.83e-04	1.82	1.78
140×140	7037	2.33e-03	5.46e-04	1.75	1.70

Rotating Combo

In our second example, we rotate a combination of the rotating hump above, a cone, and a slotted disk using the same velocity field

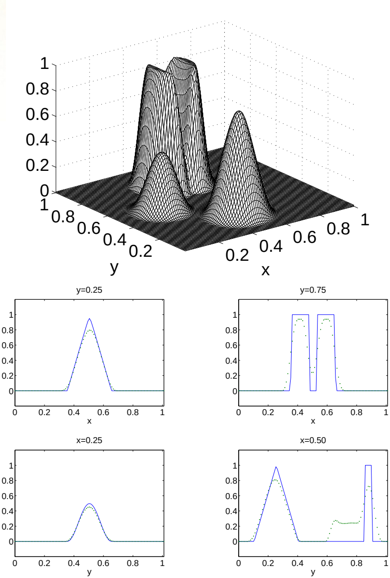


Rotating Combo: Computational Cost

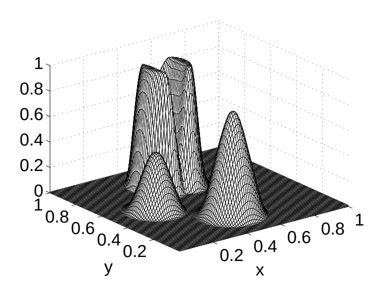
Grid Size	40×40	80×80	160×160	320×320
OBT	4.00	34.21	422.85	4108.27
FCRT	0.83	5.48	45.27	375.90
LVLT	0.89	5.84	45.38	362.65

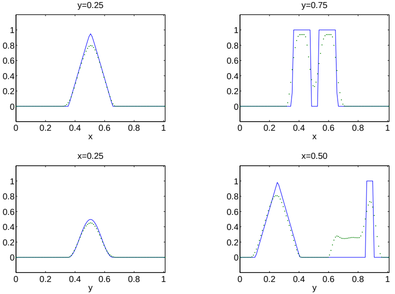
The cost of OBT is about 10 times higher than FCRT or LVLT.

Rotating Combo: Qualitative Results for OBT

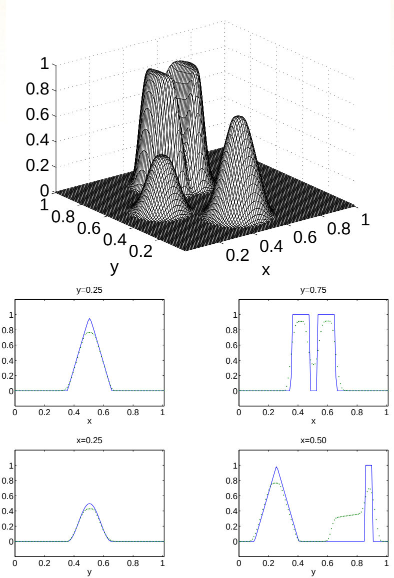


Rotating Combo: Qualitative Results for FCRT



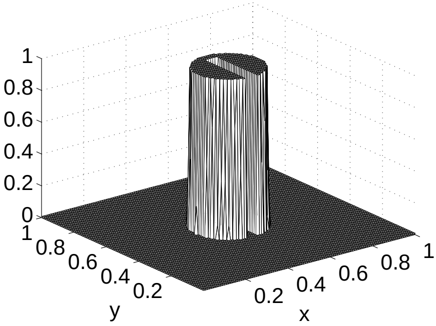


Rotating Combo: Qualitative Results for LVLT



Rotating Slotted Disk

In our third example, we rotate the slotted disk about it's axis one full revolution.



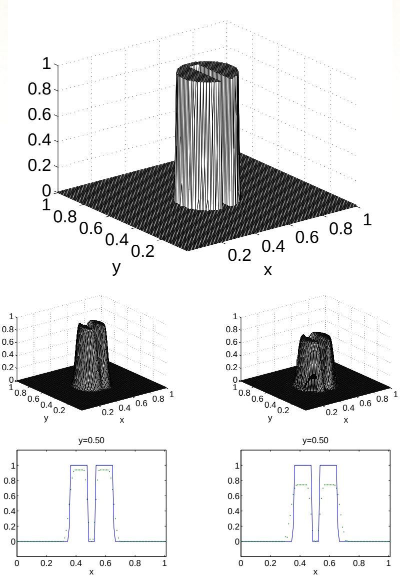
Rotating Slotted Disk: Robustness

	$1/\Delta t$=100 CFL=1.00	$1/\Delta t$=62 CFL=1.60	$1/\Delta t$=61 CFL=1.62	$1/\Delta t$=45 CFL=2.20
OBT	**2.14e-02**	**2.37e-02**	**2.38e-02**	**2.60e-02**
FCRT	**1.97e-02**	**2.19e-02**	**2.21e-02**	**3.00e-02**
LVLT	**2.14e-02**	**2.36e-02**	8.15e-01	3.47e+54

	$1/\Delta t$=44 CFL=2.25	$1/\Delta t$=19 CFL=5.50	$1/\Delta t$=18 CFL=5.21
OBT	**2.62e-02**	**4.02e-02**	4.36e-02
FCRT	6.00e+06	9.45e+38	1.83e+40
LVLT	2.85e+56	2.83e+79	6.23e+77

OBT is far more robust than the other methods.

Rotating Slotted Disk: Qualitative Results for OBT



Summary and Conclusions

- Dual reformulation of OBR provides an effective way to recast OBR as a bound constrained optimization problem.
- Reflective-Newton method gives an efficient algorithm to solve the dual reformulation.
- OBR works well on transport problems.
    - Slower than FCR or van Leer.
    - Far more robust than either method.