COMMUN. MATH. SCI.
Vol. 13, No. 1, pp. 15–48


# NUMERICAL METHOD FOR OPTIMAL CONTROL PROBLEMS GOVERNED BY NONLINEAR HYPERBOLIC SYSTEMS OF PDES*

MICHAEL HERTY†, ALEXANDER KURGANOV‡, AND DMITRY KUROCHKIN§

**Abstract.** We develop a numerical method for the solution to linear adjoint equations arising, for example, in optimization problems governed by hyperbolic systems of nonlinear conservation and balance laws in one space dimension. Formally, the solution requires one to numerically solve the hyperbolic system forward in time and a corresponding linear adjoint system backward in time. Numerical results for the control problem constrained by either the Euler equations of gas dynamics or isothermal gas dynamics equations are presented. Both smooth and discontinuous prescribed terminal states are considered.

**Key words.** PDE-constrained optimization problems, hyperbolic systems of conservation and balance laws, linear adjoint system, least-square cost functional, central-upwind finite-volume scheme, upwind finite-difference scheme.

**AMS subject classifications.** 49M29, 49M05, 49M10, 49K20, 49N50, 65M06, 65M08, 65M20.

## 1. Introduction

In this paper, we develop novel numerical optimization methods for linear adjoint equations arising formally from optimal control problems governed by hyperbolic systems of nonlinear PDEs. In particular, we focus on equations arising in optimization problems governed by one-dimensional (1-D) hyperbolic systems of conservation and balance laws. Such problems arise in a variety of applications such as gas dynamics, fuel cell control, optimal treatment in cancer therapy, flow control design, traffic flow, and many other areas, in which inverse problems for the corresponding initial value problems (IVP) are to be solved. The mathematical formulation of the minimization problem we are concerned with is stated as follows:

$$\min_{\boldsymbol{w}_0} J(\boldsymbol{w}(\cdot,T);\boldsymbol{w}_d(\cdot)), \tag{1.1}$$

where $J$ is a given functional and $\boldsymbol{w}(x,t)$ is the unique entropy solution of the following IVP for the hyperbolic systems of balance laws:

$$\begin{aligned} &\frac{\partial \boldsymbol{w}(x,t)}{\partial t}+\frac{\partial \boldsymbol{f}(\boldsymbol{w}(x,t))}{\partial x}=\boldsymbol{h}(\boldsymbol{w}(x,t),x,t), \qquad x\in\mathbb{R}, \quad t\in(0,T],\\ &\boldsymbol{w}(x,0)=\boldsymbol{w}_0(x), \qquad\qquad x\in\mathbb{R}. \end{aligned} \tag{1.2}$$

Here, $\boldsymbol{w}\colon \mathbb{R}\times[0,T]\to\mathbb{R}^s$, $\boldsymbol{w}_0(x)$ is an arbitrary bounded measurable function on $\mathbb{R}$, the corresponding nonlinear flux is denoted by $\boldsymbol{f}(\boldsymbol{w})$, $\boldsymbol{h}(\boldsymbol{w},x,t)$ is a source term, and the terminal state $\boldsymbol{w}_d(x)$ is prescribed at time $t=T$.

We consider among others the integral least-square cost functional of the form

$$J(\boldsymbol{w}(\cdot,T);\boldsymbol{w}_d(\cdot)):=\frac{1}{2}\int\limits_{\mathbb{R}}|\boldsymbol{w}(x,T)-\boldsymbol{w}_d(x)|^2\,dx. \tag{1.3}$$

*Received: August 26, 2013; accepted (in revised form): November 16, 2013. Communicated by Alberto Bressan.

†RWTH Aachen University, Department of Mathematics, Templergraben 55, D-52056 Aachen, Germany (herty@mathc.rwth-aachen.de).

‡Mathematics Department, Tulane University, New Orleans, LA 70118, USA (kurganov@math.tulane.edu).

§Mathematics Department, Tulane University, New Orleans, LA 70118, USA (dkurochk@tulane.edu).

In recent years, PDE-constrained optimization problems governed by scalar conservation and balance laws have been intensively studied both analytically [5, 6, 32, 34, 8] and numerically [1, 9, 14, 17, 18, 32, 34, 8]. However, only a few of the analytical results have been extended to the system case [6]. The analytical results and the rigorous treatment of the optimal control problem is subject to the fact that the semigroup generated by a nonlinear hyperbolic conservation/balance law is generically nondifferentiable in $L^1$ even in the 1-D scalar case. We refer to [1, 2, 3, 4, 5, 8, 17, 18, 13, 14, 15, 16, 33] for more details on the differential structure of solutions. In the 1-D scalar case, convergence results for first-order numerical schemes including shock variations have been established in [1, 9, 12, 14, 34], where a variety of numerical methods for the optimal control problems governed by *scalar* hyperbolic equations have been discussed. In particular, in the recent work on scalar equations [17, 18], the linear adjoint equation has been discretized using a Lax-Friedrichs-type scheme, obtained by including conditions along shocks and modifying the Lax-Friedrichs numerical viscosity. Convergence of the modified Lax-Friedrichs scheme has been rigorously proved in the case of a smooth convex flux function. Convergence results on the linear adjoint equation in the scalar case have also been obtained in [32] for the class of schemes satisfying the one-sided Lipschitz condition (OSLC) and in [1] for a first-order implicit-explicit finite-volume method. To the best of our knowledge, no convergence theory has been established for numerical methods for control problems governed by hyperbolic systems of conservation/balance laws.

In this work, we numerically study the linear adjoint equations arising in the formal optimality conditions to the problem (1.1), (1.2) and focus on designing numerical methods in the case of systems. In order to develop a numerical scheme and test the suggested procedure we only formally (in the case of smooth solutions) compute the optimality system and numerically study the nonsmooth case. The main source of difficulty for nonlinear hyperbolic system (1.2) comes from the loss of smoothness of its solution, which may develop discontinuities even for infinitely smooth initial conditions. In order to accurately capture the discontinuous parts of the solution, one can use high-resolution shock capturing finite-volume methods. Our particular choice is the second-order semi-discrete central-upwind scheme, which was introduced in [23, 24, 25] as a reliable "black-box" solver for general hyperbolic systems of conservation and balance laws. The linear adjoint system, however, is a nonconservative linear system of first-order PDEs with variable, generically discontinuous coefficients. Therefore, it has to be treated differently. To the best of our knowledge, no high-resolution numerical methods for liner hyperbolic systems with discontinuous coefficients are available. We develop a second-order Roe-type scheme for the adjoint equation. A detailed description of the proposed scheme is presented in Section 3.2.

A convergence analysis of the developed numerical method for the optimization problem governed by nonlinear hyperbolic systems of PDEs as well as a numerical treatment of the proposed conditions along a-priori known shock locations (see [6]) seems to be out of reach at this stage. The discussion of the convergence of the proposed scheme applied to a scalar equation will be studied in a forthcoming paper. Here, we focus on the development of a new numerical approach.

This paper is organized as follows. First, in Section 2, we introduce a Lagrange functional for the optimization problem (1.1)–(1.3) and present the optimality system for this problem. Then, in Section 3, we describe the designed optimization method. In Section 3.1, we briefly review the central-upwind scheme used to solve the forward equation. In Section 3.2, the proposed scheme for the adjoint system is presented in

detail. Then, in Section 3.3, we describe a conservative nonlinear filter, which may be used in the proposed optimization algorithm. Finally, Section 4 is dedicated to a variety of numerical experiments.

## 2. Formal derivation of the linear adjoint equations

In this section, we formally derive the linear adjoint equations assuming (for now) the solutions of (1.2) are sufficiently smooth. To this end, we consider the Lagrangian corresponding to the constrained optimization problem (1.1)–(1.3):

$$L(\boldsymbol{w}(\cdot),\boldsymbol{q}(\cdot))=\frac{1}{2}\int\limits_{\mathbb{R}}|\boldsymbol{w}(x,T)-\boldsymbol{w}_d(x)|^2\,dx - \int\limits_0^T\int\limits_{\mathbb{R}}\boldsymbol{q}(x,t)\left(\frac{\partial\boldsymbol{w}(x,t)}{\partial t}+\frac{\partial\boldsymbol{f}(\boldsymbol{w}(x,t))}{\partial x}-\boldsymbol{h}(\boldsymbol{w}(x,t),x,t)\right)dx\,dt. \tag{2.1}$$

Integrating by parts and computing variations of the Lagrange functional (2.1) with respect to $\boldsymbol{w}$ leads to the following adjoint system of linear equations:

$$-\frac{\partial\boldsymbol{q}(x,t)}{\partial t}-\frac{\partial\boldsymbol{f}(\boldsymbol{w})}{\partial\boldsymbol{w}}\frac{\partial\boldsymbol{q}(x,t)}{\partial x}=\frac{\partial\boldsymbol{h}(\boldsymbol{w},x,t)}{\partial\boldsymbol{w}}\boldsymbol{q}(x,t),\quad x\in\mathbb{R},\quad t\in[0,T), \tag{2.2}$$

and the terminal condition

$$\boldsymbol{q}(x,T)=\boldsymbol{w}(x,T)-\boldsymbol{w}_d(x),\quad x\in\mathbb{R}, \tag{2.3}$$

where $\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{w}}$ and $\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{w}}$ denote corresponding Jacobian matrices.

The coupled systems (1.1)–(1.3) and (2.2), (2.3) together with

$$\boldsymbol{q}(x,0)=0\ \text{ a.e. }\ x\in\mathbb{R}, \tag{2.4}$$

represent the formal first-order optimality system for the problem (1.1)–(1.3), in which (1.2) should be solved forward in time from $t=0$ to $t=T$, while the adjoint system (2.2), (2.3) should be solved backward in time.

Note that it is well-known (see, e.g., [6, 8, 33]) that in the case of nonsmooth solutions, the previous equations have to be complemented by variations of possible shock positions in $\boldsymbol{w}$. We refer the reader to the discussion in the Introduction on further references and details on the arising additional equations. Here, we focus on a numerical discretization of the backward problem (2.2), (2.3) in the case of a system of hyperbolic equations.

## 3. Numerical method

In this section, we present the iterative algorithm for the optimal control problem (1.1)–(1.3) based on the formal optimality system derived in Section 2. In order to numerically construct the initial data, which minimize the cost functional (1.1), we generate a sequence $\{\boldsymbol{w}_0^{(m)}(x)\},m=0,1,2,\dots$ of initial conditions as described below. In order to avoid an ambiguity, from now on the solution $\boldsymbol{w}_0$ of the optimization problem (1.1)–(1.3) will be called the *recovered initial data*, while the corresponding solution of the system (1.2) will be referred to as the *recovered solution*.

We assume that the two tolerance values, $\varepsilon_J$ for the cost functional $J$ and $\varepsilon_{\Delta J}$ for its change, have been chosen. Note that the second tolerance parameter $\varepsilon_{\Delta J}$ is needed because the cost functional $J$ may not converge to zero, and then the iterative process

has to be stopped as the change of $J$ becomes insignificant. The following algorithm may be seen as a block Gauß-Seidel iteration. Compared with existing approaches Steps 2, 5, and 6 in the algorithm below include novel contributions.

ALGORITHM 3.1.

*Step 1. Choose an initial guess* $\boldsymbol{w}_0^{(0)}(x)$ *for the initial data* $\boldsymbol{w}_0(x)$. *Set* $m:=0$.

*Step 2. Numerically solve the hyperbolic system (1.2) with the initial state* $\boldsymbol{w}_0(x)=\boldsymbol{w}_0^{(m)}(x)$ *forward in time from* $t=0$ *to* $t=T$ *by the central-upwind scheme described in Section 3.1. We denote the obtained solution by* $\boldsymbol{w}^{(m)}(x,t)$.

*Step 3. Compute the cost functional*

$$J(\boldsymbol{w}^{(m)}(\cdot,T);\boldsymbol{w}_d(\cdot)):=\frac{1}{2}\int\limits_{\mathbb{R}}\left|\boldsymbol{w}^{(m)}(x,T)-\boldsymbol{w}_d(x)\right|^2dx.$$

*Step 4. If either*

$$J(\boldsymbol{w}^{(m)}(\cdot,T);\boldsymbol{w}_d(\cdot))\le\varepsilon_J,$$

*or*

$$m>0 \text{ and } \left|J(\boldsymbol{w}^{(m)}(\cdot,T);\boldsymbol{w}_d(\cdot))-J(\boldsymbol{w}^{(m-1)}(\cdot,T);\boldsymbol{w}_d(\cdot))\right|\le\varepsilon_{\Delta J},$$

*stop the iteration process. Obtained* $\boldsymbol{w}_0^{(m)}(x)$ *will be the approximation to the optimal control.*

*Step 5. Numerically solve the adjoint system (2.2), (2.3) subject to the terminal condition* $\boldsymbol{q}(x,T)=\boldsymbol{w}_0^{(m)}(x)-\boldsymbol{w}_d(x)$ *backward in time from* $t=T$ *to* $t=0$ *using the second-order upwind scheme described in Section 3.2. The solution is denoted by* $\boldsymbol{q}^{(m)}(x,t)$.

*Step 6. (This step may or may not be included.) In order to decrease the total variation of the latest control* $\boldsymbol{w}_0^{(m)}(x)$, *apply one of the nonlinear filters* $\mathcal{F}$ *(described in Section 3.3 below) to it:*

$$\boldsymbol{w}_0^{(m)}(\cdot):=\mathcal{F}\left\{\boldsymbol{w}_0^{(m)}(\cdot)\right\}. \tag{3.1}$$

*Step 7. Update the control* $\boldsymbol{w}_0^{(m)}(x)$ *using either a gradient descent or quasi-Newton method [7, 22, 30].*

*Step 8. Set* $m:=m+1$. *Go to Step 2.*

REMARK 3.1. A choice of the initial guess $\boldsymbol{w}_0^{(0)}(x)$ in Step 1 of the iteration process may affect not only the overall convergence time, but also the numerical optimization result itself (see Example 1 in Section 4.3) because the cost functional (1.3) may have several local minima with respect to $\boldsymbol{w}_0(x)$. Unless otherwise is indicated, in the numerical experiments we choose the initial guess $\boldsymbol{w}_0^{(0)}(x)$ according to the following procedure.

We consider the IVP

$$\begin{aligned}&\frac{\partial\widehat{\boldsymbol{w}}(x,t)}{\partial t}+\frac{\partial\boldsymbol{f}(\widehat{\boldsymbol{w}}(x,t))}{\partial x}=-\boldsymbol{h}(\widehat{\boldsymbol{w}}(x,t),-x,T-t), && x\in\mathbb{R},\quad t\in(0,T],\\ &\widehat{\boldsymbol{w}}(x,0)=\boldsymbol{w}_d(-x), && x\in\mathbb{R},\end{aligned} \tag{3.2}$$

and numerically solve it forward in time by using the central-upwind scheme described in Section 3.1. Then the initial guess $\boldsymbol{w}_0^{(0)}(x)$ is chosen to be the solution of (3.2) at time $t=T$. More precisely, we set

$$\boldsymbol{w}_0^{(0)}(x):=\widehat{\boldsymbol{w}}(-x,T). \tag{3.3}$$

Notice that (3.2) is obtained from (1.2) by formally reverting both $t$ and $x$ and using $\boldsymbol{w}_d$ as the initial condition. Clearly, for a time-reversible PDE this would give the perfect initial guess. It is well-known that the nonlinear conservation laws are not time-reversible because the solution dissipates at the presence of shock discontinuities; see, e.g., [10, 26, 29]. However, the initial guess obtained using the above procedure seems to be the perfect point to start the search for the optimal control.

REMARK 3.2. Instead of the integral least-square cost functional (1.3) in Step 3 of Algorithm 3.1 one may use its smoothed version defined by

$$J_\delta(\boldsymbol{w}(\cdot,T);\boldsymbol{w}_d(\cdot)):=\frac{1}{2}\int\limits_{\mathbb{R}}\left|(\boldsymbol{w}*\varphi_\delta)(x,T)-(\boldsymbol{w}_d*\varphi_\delta)(x)\right|^2dx, \tag{3.4}$$

where $\varphi_\delta:\mathbb{R}\to\mathbb{R}$ is a smoothing kernel satisfying

$$\varphi_\delta(x)=\frac{1}{\delta}\varphi\Big(\frac{x}{\delta}\Big),\quad \delta>0,\quad \int\limits_{\mathbb{R}}\varphi(x)dx=1,$$

and $*$ denotes a componentwise convolution in $x$. In the numerical experiments reported in Section 4, the Gaussian kernel $\varphi_\delta(x)=\frac{1}{\sqrt{2\pi}\delta}e^{-x^2/2\delta^2}$ has been used, but we note that other smooth kernels give similar results.

In the case of $J_\delta$ and a scalar conservation law with convex flux function, rigorous convergence results for several first-order finite-volume methods applied to both the nonlinear forward equation (1.2) and the adjoint equation (2.2) have been established (see, e.g., [1, 8, 32] for further discussion and references).

REMARK 3.3. Note that the full solution $\boldsymbol{q}^{(m)}(x,t)$ of the adjoint system does not need to be stored during the iterations; only the result for $\boldsymbol{q}^{(m)}(x,0)$ at time $t=0$ will be used upon completion of the iteration in Step 5.

REMARK 3.4. Applying a filter to $\boldsymbol{w}_0^{(m)}$ in Step 6 is the way to adjust the current initial guess and thus no formal restrictions on the filter should be enforced. Because the optimal control $\boldsymbol{w}_0$ of the optimization problem (1.1)–(1.3) is generically nonunique, a filter will help to direct the optimization process towards the initial data with the desired properties. However, in order to guarantee the convergence, the filter should be switched off when the values of

$$J(\boldsymbol{w}^{(m)}(\cdot,T);\boldsymbol{w}_d(\cdot))$$

and/or

$$\Big|J(\boldsymbol{w}^{(m)}(\cdot,T);\boldsymbol{w}_d(\cdot))-J(\boldsymbol{w}^{(m-1)}(\cdot,T);\boldsymbol{w}_d(\cdot))\Big|$$

are sufficiently small.

In addition, the filter does not have to be used at each $m$: It can be applied, for example, only if the variation of $\boldsymbol{w}_0^{(m)}(x)$ is large. In fact, because the filter

will be eventually switched off, instead of a filter, one may also consider any type of perturbation (including a random one) of $\boldsymbol{w}_0^{(m)}$ in Step 6, namely any $\mathcal{F}\{\cdot\}$, which will help to make sure that the filtered iterate $\boldsymbol{w}_0^{(m)}$ satisfies the desired properties of the optimal $\boldsymbol{w}_0$.

We would like to emphasize that both the type of the filer $\mathcal{F}$ and the way it is being used in the algorithm may significantly affect the optimization result for the recovered initial data in the case of time-irreversible PDEs. However, it rather should be seen as an advantage because it provides a tool of enforcing convergence to the best (in a certain sense) optimal control $\boldsymbol{w}_0$ in the problem (1.1)–(1.3).

REMARK 3.5. If we use a steepest descent update in Step 7 for some stepsize $\sigma_m > 0$ as

$$\boldsymbol{w}_0^{(m+1)}(x) = \boldsymbol{w}_0^{(m)}(x) - \sigma_m \boldsymbol{q}^{(m)}(x,0),$$

then, due to a global finite-volume approximation of $\boldsymbol{w}$, we obtain a piecewise polynomial control $\boldsymbol{w}_0^{(m+1)}$ in this step of Algorithm 3.1. The fact that the control $\boldsymbol{w}_0^{(m+1)}$ is always piecewise polynomial prevents the accumulation of discontinuities in the forward solution in our algorithm. Clearly, other (higher-order) gradient-based optimization methods can be used to speed up the convergence, especially in the advance stages of the above iterative procedure; see, e.g., [22, 30] for more details.

**3.1. Godunov-type central-upwind scheme for (1.2).** In Step 2 of Algorithm 3.1, the hyperbolic system (1.2) is being solved using the second-order central-upwind scheme introduced in [23]; see also [24, 25]. This scheme is a Riemann-problem-solver-free Godunov-type finite-volume method, which can be applied as an efficient highly accurate "black-box" solver to a wide variety of nonlinear hyperbolic systems.

Here, we briefly describe the scheme. We consider the IVP (1.2). For simplicity, we assume that $\boldsymbol{h}(\boldsymbol{w}(x,t),x,t) \equiv \boldsymbol{0}$ and take a uniform spatial grid with $x_\alpha := \alpha\Delta x$. Given the time level $t \in [0,T]$, we assume that the cell averages over the cells $I_j := [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$,

$$\overline{\boldsymbol{w}}_j(t) \approx \frac{1}{\Delta x}\int\limits_{I_j} \boldsymbol{w}(x,t)dx, \tag{3.5}$$

are available. They are then evolved in time using the semi-discrete central-upwind scheme:

$$\frac{d\overline{\boldsymbol{w}}_j(t)}{dt} = -\frac{\boldsymbol{H}_{j+\frac{1}{2}}(t) - \boldsymbol{H}_{j-\frac{1}{2}}(t)}{\Delta x}, \tag{3.6}$$

where the numerical flux $\boldsymbol{H}_{j+\frac{1}{2}}$ is given by

$$\boldsymbol{H}_{j+\frac{1}{2}} := \frac{a^+_{j+\frac{1}{2}}\boldsymbol{f}(\boldsymbol{w}^-_{j+\frac{1}{2}}) - a^-_{j+\frac{1}{2}}\boldsymbol{f}(\boldsymbol{w}^+_{j+\frac{1}{2}})}{a^+_{j+\frac{1}{2}} - a^-_{j+\frac{1}{2}}} + a^+_{j+\frac{1}{2}}a^-_{j+\frac{1}{2}}\left[\frac{\boldsymbol{w}^+_{j+\frac{1}{2}} - \boldsymbol{w}^-_{j-\frac{1}{2}}}{a^+_{j+\frac{1}{2}} - a^-_{j+\frac{1}{2}}} - \boldsymbol{d}_{j+\frac{1}{2}}\right]. \tag{3.7}$$

From now on we suppress the time-dependence of all indexed quantities in order to shorten the notation. In (3.7), $\boldsymbol{w}^-_{j+\frac{1}{2}}$ and $\boldsymbol{w}^+_{j+\frac{1}{2}}$ are the left- and right-sided values of the piecewise linear reconstruction of $\boldsymbol{w}$,

$$\widetilde{\boldsymbol{w}}(x) := \sum_j \{\overline{\boldsymbol{w}}_j + \boldsymbol{s}_j(x - x_j)\}\chi_{I_j}(x), \quad \chi_{I_j}(x) = \begin{cases} 1, \text{ if } x \in I_j, \\ 0, \text{ otherwise,} \end{cases} \tag{3.8}$$

at the point $x=x_{j+\frac12}$:

$$\boldsymbol{w}^-_{j+\frac12}:=\overline{\boldsymbol{w}}_j+\frac{\Delta x}{2}\boldsymbol{s}_j,\quad \boldsymbol{w}^+_{j+\frac12}:=\overline{\boldsymbol{w}}_{j+1}-\frac{\Delta x}{2}\boldsymbol{s}_{j+1}. \tag{3.9}$$

To make the reconstruction (and thus the entire scheme) non-oscillatory, the slopes $\boldsymbol{s}_j$ have to be computed using a nonlinear limiter. In this paper we use the generalized minmod limiter (see, e.g., [27, 28, 31, 35]):

$$\boldsymbol{s}_j:=\text{minmod}\left(\theta\frac{\overline{\boldsymbol{w}}_{j+1}-\overline{\boldsymbol{w}}_j}{\Delta x},\frac{\overline{\boldsymbol{w}}_{j+1}-\overline{\boldsymbol{w}}_{j-1}}{2\Delta x},\theta\frac{\overline{\boldsymbol{w}}_j-\overline{\boldsymbol{w}}_{j-1}}{\Delta x}\right),\quad \theta\in[1,2], \tag{3.10}$$

where the minmod function is defined as follows:

$$\text{minmod}(z_1,z_2,\ldots):=\begin{cases}\min\limits_j\{z_j\}, & \text{if } z_j>0,\forall j,\\ \max\limits_j\{z_j\}, & \text{if } z_j<0,\forall j,\\ 0, & \text{otherwise.}\end{cases} \tag{3.11}$$

Note that the parameter $\theta$ is used to control the amount of numerical dissipation: Larger values of $\theta$ correspond to less dissipative schemes. Also note that in the vector case, the minmod function (3.11) is applied in a componentwise manner.

The one-sided local speeds $a^-_{j+\frac12}$ and $a^+_{j+\frac12}$ in (3.7) can be estimated by

$$\begin{aligned}a^-_{j+\frac12}&:=\min\Big\{\lambda_1\big(A(\boldsymbol{w}^-_{j+\frac12})\big),\lambda_1\big(A(\boldsymbol{w}^+_{j+\frac12})\big),0\Big\},\\ a^+_{j+\frac12}&:=\max\Big\{\lambda_s\big(A(\boldsymbol{w}^-_{j+\frac12})\big),\lambda_s\big(A(\boldsymbol{w}^+_{j+\frac12})\big),0\Big\},\end{aligned} \tag{3.12}$$

where $\lambda_1<\lambda_2<\ldots<\lambda_s$ are the eigenvalues of the Jacobian $A:=\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}$.

Finally, in the central-upwind flux (3.7), the built-in "anti-diffusion" term $\boldsymbol{d}_{j+\frac12}$ is given by

$$\boldsymbol{d}_{j+\frac12}=\text{minmod}\left(\frac{\boldsymbol{w}^+_{j+\frac12}-\boldsymbol{w}^*_{j+\frac12}}{a^+_{j+\frac12}-a^-_{j+\frac12}},\frac{\boldsymbol{w}^*_{j+\frac12}-\boldsymbol{w}^-_{j+\frac12}}{a^+_{j+\frac12}-a^-_{j+\frac12}}\right), \tag{3.13}$$

where the intermediate values $\boldsymbol{w}^*_{j+\frac12}$ are given by

$$\boldsymbol{w}^*_{j+\frac12}=\frac{a^+_{j+\frac12}\boldsymbol{w}^+_{j+\frac12}-a^-_{j+\frac12}\boldsymbol{w}^-_{j+\frac12}-\big\{\boldsymbol{f}(\boldsymbol{w}^+_{j+\frac12})-\boldsymbol{f}(\boldsymbol{w}^-_{j+\frac12})\big\}}{a^+_{j+\frac12}-a^-_{j+\frac12}}; \tag{3.14}$$

see [23] for details.

The semi-discretization (3.6), (3.7), (3.9)–(3.14) is a system of ODEs, which should be integrated using a (nonlinearly) stable and sufficiently accurate ODE solver. In our numerical experiments, we have used the third-order strong stability preserving (SSP) Runge-Kutta method from [19, 20].

At the end, after solving the ODE system (3.6), (3.7), (3.9)–(3.14) from time $t=0$ to the final time $t=T$, we will obtain the set of cell averages $\overline{\boldsymbol{w}}_j(t^n)$ for $n=0,1,\ldots,N$, and thus the set of $N+1$ piecewise linear interpolants (3.8), reconstructed at the same time levels $t^0,t^1,\ldots,t^N$. These reconstructions are to be stored and then used in the numerical solution of the adjoint system (2.2), (2.3), as it is described in the next section.

**3.2. Upwind scheme for the adjoint system (2.2), (2.3).** In this section, we describe the second-order semi-discrete upwind scheme used in Step 5 of Algorithm 3.1 for solving the adjoint system of PDEs (2.2). Because $\boldsymbol{w}(x,t)$ has been computed in Step 2, the system (2.2) is the following linear system with variable coefficients:

$$\frac{\partial \boldsymbol{q}(x,t)}{\partial t}+A(x,t)\frac{\partial \boldsymbol{q}(x,t)}{\partial x}=B(x,t)\boldsymbol{q}(x,t),\quad x\in\mathbb{R},\quad t\in[0,T), \tag{3.15}$$

where

$$A(x,t):=\frac{\partial \boldsymbol{f}(\boldsymbol{w}(x,t))}{\partial \boldsymbol{w}}\quad\text{and}\quad B(x,t):=-\frac{\partial \boldsymbol{h}(\boldsymbol{w}(x,t),x,t)}{\partial \boldsymbol{w}}. \tag{3.16}$$

To numerically solve the IVP (3.15), (2.3) backward in time, we proceed as follows. We first diagonalize the matrix $A(x,t)$:

$$A(x,t)=R(x,t)\Lambda(x,t)R^{-1}(x,t), \tag{3.17}$$

where $\Lambda(x,t)=\operatorname{diag}\{\lambda_1(x,t),\lambda_2(x,t),\dots,\lambda_s(x,t)\}$. We then introduce the characteristic variables

$$\boldsymbol{\psi}(x,t):=R^{-1}(x,t)\boldsymbol{q}(x,t)$$

and rewrite the system (3.15) as

$$\begin{aligned}&\frac{\partial \boldsymbol{\psi}(x,t)}{\partial t}+\Lambda(x,t)\frac{\partial \boldsymbol{\psi}(x,t)}{\partial x}\\ &=R^{-1}(x,t)\left(B(x,t)R(x,t)-\frac{\partial R(x,t)}{\partial t}-A(x,t)\frac{\partial R(x,t)}{\partial x}\right)\boldsymbol{\psi}(x,t).\end{aligned}$$

Because $\Lambda$ is diagonal, we now design the second-order semi-discrete upwind scheme taking into account that the system is being solved backward in time:

$$\begin{aligned}\frac{d\boldsymbol{\psi}_j(t)}{dt}=&-\left[\Lambda^+(x_j,t)\boldsymbol{\psi}_x^+(t)+\Lambda^-(x_j,t)\boldsymbol{\psi}_x^-(t)\right]\\ &+R^{-1}(x_j,t)\left(B(x_j,t)R(x_j,t)-\frac{\partial R(x_j,t)}{\partial t}-A(x_j,t)\frac{\partial R(x_j,t)}{\partial x}\right)\boldsymbol{\psi}_j(t),\end{aligned} \tag{3.18}$$

where $\boldsymbol{\psi}_j(t):=\boldsymbol{\psi}(x_j,t)$ and

$$\Lambda^{\pm}=\operatorname{diag}\{\lambda_1^{\pm}(x,t),\lambda_2^{\pm}(x,t),\dots,\lambda_s^{\pm}(x,t)\}$$

with $\lambda_i^+:=\max\{\lambda_i,0\}$ and $\lambda_i^-:=\min\{\lambda_i,0\}$.

As before, we use the nonlinear minmod limiter to prevent (minimize) spurious oscillations. The limiter is applied to the second numerical derivatives and thus the values of $\boldsymbol{\psi}_x^{\pm}$ are computed as follows:

$$\begin{aligned}\boldsymbol{\psi}_x^-(t)&=\frac{\boldsymbol{\psi}_j-\boldsymbol{\psi}_{j-1}}{\Delta x}+\operatorname{minmod}\left(\frac{\boldsymbol{\psi}_j-2\boldsymbol{\psi}_{j-1}+\boldsymbol{\psi}_{j-2}}{2\Delta x},\frac{\boldsymbol{\psi}_{j+1}-2\boldsymbol{\psi}_j+\boldsymbol{\psi}_{j-1}}{2\Delta x}\right),\\ \boldsymbol{\psi}_x^+(t)&=\frac{\boldsymbol{\psi}_{j+1}-\boldsymbol{\psi}_j}{\Delta x}-\operatorname{minmod}\left(\frac{\boldsymbol{\psi}_{j+1}-2\boldsymbol{\psi}_j+\boldsymbol{\psi}_{j-1}}{2\Delta x},\frac{\boldsymbol{\psi}_{j+2}-2\boldsymbol{\psi}_{j+1}+\boldsymbol{\psi}_j}{2\Delta x}\right),\end{aligned} \tag{3.19}$$

where the minmod function is defined in (3.11).

It is more convenient to rewrite the semi-discrete scheme (3.18), (3.19) in terms of the original variables $\boldsymbol{q}$. Denoting

$$\boldsymbol{q}_j(t):=\boldsymbol{q}(x_j,t),\quad R_j(t):=R(x_j,t),\quad A^{\pm}(x,t):=R(x,t)\Lambda^{\pm}(x,t)R^{-1}(x,t),$$

defining the matrix

$$P_{j+\frac12}(t):=R_j(t)R_{j+1}^{-1}(t), \tag{3.20}$$

and omitting for simplicity the dependence of all indexed quantities on $t$, we obtain

$$\frac{d\boldsymbol{q}_j}{dt}=-\left[A^+(x_j,t)\boldsymbol{q}_x^+ + A^-(x_j,t)\boldsymbol{q}_x^-\right]+\left(B(x_j,t)-A(x_j,t)\frac{\partial R(x_j,t)}{\partial x}R^{-1}(x_j,t)\right)\boldsymbol{q}_j, \tag{3.21}$$

where

$$\begin{aligned}
\boldsymbol{q}_x^- &= \frac{\boldsymbol{q}_j-P_{j-\frac12}^{-1}\boldsymbol{q}_{j-1}}{\Delta x}\\
&+R_j\,\mathrm{minmod}\left(R_j^{-1}\frac{P_{j+\frac12}\boldsymbol{q}_{j+1}-2\boldsymbol{q}_j+P_{j-\frac12}^{-1}\boldsymbol{q}_{j-1}}{2\Delta x},R_{j-1}^{-1}\frac{P_{j-\frac12}\boldsymbol{q}_j-2\boldsymbol{q}_{j-1}+P_{j-\frac32}^{-1}\boldsymbol{q}_{j-2}}{2\Delta x}\right),\\
\boldsymbol{q}_x^+ &= \frac{P_{j+\frac12}\boldsymbol{q}_{j+1}-\boldsymbol{q}_j}{\Delta x}\\
&-R_j\,\mathrm{minmod}\left(R_j^{-1}\frac{P_{j+\frac12}\boldsymbol{q}_{j+1}-2\boldsymbol{q}_j+P_{j-\frac12}^{-1}\boldsymbol{q}_{j-1}}{2\Delta x},R_{j+1}^{-1}\frac{P_{j+\frac32}\boldsymbol{q}_{j+2}-2\boldsymbol{q}_{j+1}+P_{j+\frac12}^{-1}\boldsymbol{q}_j}{2\Delta x}\right).
\end{aligned} \tag{3.22}$$

Notice that in order to use the semi-discrete scheme (3.21), (3.22), one has to evaluate the matrices $A(x,t)$ and $B(x,t)$, defined in (3.16). Because the computed solution of the IVP (1.2) is only available at the discrete time levels $t^0,t^1,\ldots,t^N$, we will only have $A(x,t^n)$ and $B(x,t^n)$ evaluated according to (3.16) with $\boldsymbol{w}(x,t^n)=\widetilde{\boldsymbol{w}}(x,t^n)$ for $n=0,1,\ldots,N$. Therefore, the ODE system (3.21), (3.22) has to be integrated backward in time using the ODE solver that uses the information from time levels $t^0,t^1,\ldots,t^N$ only—not from any intermediate times. In the numerical experiments reported in Section 4, we have used the one-step two-stages SSP Runge-Kutta (Heun) method ([19, 20]), which satisfies the above requirement.

REMARK 3.6. Note that if we either consider a scalar case of the problem (3.15), (2.3), for which $R(x,t)\equiv1$, or simplify the semi-discrete scheme (3.21), (3.22) by locally freezing the coefficient matrix $A(x,t)$ at the point $(x_j,t)$, that is, by setting $A(x,t)\equiv A(x_j,t)$ and thus $R(x,t)\equiv R(x_j,t)$ and $\partial R(x,t)/\partial x\equiv0$ for $x$ near $x_j$, then the second-order semi-discrete upwind scheme reduces to

$$\frac{d\boldsymbol{q}_j}{dt}=-\left[A^+(x_j,t)\boldsymbol{q}_x^+ + A^-(x_j,t)\boldsymbol{q}_x^-\right]+B(x_j,t)\boldsymbol{q}_j. \tag{3.23}$$

Here, $\boldsymbol{q}_x^{\pm}$ are given by (3.22) with $P_{j+\frac12}\equiv I_s$, $\forall j$, where $I_s$ states for the $s\times s$ identity matrix.

REMARK 3.7. The first-order version of the semi-discrete upwind scheme (3.21), (3.22) is obtained by replacing (3.22) with

$$\boldsymbol{q}_x^-=\frac{\boldsymbol{q}_j-P_{j-\frac12}^{-1}\boldsymbol{q}_{j-1}}{\Delta x},\qquad \boldsymbol{q}_x^+=\frac{P_{j+\frac12}\boldsymbol{q}_{j+1}-\boldsymbol{q}_j}{\Delta x}. \tag{3.24}$$

**3.3. Conservative nonlinear filters.** In this section, we briefly describe conservative nonlinear filters, which may be used in Step 6 of Algorithm 3.1 to reduce the total variation of the iterates $\boldsymbol{w}_0^{(m)}$. These filters were proposed in [11].

The simplest filter applied to the set of cell averages of a conservative variable $\phi$ satisfies the following properties (for details, see [11, Algorithm 2.2]):

1. After the filter has been applied, the total variation of $\phi$,
$$TV(\phi) := \sum_j |\bar{\phi}_{j+1} - \bar{\phi}_j|,$$
must not increase and the number of updated values of $\bar{\phi}_j$ should be as small as possible.
2. If $\bar{\phi}_j$ is a local maximum (minimum) then $\bar{\phi}_j$ is decreased (increased) and one of the neighboring values, either $\bar{\phi}_{j-1}$ or $\bar{\phi}_{j+1}$, is increased (decreased) in a way that $\sum_j \bar{\phi}_j \Delta x_j$ does not change.
3. The filtering processes must not introduce additional artificial oscillations and thus the sign of both $\bar{\phi}_{j+1} - \bar{\phi}_j$ and $\bar{\phi}_j - \bar{\phi}_{j-1}$ should not change.

The way this filter acts on a discrete set of values is schematically shown in figure 3.1.

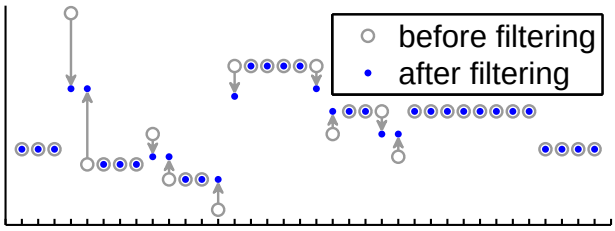


FIG. 3.1. *Conservative nonlinear filter.*

Notice that the filter, described above, is not restricted to a uniform grid case. Because after applying the filter the extrema will consist of at least two neighboring cell averages (see figure 3.1), the equal neighboring values can be combined into new cell averages over larger cells (plateaus) without loss of information. One may consider applying the filter again then as it is described in [11, Algorithm 2.4]. Applying the filter $\ell$ times, combining the cell averages each time into platos, leads to the result that consists of plateaus of length $\ell+1$ (at least). In the numerical experiments, presented in Section 4, we have used the conservative nonlinear filter with $\ell = 10$ (examples 2b, 4b, and 5b).

## 4. Numerical optimization results

In this section, the performance of the optimization method described in Algorithm 3.1 is demonstrated on several examples of PDE-constrained control problems governed by the systems of conservation laws. We begin with considering the optimization problem (1.1)–(1.3) governed by the Euler equations of gas dynamics and solve this problem with a smooth prescribed solution. We then consider two examples with the system (1.2) being the isothermal gas dynamics equations. Finally, we

consider several examples of the control problem governed by Euler equations of gas dynamics with discontinuous prescribed terminal states.

In all of the numerical experiments except for the ones reported in examples 1 and 2, we choose the initial guess $\boldsymbol{w}_0^{(0)}$ according to (3.3) by numerically solving the IVP (3.2) using the central-upwind scheme described in Section 3.1. In Example 1, we illustrate that the choice of the initial guess is very important in the nonsmooth cases.

The forward IVP (1.2) is numerically solved by the same central-upwind scheme. We have used the minmod parameter $\theta=1.3$ in all of the numerical experiments except for Example 4, where $\theta=1.1$ has been used.

In all of the numerical experiments except for Example 2, we select a sufficiently large computational domain so that the solution remains flat at its endponts. We then use the Neumann boundary conditions together with the ghost cell technique to minimize any possible reflected waves.

Notice that if the filter (Step 6) in Algorithm 3.1 is switched on, the same notation $\boldsymbol{w}_0^{(m)}$ is used for both filtered and unfiltered controls. In all of the numerical examples, whenever the filter has been employed for the iteration step $m$, we plot the cost functional (1.3) (or its smoothed version (3.4)) that corresponds to the filtered initial data $\boldsymbol{w}_0^{(m)}$.

**4.1. Isothermal gas dynamics equations.** We consider the optimization problem (1.1)–(1.3) governed by the isothermal gas dynamics equations. We minimize the cost functional,

$$J(\boldsymbol{w}(\cdot,T);\boldsymbol{w}_d(\cdot))=\frac{1}{2}\int\limits_{\mathbb{R}}\left|(\rho(x,T),\mathcal{M}(x,T))^T-(\rho_d(x),\mathcal{M}_d(x))^T\right|^2dx, \tag{4.1}$$

or its smoothed version (3.4) subject to the system of isothermal gas dynamics equations,

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \mathcal{M}\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\mathcal{M}\\ \rho u^2+p\end{bmatrix}=\boldsymbol{0},\quad p=c^2\rho,\quad (x,t)\in\mathbb{R}\times(0,T]. \tag{4.2}$$

Here, $\rho$, $u$, $\mathcal{M}\equiv\rho u$, and $p$ are the density, velocity, momentum, and pressure, respectively, and $c$ is a positive constant, which has been taken $c=1$ in the numerical experiments reported in examples 3 and 4. Here,

$$\boldsymbol{w}(x,t)=(\rho,\mathcal{M})^T,\quad \boldsymbol{f}(\boldsymbol{w})=(\mathcal{M},\rho u^2+p)^T,\quad \boldsymbol{h}(\boldsymbol{w},x,t)=\boldsymbol{0},$$

and the terminal state,

$$\boldsymbol{w}_d(x):=\Big(\rho_d(x),\mathcal{M}_d(x)\equiv\rho_d(x)u_d(x)\Big)^T,$$

is as before prescribed at time $t=T$.

To implement the upwind scheme from Section 3.2 for the adjoint problem (2.2), (2.3), we compute the matrices $A(x,t)$ and $B(x,t)$, which are

$$A(x,t)=\begin{bmatrix}0 & 1\\ c^2-u^2 & 2u\end{bmatrix},\quad B(x,t)\equiv 0.$$

Then, the diagonalization (3.17) of $A(x,t)$ results in

$$R(x,t)=\begin{bmatrix}1 & 1\\ u-c & u+c\end{bmatrix},\quad \Lambda(x,t)=\begin{bmatrix}u-c & 0\\ 0 & u+c\end{bmatrix},\quad R^{-1}(x,t)=\frac{1}{2c}\begin{bmatrix}c+u & -1\\ c-u & 1\end{bmatrix}.$$

Finally, we numerically solve the adjoint equation (3.15) by the semi-discrete upwind scheme (3.21), (3.22), where the matrix $P_{j+\frac{1}{2}}$ defined in (3.20) and its inverse $P^{-1}_{j+\frac{1}{2}}$ are

$$P_{j+\frac{1}{2}} = \begin{bmatrix} 1 & 0 \\ u_j - u_{j+1} & 1 \end{bmatrix}, \quad P^{-1}_{j+\frac{1}{2}} = \begin{bmatrix} 1 & 0 \\ u_{j+1} - u_j & 1 \end{bmatrix},$$

and because $c$ is constant,

$$\frac{\partial R(x_j,t)}{\partial x} = \frac{\partial u(x_j,t)}{\partial x} \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

In the latter formula, we use the minmod function to evaluate the velocity derivatives:

$$\frac{\partial u(x_j,t)}{\partial x} = \text{minmod}\left(\theta \frac{u_{j+1}-u_j}{\Delta x}, \frac{u_{j+1}-u_{j-1}}{2\Delta x}, \theta \frac{u_j - u_{j-1}}{\Delta x}\right).$$

In the numerical experiments reported below, we have chosen the same value of the parameter $\theta$, which has been used in (3.9), (3.10).

**4.2. Euler equations of gas dynamics.** In order to numerically solve the control problem (1.1)–(1.3) constrained by the Euler equations of gas dynamics, we minimize the following functional:

$$J(\boldsymbol{w}(\cdot,T);\boldsymbol{w}_d(\cdot)) = \frac{1}{2}\int_{\mathbb{R}} \left|(\rho(x,T),\mathcal{M}(x,T),E(x,T))^T - (\rho_d(x),\mathcal{M}_d(x),E_d(x))^T\right|^2 dx. \tag{4.3}$$

Here,

$$\boldsymbol{w}(x,t) = (\rho,\mathcal{M},E)^T, \quad \boldsymbol{f}(\boldsymbol{w}) = (\mathcal{M},\rho u^2 + p, u(E+p))^T, \quad \boldsymbol{h}(\boldsymbol{w},x,t) = \boldsymbol{0},$$

where $\rho$, $u$, $\mathcal{M} \equiv \rho u$, $p$, and $E$ are the density, velocity, momentum, pressure, and total energy, respectively. The governing hyperbolic system of conservation laws is

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \mathcal{M} \\ E \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \mathcal{M} \\ \rho u^2 + p \\ u(E+p) \end{bmatrix} = \boldsymbol{0}, \quad E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2, \quad (x,t) \in I \times (0,T], \tag{4.4}$$

where $\gamma$ is an adiabatic constant taken to be $\gamma = 1.4$, which corresponds to diatomic ideal gases. The terminal state,

$$\boldsymbol{w}_d(x) := \left(\rho_d(x), \mathcal{M}_d(x) = \rho_d(x)u_d(x), E_d(x) = \frac{p_d(x)}{\gamma - 1} + \frac{1}{2}\rho_d(x)u_d^2(x)\right)^T,$$

is as before prescribed at time $t = T$.

To implement the upwind scheme from Section 3.2 for the adjoint problem (2.2), (2.3), we compute the matrices $A(x,t)$ and $B(x,t)$, which are

$$A(x,t) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}u^2 & (3-\gamma)u & \gamma - 1 \\ -\frac{\gamma E}{\rho}u + (\gamma-1)u^3 & \frac{\gamma E}{\rho} + \frac{3}{2}(1-\gamma)u^2 & \gamma u \end{bmatrix}, \quad B(x,t) \equiv 0.$$

Then, the diagonalization (3.17) of $A(x,t)$ results in

$$\Lambda(x,t)=\mathrm{diag}\{\lambda_1(x,t),\lambda_2(x,t),\lambda_3(x,t)\},\quad \lambda_1(x,t)=u-c,\ \lambda_2(x,t)=u,\ \lambda_3(x,t)=u+c, \tag{4.5}$$

where $c:=\sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound. The matrices $R(x,t)$ and $R^{-1}(x,t)$ defined in (3.17) are given by

$$R(x,t)=\begin{bmatrix}1 & 1 & 1\\ u-c & u & u+c\\ \mathcal{H}-cu & \frac{1}{2}u^2 & \mathcal{H}+cu\end{bmatrix},\quad R^{-1}(x,t)=\begin{bmatrix}\frac{1}{2}(b_1+\frac{u}{c}) & \frac{1}{2}(-b_2u-\frac{1}{c}) & \frac{1}{2}b_2\\ 1-b_1 & b_2u & -b_2\\ \frac{1}{2}(b_1-\frac{u}{c}) & \frac{1}{2}(-b_2u+\frac{1}{c}) & \frac{1}{2}b_2\end{bmatrix},$$

where $b_1:=\frac{1}{2}b_2u^2$, $b_2:=\frac{\gamma-1}{c^2}$, and the enthalpy $\mathcal{H}:=\frac{E+p}{\rho}$.

In examples 2, 5, and 7, we numerically solve the adjoint equation (3.15) by the semi-discrete upwind scheme (3.23), where $\boldsymbol{q}_x^{\pm}(t)$ are given by (3.22) with $P_{j+\frac{1}{2}}\equiv I_3$, $\forall j$.

Finally, in the numerical experiments reported in Example 6, we have used the semi-discrete upwind scheme (3.21), (3.22) to solve the adjoint problem (3.15). In this case, the matrix $P_{j+\frac{1}{2}}$ defined in (3.20) is given by

$$P_{j+\frac{1}{2}}=\begin{bmatrix}1 & 0 & 0\\ u_j-\frac{c_j}{c_{j+1}}u_{j+1} & \frac{c_j}{c_{j+1}} & 0\\ \frac{1}{2}\left(u_j-\frac{c_j}{c_{j+1}}u_{j+1}\right)^2 & \frac{c_j}{c_{j+1}}\left(u_j-\frac{c_j}{c_{j+1}}u_{j+1}\right) & \left(\frac{c_j}{c_{j+1}}\right)^2\end{bmatrix},$$

and thus,

$$P^{-1}_{j+\frac{1}{2}}=\begin{bmatrix}1 & 0 & 0\\ u_{j+1}-\frac{c_{j+1}}{c_j}u_j & \frac{c_{j+1}}{c_j} & 0\\ \frac{1}{2}\left(u_{j+1}-\frac{c_{j+1}}{c_j}u_j\right)^2 & \frac{c_{j+1}}{c_j}\left(u_{j+1}-\frac{c_{j+1}}{c_j}u_j\right) & \left(\frac{c_{j+1}}{c_j}\right)^2\end{bmatrix},$$

and the derivative $\frac{\partial R(x_j,t)}{\partial x}$ in (3.21) is computed by substituting $x=x_j$ into

$$\frac{\partial R}{\partial x}=\frac{\partial u}{\partial x}\begin{bmatrix}0 & 0 & 0\\ 1 & 1 & 1\\ -c & u & c\end{bmatrix}+\frac{\partial c}{\partial x}\begin{bmatrix}0 & 0 & 0\\ -1 & 0 & 1\\ -u & 0 & u\end{bmatrix}+\frac{\partial \mathcal{H}}{\partial x}\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 1 & 0 & 1\end{bmatrix}.$$

The derivatives on the right-hand side of the latter formula are evaluated using the minmod limiter as follows:

$$\frac{\partial g(x_j,t)}{\partial x}=\mathrm{minmod}\left(\theta\frac{g_{j+1}-g_j}{\Delta x},\frac{g_{j+1}-g_{j-1}}{2\Delta x},\theta\frac{g_j-g_{j-1}}{\Delta x}\right),$$

where the function $g$ is either $u$, $c$, or $\mathcal{H}$. In Example 6, we have taken $\theta=1.3$.

### 4.3. Numerical examples.

**Example 1: Inviscid Burgers equation with discontinuous $w_d$.** Let us consider a simple scalar example, where the optimal control problem is governed by the inviscid Burgers equation,

$$\frac{\partial w(x,t)}{\partial t}+\frac{\partial}{\partial x}\left(\frac{w(x,t)^2}{2}\right)=0, \tag{4.6}$$

subject to the discontinuous terminal datum

$$w_d(x) = \begin{cases} 1, & \text{if } 0 \le x \le 1, \\ 0, & \text{otherwise,} \end{cases} \tag{4.7}$$

prescribed at $T = 1$. We solve the problem in the interval $[-1,2]$ on a uniform grid with $\Delta x = 1/200$. We take $\varepsilon_J = 3 \cdot 10^{-2}$ and $\varepsilon_{\Delta J} = 10^{-7}$ in the stopping criteria in Step 4 of Algorithm 3.1.

One can show that the optimal solution is obtained for

$$w_0(x) = \begin{cases} 1, & \text{if } -\frac{1}{2} < x \le 0, \\ 1-x, & \text{if } 0 < x \le 1, \\ 0, & \text{otherwise,} \end{cases}$$

for which

$$w(x,T) = \begin{cases} x + \frac{1}{2}, & \text{if } -\frac{1}{2} \le x \le \frac{1}{2}, \\ 1, & \text{if } -\frac{1}{2} \le x \le 1, \\ 0, & \text{otherwise} \end{cases}$$

and $J(w(\cdot,T), w_d(\cdot)) = 1/24$.

We first take the initial guess $w_0^{(0)}$ according to (3.3) and show it together with the corresponding solution $w^{(0)}(x,1)$ in figure 4.1. As one can see the optimal solution is quite accurately recovered with no iterations required just by this initial guess. The convergence of Algorithm 3.1 (with the filter in Step 6 being switched off) is demonstrated in figures 4.2 and 4.4 (left).

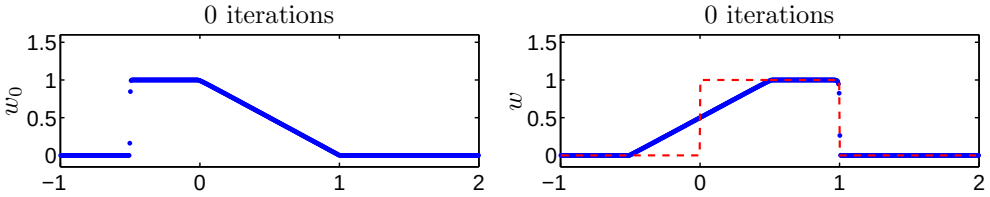


FIG. 4.1. *Example 1: Left: Initial guess $w_0^{(0)}(x)$ (plotted with points) obtained according to (3.3); Right: Recovered solution $w^{(0)}(x,1)$ (plotted with points) and the terminal state $u_d(x)$ (dashed line).*

On the other hand, if one takes a different initial guess, the optimal solution may or may not be recovered. For instance, let us take

$$w_0^{(0)}(x) \equiv 0. \tag{4.8}$$

The recovered solution is now different (see figure 4.3), and as one can see in figure 4.4 (right), the value of the cost functional stabilizes at about 0.16593, which is about 4 times larger than the actual minimal value. The reason for the failure of the initial guess (4.8) is that the studied minimization problem may have several local minima. This stresses the importance of selecting the initial guess according to (3.3) in the case of discontinuous terminal state.

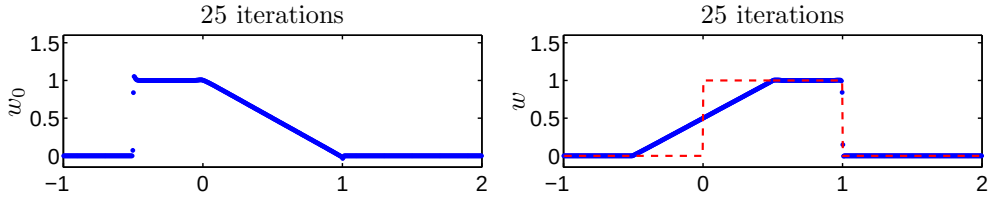


FIG. 4.2. *Example 1: Left: $w_0^{(25)}(x)$ (plotted with points); Right: Recovered solution $w^{(25)}(x,1)$ (plotted with points) and the terminal state $w_d(x)$ (dashed line). The initial guess is chosen according to (3.3).*

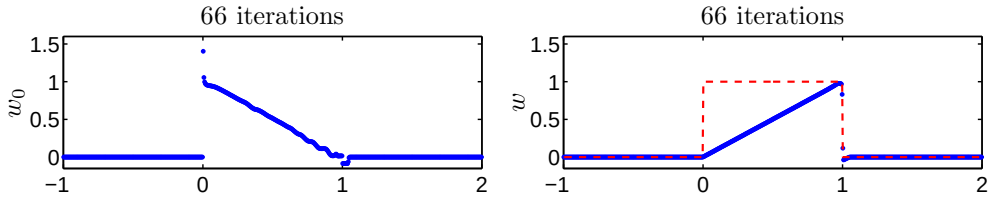


FIG. 4.3. *Example 1: Left: $w_0^{(66)}(x)$ (plotted with points); Right: Recovered solution $w^{(66)}(x,1)$ (plotted with points) and the terminal state $w_d(x)$ (dashed line). The initial guess is chosen according to (4.8).*

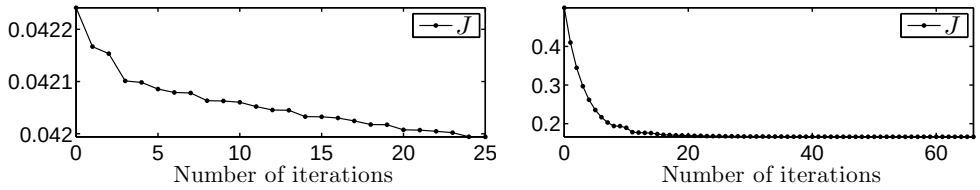


FIG. 4.4. *Example 1: Dependence of the cost functional for the initial guess, $w_0^{(0)}(x)$, given by (3.3) (left) and by (4.8) (right).*

**Example 2: $\boldsymbol{w}_d$ is a smooth solution of the Euler equations.** We first consider the optimization problem (1.1), (4.3), (4.4) in the case of smooth terminal state $\boldsymbol{w}_d(x)=(\rho_d(x),\mathcal{M}_d(x),E_d(x))^T$, which is a solution of a time-reversible IVP for the Euler equation of gas dynamics.

We solve the problem in the interval $[0,1]$ with the periodic boundary conditions. The terminal state is the solution of the system (4.4) subject to the smooth 1-periodic initial data

$$(\rho(x,0),u(x,0),p(x,0))=(1+0.2\sin(2\pi x),1,1).$$

It is easy to show that the solution of this IVP admits a unique smooth solution

$$(\rho(x,t),u(x,t),p(x,t))=(1+0.2\sin(2\pi(x-t)),1,1),\quad (x,t)\in[0,1]\times[0,T],$$

for any $T$. In our numerical experiments, we have taken $T=0.16$ and thus the corre-

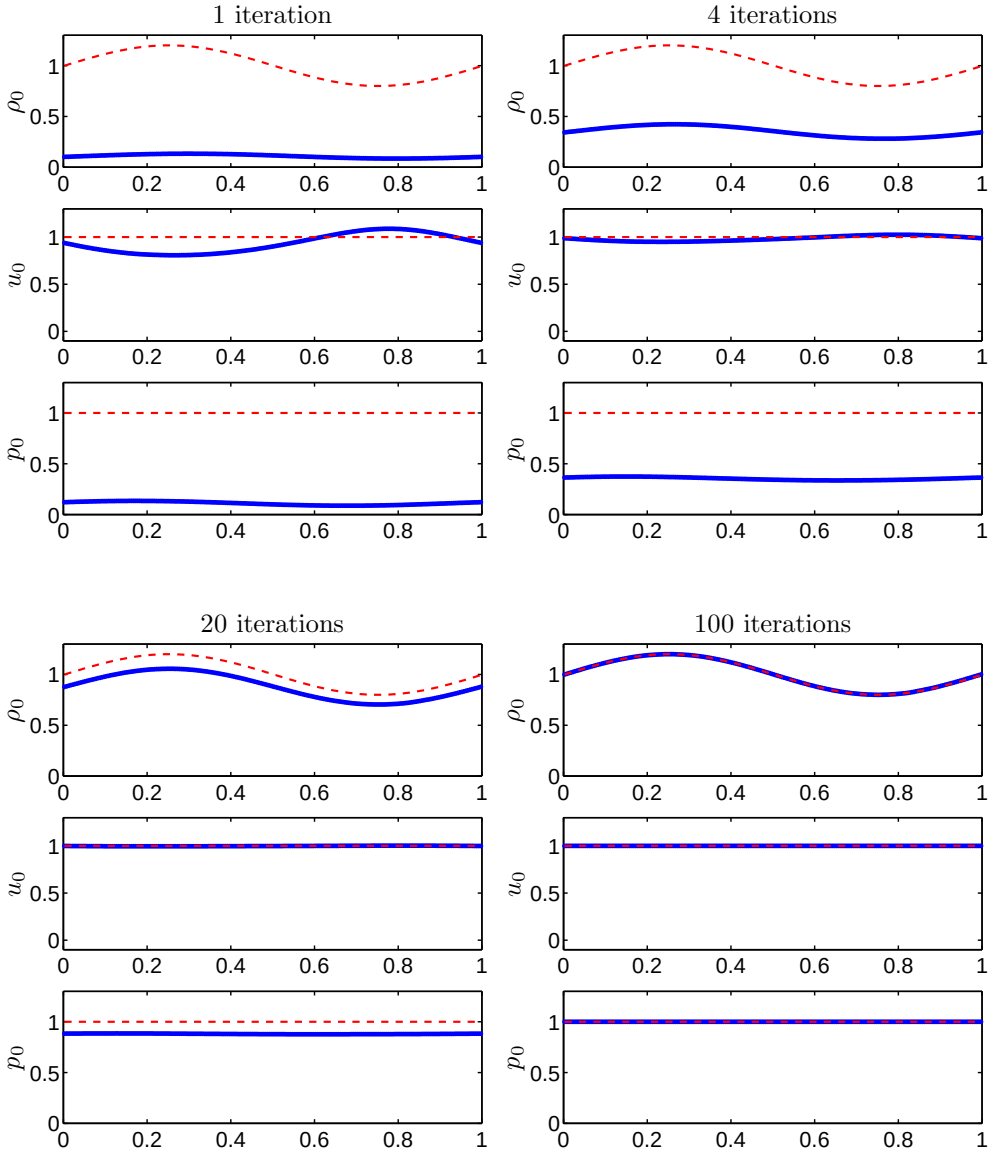


FIG. 4.5. *Example 2: Recovered initial data* $\left(\rho_0^{(m)}(x), u_0^{(m)}(x), p_0^{(m)}(x)\right)$ *for* $m = 1, 4, 20, 100$ *(plotted with points) and the exact initial data (dashed line).*

sponding terminal state is

$$(\rho_d(x), u_d(x), p_d(x)) = (1 + 0.2\sin(2\pi(x - 0.16), 1, 1). \tag{4.9}$$

Because the solution of the IVP studied in this example is time-reversible, the initial guess chosen in accordance with (3.2), (3.3) would be the optimal control for the problem (1.1), (4.3), (4.4) and no iterations would be required. In order to demonstrate the convergence of Algorithm 3.1 (implemented without Step 6), we

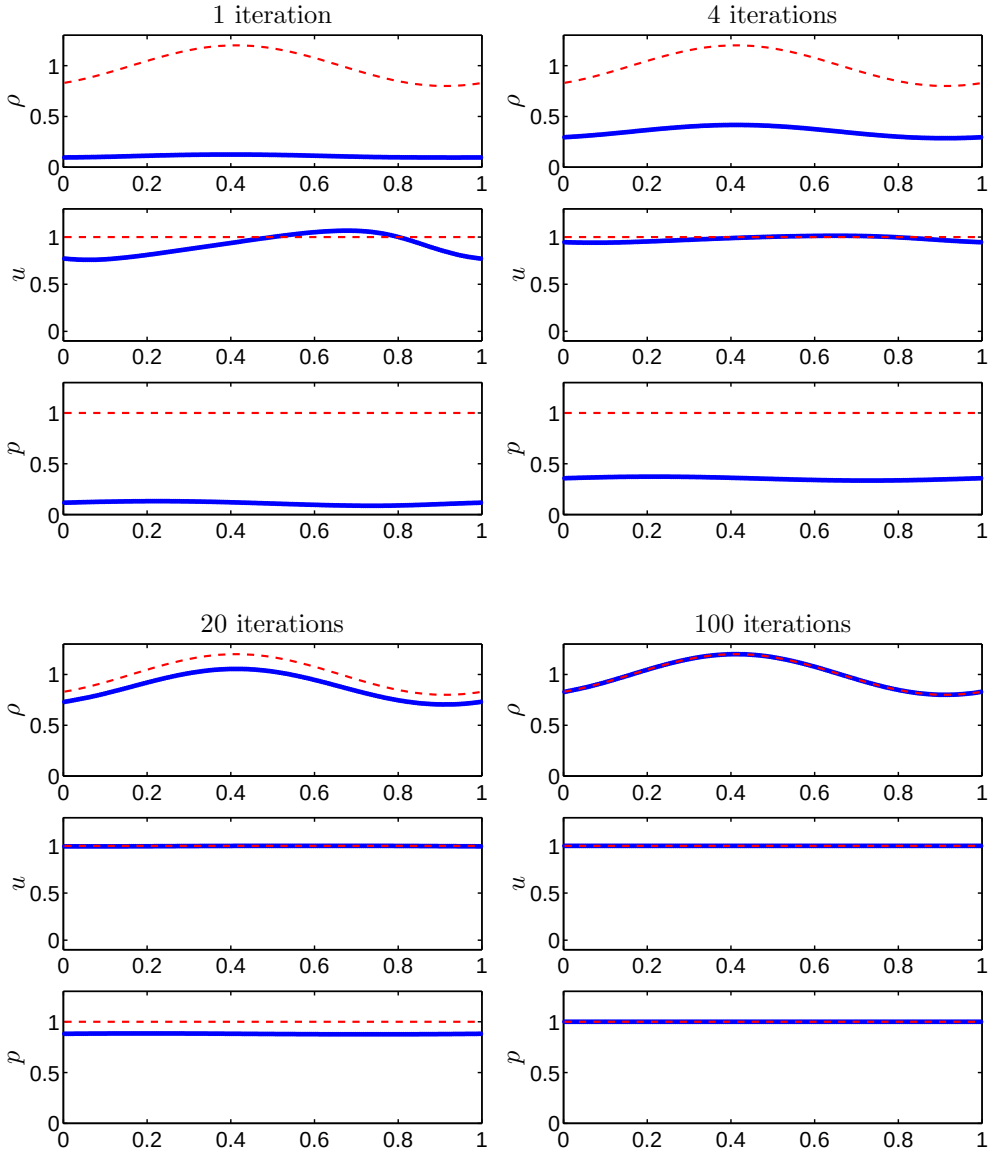


FIG. 4.6. *Example 2: Recovered solution* $\big(\rho^{(m)}(x,0.16),p^{(m)}(x,0.16),u^{(m)}(x,0.16)\big)$ *for* $m=1,4,20,100$ *(plotted with points) and the terminal state* $\big(\rho_d(x),u_d(x),p_d(x)\big)$ *(dashed line).*

choose the following constant initial guess:

$$\big(\rho_0^{(0)}(x),u_0^{(0)}(x),p_0^{(0)}(x)\big)\equiv(0.01,0,0.01). \tag{4.10}$$

The obtained numerical results for the recovered initial data $\big(\rho_0^{(m)}(x),u_0^{(m)}(x),p_0^{(m)}(x)\big)$ are shown in figure 4.5 for $m=1,4,20,100$. The corresponding recovered solutions at $T=0.16$ are presented in figure 4.6. Finally, in figure 4.7 we show the behavior of the $L^1$-errors and minimization functional (4.3)

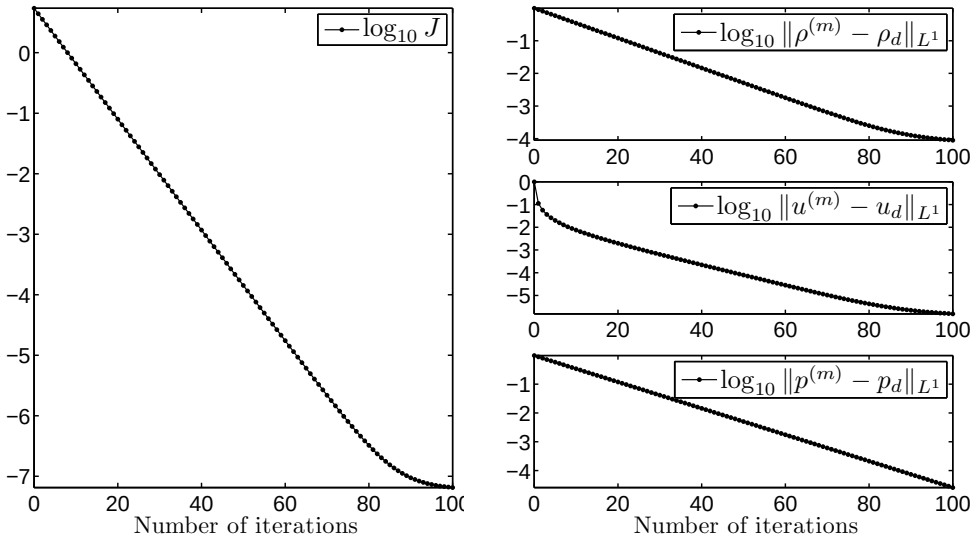


FIG. 4.7. *Example 2: Dependence of the logarithm of the cost functional (4.3) (left) and the $L^1$-errors (right) on the number of iterations.*

using a logarithmic scale. All of the computations have been performed using a uniform grid with $\Delta x = 1/200$.

As one can see in figures 4.5–4.7, even for the "unprepared" initial guess (4.10), our method achieves very high resolution of all components of the solution.

**Example 3: $\boldsymbol{w}_d$ is a continuous solution of the isothermal gas dynamics equations.** In this example, we consider the optimization problem (1.1), (4.1) governed by the isothermal gas dynamics equation (4.2) with the continuous terminal state $\boldsymbol{w}_d(x) = (\rho_d(x), \mathcal{M}_d(x))^T$ prescribed at time $T = 0.15$. We solve the problem in the interval $[-0.5, 1.5]$ and use a uniform grid with $\Delta x = 1/200$.

The terminal state $\boldsymbol{w}_d$ is a solution of the isothermal gas dynamics equation (4.2) subject to the following discontinuous initial data:

$$(\rho(x,0), u(x,0)) = \begin{cases} (1,-2), & \text{if } x \le 0.5, \\ (1,2), & \text{if } x > 0.5. \end{cases} \tag{4.11}$$

In the experiments, we obtain $\boldsymbol{w}_d$ numerically using the second-order semi-discrete central-upwind scheme (Section 3.1) on a uniform grid with $\Delta x = 1/200$. The optimization is performed on the same uniform grid with $\varepsilon_J = 3 \cdot 10^{-5}$ and $\varepsilon_{\Delta J} = 10^{-12}$ in the stopping criteria in Step 4 of Algorithm 3.1.

The initial guess $\big(\rho_0^{(0)}(x), u_0^{(0)}(x)\big)$ obtained by (3.3) is shown in figure 4.8 (left), and the corresponding solution, $\big(\rho^{(0)}(x,0.15), u^{(0)}(x,0.15)\big)$, is shown in figure 4.8 (right). We would like to point out that it is not surprising that the discrepancy in the control $\boldsymbol{w}_0^{(0)}$ and the original initial data (4.11) is located near $x = 0$ (see figure 4.8, left). One can easily see that the solution of the IVP (4.2), (4.11) is not time-reversible and it is discontinuous exactly at $(x,t) = (0,0)$.

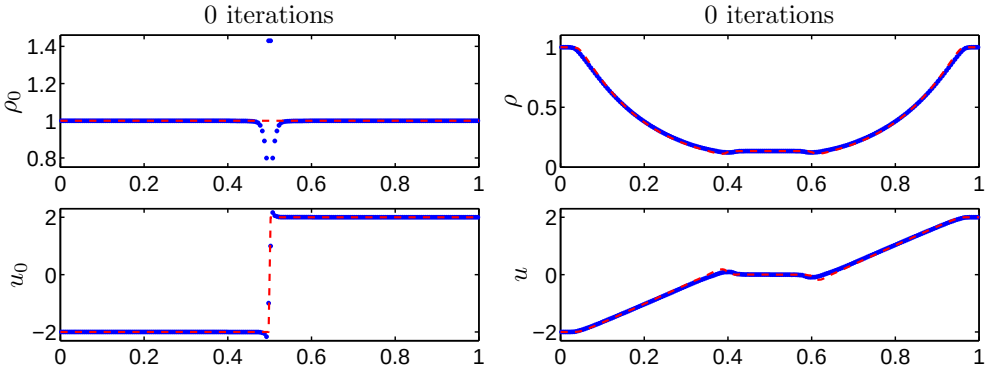


FIG. 4.8. *Example 3: Left: Initial guess* $\left(\rho_0^{(0)}(x),u_0^{(0)}(x)\right)$ *(plotted with points) and the initial data (4.11) (dashed line); Right: Recovered solution* $\left(\rho^{(0)}(x,0.15),u^{(0)}(x,0.15)\right)$ *(plotted with points) and the terminal state* $\left(\rho_d(x),u_d(x)\right)$ *(dashed line).*

Though the results presented in figure 4.8 already look quite good, we would like to improve the solution by applying Algorithm 3.1.

*Example 3a:* We first implement Algorithm 3.1 without Step 6, that is, with the filter being switched off. The recovered initial data $\left(\rho_0^{(2361)}(x),u_0^{(2361)}(x)\right)$ are shown in figure 4.9 (left) and the corresponding recovered solution is plotted in figure 4.9 (right). In figure 4.10, we demonstrate the behavior of the $L^1$-errors together with the cost functional (4.1) for $m=1,\ldots,9750$ iterations using a logarithmic scale. As one can clearly see, given the tolerance values the solution has been fully recovered. Moreover, the cost functional keeps decreasing (see figure 4.10, left), and the recovered solution, in fact, can be further improved. The initial data, however, do not fully match the original initial conditions (4.11). To decrease the total variation of the recovered initial density, we switch on the filter.

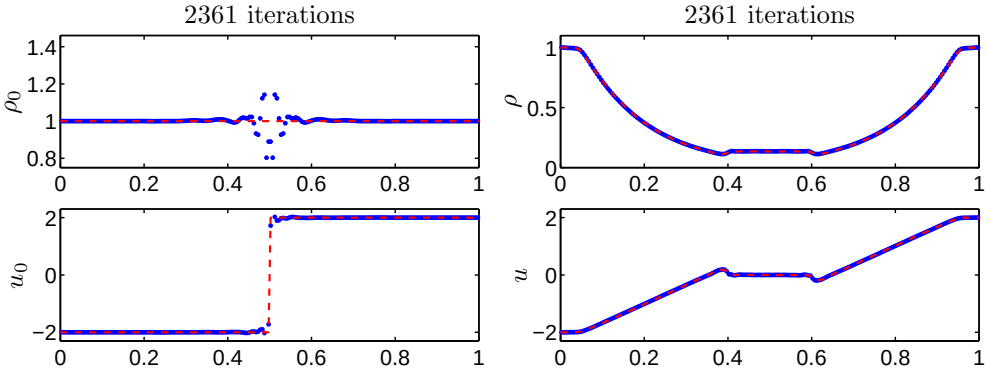


FIG. 4.9. *Example 3a: Left:* $\left(\rho_0^{(2361)}(x),u_0^{(2361)}(x)\right)$ *(plotted with points) and the initial data (4.11) (dashed line); Right: Recovered solution* $\left(\rho^{(2361)}(x,0.15),u^{(2361)}(x,0.15)\right)$ *(plotted with points) and the terminal state* $\left(\rho_d(x),u_d(x)\right)$ *(dashed line).*

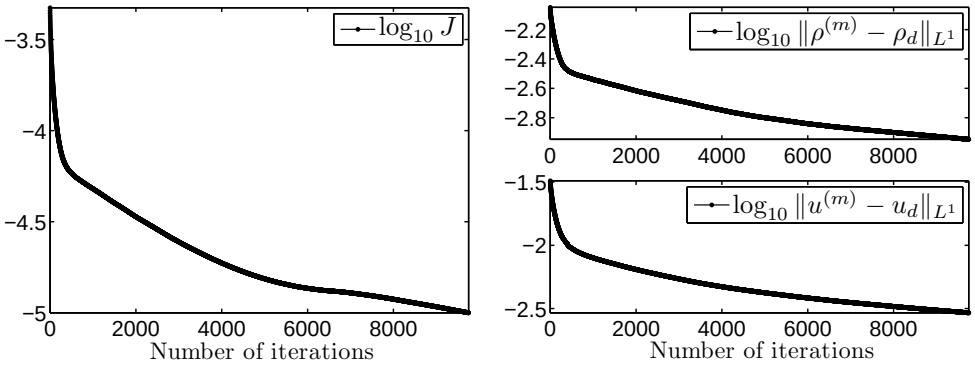


FIG. 4.10. *Example 3a: Dependence of the logarithm of the cost functional (4.1) (left) and the $L^1$-errors (right) on the number of iterations.*

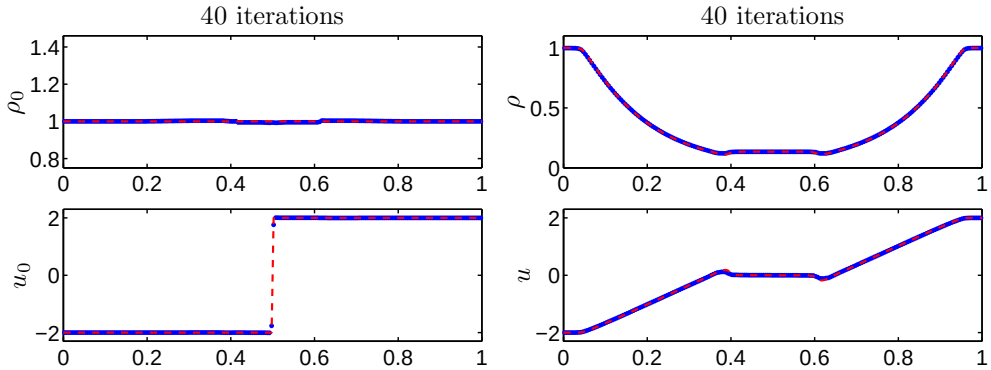


FIG. 4.11. *Example 3b: Left: $\left(\rho_0^{(40)}(x), u_0^{(40)}(x)\right)$ (plotted with points) and the initial data (4.11) (dashed line); Right: Recovered solution $\left(\rho^{(40)}(x,0.15), u^{(40)}(x,0.15)\right)$ (plotted with points) and the terminal state $\left(\rho_d(x), u_d(x)\right)$ (dashed line).*

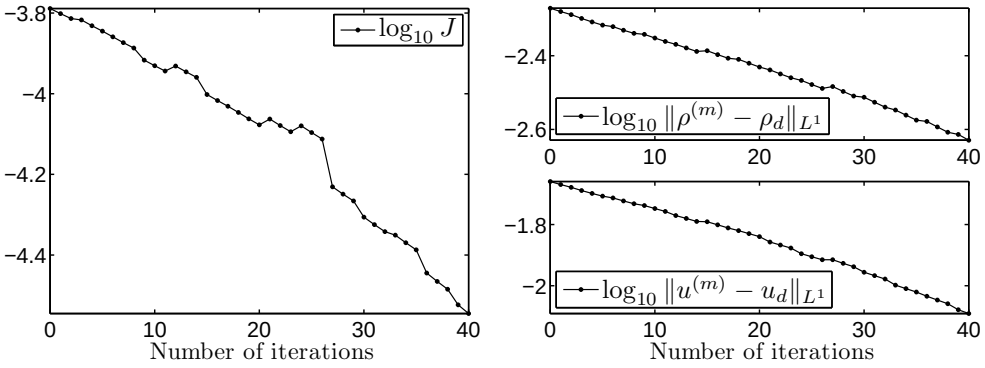


FIG. 4.12. *Example 3b: Dependence of the logarithm of the cost functional (4.1) (left) and the $L^1$-errors (right) on the number of iterations.*

*Example 3b:* We now switch the filter in Step 6 of Algorithm 3.1 on. We implement the filter with $\ell=10$ (see Section 3.3 for details). The filter is applied to variables $\rho_0^{(m)}$ and $u_0^{(m)}$ at every third step starting from $m=0$.

The recovered initial data $\boldsymbol{w}_0^{(40)}(x)$ and solution $\boldsymbol{w}^{(40)}(x,0.15)$ are shown in figure 4.11, where one can see a remarkably sharp recovery of the discontinuous initial data. The $L^1$-errors and the cost functional (4.3) are plotted in figure 4.12 for $m=1,\dots,64$ iterations using a logarithmic scale. Because the filter is used, the curves in figure 4.12 do not decay in such a smooth and monotone manner as they do in the unfiltered case (figure 4.10), but both the cost functional $J$ and $L^1$-errors decrease much faster now though they occasionally have small jumps up. The jumps of the cost functional are due to the filter which shifts the control towards the initial data with smaller total variation so that the control update in Step 7 of Algorithm 3.1 results in the convergence towards the initial data, which are very similar to (4.11).

**Example 4: $\boldsymbol{w}_d$ is a discontinuous solution of the isothermal gas dynamics equations.** In this example, we consider the optimization problem (1.1), (4.1), (4.2) with the terminal state $\boldsymbol{w}_d(x)=(\rho_d(x),\mathcal{M}_d(x))^T$ containing two shock waves (it is prescribed at $T=0.2$). We again solve the problem in the interval $[-0.5,1.5]$ and use a uniform grid with $\Delta x=1/200$.

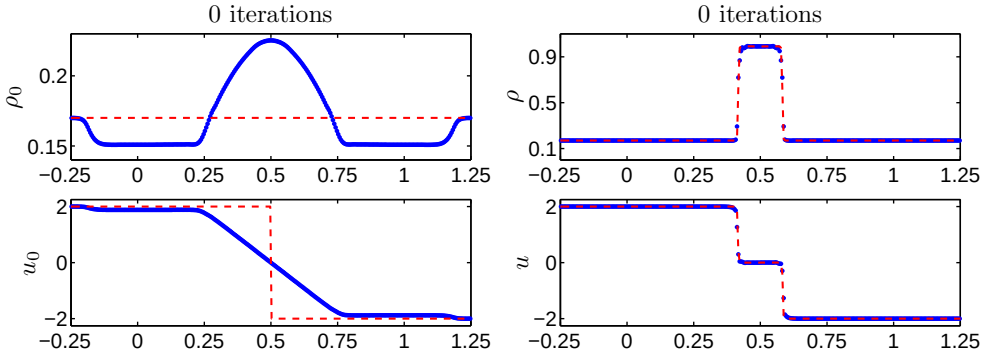


FIG. 4.13. *Example 4: Left: Initial guess* $\big(\rho_0^{(0)}(x),u_0^{(0)}(x)\big)$ *(plotted with points) and the initial data (4.12)(dashed line); Right: Recovered solution* $\big(\rho^{(0)}(x,0.2),u^{(0)}(x,0.2)\big)$ *(plotted with points) and the terminal state* $\big(\rho_d(x),u_d(x)\big)$ *(dashed line).*

The terminal state is a solution of (4.2) subject to the following Riemann initial data:

$$(\rho(x,0),u(x,0))=\begin{cases}(0.17,2), & \text{if } x\le 0.5,\\ (0.17,-2), & \text{if } x>0.5.\end{cases} \tag{4.12}$$

As in Example 3, we obtain $\boldsymbol{w}_d$ numerically using the second-order semi-discrete central-upwind scheme on a uniform grid with $\Delta x=1/200$. The optimization is performed on the same uniform grid with $\varepsilon_J=3\cdot10^{-5}$ and $\varepsilon_{\Delta J}=10^{-12}$ in the stopping criteria in Step 4 of Algorithm 3.1.

The initial guess obtained by (3.3) and the corresponding solution, $\big(\rho^{(0)}(x,0.2),u^{(0)}(x,0.2)\big)$, are shown in figure 4.13. As one can see there, the solution is not well recovered near the discontinuities yet. In order to improve the result, we implement the optimization Algorithm 3.1 with the filter in Step 6 being switched

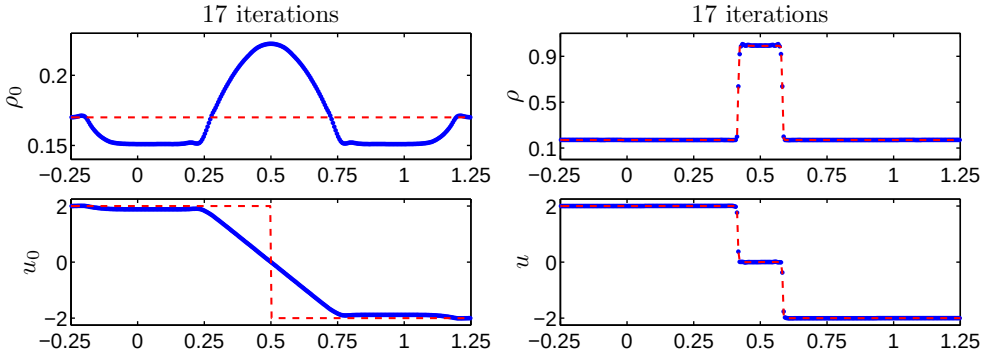


FIG. 4.14. *Example 4: Left:* $\left(\rho_0^{(17)}(x), u_0^{(17)}(x)\right)$ *(plotted with points) and the initial data (4.12) (dashed line); Right: Recovered solution* $\left(\rho^{(17)}(x,0.2), u^{(17)}(x,0.2)\right)$ *(plotted with points) and the terminal state* $\left(\rho_d(x), u_d(x)\right)$ *(dashed line).*

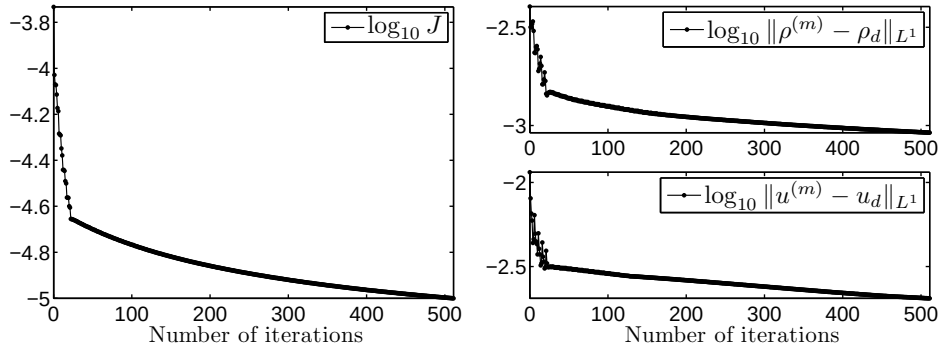


FIG. 4.15. *Example 4: Dependence of the logarithm of the cost functional (4.1) (left) and the* $L^1$*-errors (right) on the number of iterations.*

off. The results are shown in the figure 4.14 and the cost functional (4.3) together with the corresponding $L^1$-errors for the recovered $\rho$ and $u$ are presented in figure 4.15 for $m = 1, \ldots, 511$ iterations, though to achieve the desired accuracy, only 17 iterations were needed.

As one can see in figure 4.15 (left), the cost functional keeps decreasing monotonically while the $L^1$-errors plotted in figure 4.15 (right), on the contrary, do not experience monotone behavior in this example. As one can see, the quality of the recovered solution gets improved, while the recovered initial data do not approach (4.12). The latter is explained by nonuniqueness of the optimal control in the case of discontinuous terminal state. We would like to point out that switching the filter in Step 6 of Algorithm 3.1 on leads to no further improvement, unless the filter is applied to $\rho_0^{(m)}$ with extremely large $\ell$. This would be equivalent to searching an optimal control among the initial conditions with constant $\rho$, which is obviously not a generic case.

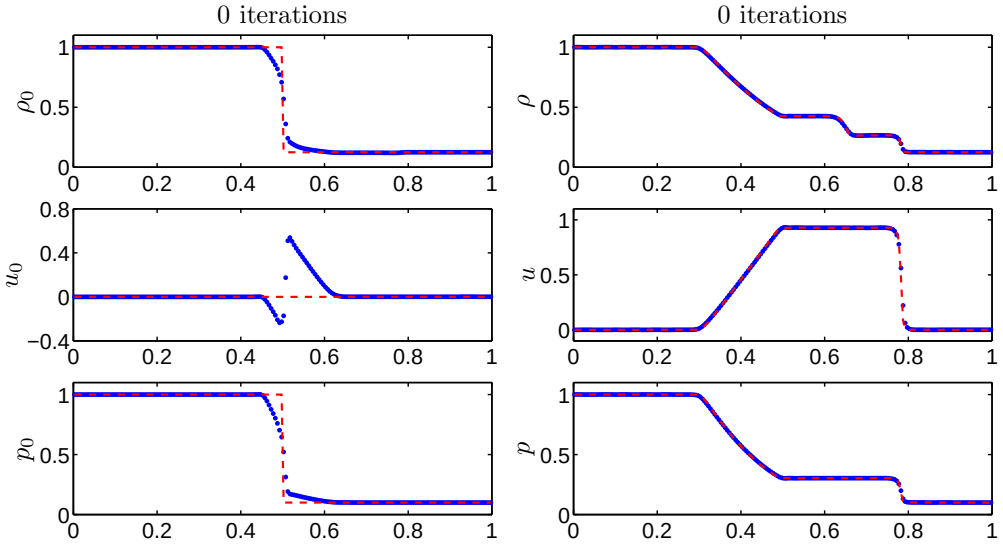


FIG. 4.16. *Example 5: Left: Initial guess* $\big(\rho_0^{(0)}(x),u_0^{(0)}(x),p_0^{(0)}(x)\big)$ *(plotted with points) and the Sod initial data (dashed line); Right: Recovered solution* $\big(\rho^{(0)}(x,0.16),u^{(0)}(x,0.16),p^{(0)}(x,0.16)\big)$ *(plotted with points) and the terminal state* $\big(\rho_d(x),u_d(x),p_d(x)\big)$ *(dashed line).*

**Example 5: $\boldsymbol{w}_d$ is a solution of the Sod shock tube problem.** In this example, the terminal state $\boldsymbol{w}_d(x)=(\rho_d(x),\mathcal{M}_d(x),E_d(x))^T$ is a nonsmooth solution of the Sod shock tube problem for the Euler equations of gas dynamics.

We solve the optimization problem (1.1), (4.3), (4.4) in the interval $[0,1]$ using a uniform grid with $\Delta x=1/200$. The terminal state $\boldsymbol{w}_d$ is prescribed at time $T=0.16$ and obtained by numerically solving the system (4.4) subject to the following Riemann initial data:

$$(\rho(x,0),u(x,0),p(x,0))=\begin{cases}(1.000,0,1.0), & \text{if } x\le 0.5,\\ (0.125,0,0.1), & \text{if } x>0.5.\end{cases} \tag{4.13}$$

The solution is computed using the second-order semi-discrete central-upwind scheme (Section 3.1) on a uniform grid with $\Delta x=1/200$. In this example, we use $\varepsilon_J=10^{-5}$ and $\varepsilon_{\Delta J}=10^{-12}$ in the stopping criteria in Step 4 of Algorithm 3.1 and solve the optimization problem on the same uniform grid.

As one can see in figure 4.16, the initial guess obtained by (3.3) (shown in figure 4.16, left) immediately leads to a very accurate recovered solution (shown in figure 4.16, right). To reach the desired tolerance, we proceed according to Algorithm 3.1.

*Example 5a:* We first implement Algorithm 3.1 with the nonlinear filter switched off (that is, with Step 6 omitted). The recovered initial data are shown in figure 4.17 (left). In figure 4.17 (right), we plot the corresponding recovered solution, and the behavior of the $L^1$-errors together with the cost functional (4.3) are shown in figure 4.18 using a logarithmic scale.

We would like to point out that in this example, the iteration process stops based on the second stopping criterion in Step 4 of Algorithm 3.1 because the size of cost

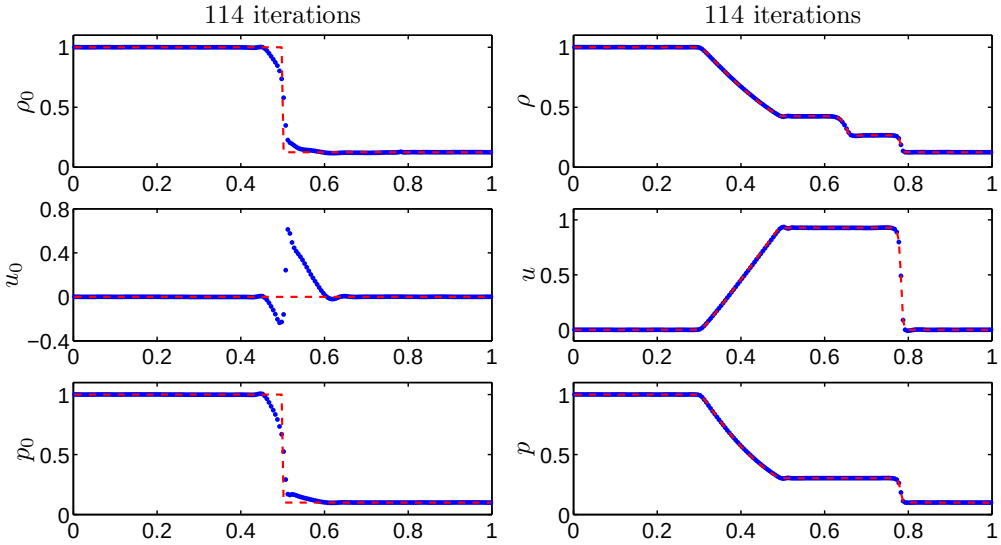


FIG. 4.17. *Example 5a: Left:* $\left(\rho_0^{(114)}(x), u_0^{(114)}(x), p_0^{(114)}(x)\right)$ *(plotted with points) and the Sod initial data (dashed line); Right: Recovered solution* $\left(\rho^{(114)}(x,0.16), u^{(114)}(x,0.16), p^{(114)}(x,0.16)\right)$ *(plotted with points) and the terminal state* $\left(\rho_d(x), u_d(x), p_d(x)\right)$ *(dashed line).*

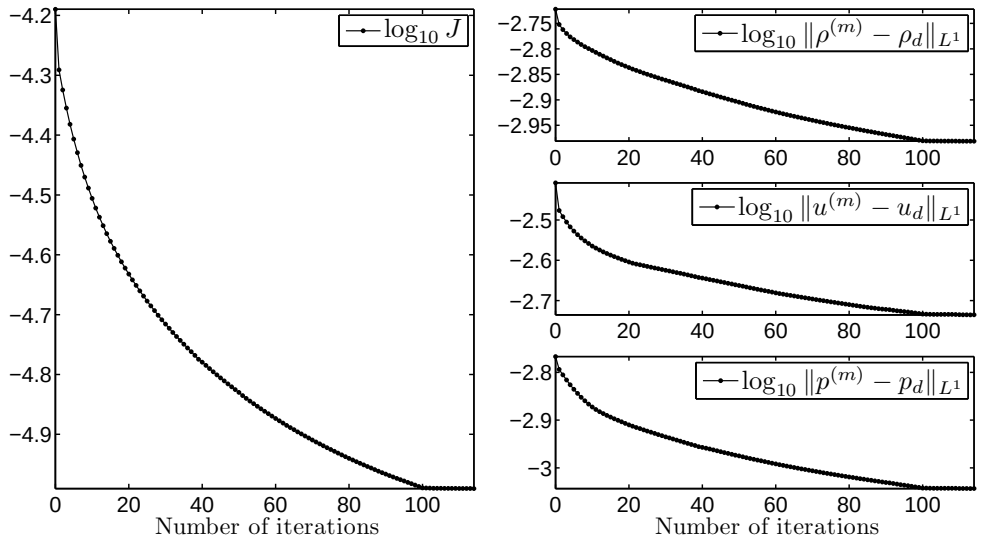


FIG. 4.18. *Example 5a: Dependence of the logarithm of the cost functional (4.3) (left) and the* $L^1$*-errors (right) on the number of iterations.*

functional $J$ stays above $\varepsilon_J = 10^{-5}$; see figure 4.18 (left). Moreover, one can see why having the second stopping criterion is important: if it is removed, the iteration process would never stop because the tolerance level $\varepsilon_J = 10^{-5}$ will never be achieved.

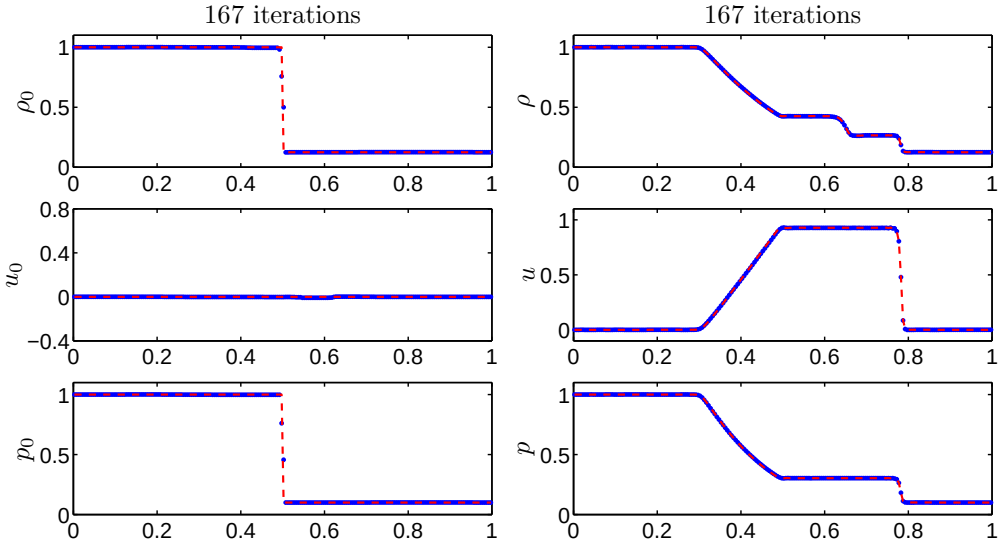


FIG. 4.19. *Example 5b: Left: Recovered initial data* $\big(\rho_0^{(167)}(x),u_0^{(167)}(x),p_0^{(167)}(x)\big)$ *(plotted with points) and the exact initial data (dashed line); Right: Recovered solution* $\big(\rho^{(167)}(x,0.16),u^{(167)}(x,0.16),p^{(167)}(x,0.16)$ *(plotted with points) and the terminal state* $\big(\rho_d(x),u_d(x),p_d(x)\big)$ *(dashed line).*

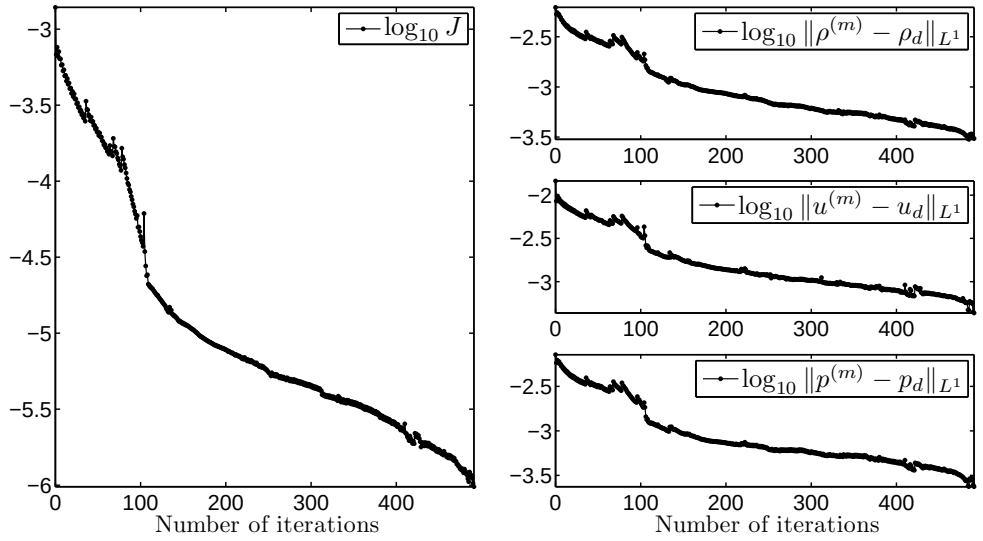


FIG. 4.20. *Example 5b: Dependence of the logarithm of the cost functional (4.3) (left) and the* $L^1$*-errors (right) on the number of iterations.*

*Example 5b:* We then repeat the previous example, but the nonlinear filter described in Section 3.3 is now switched on (that is, Step 6 is included). The filter with $\ell=10$ is applied to the variables $\rho_0^{(m)}$, $u_0^{(m)}$, and $p_0^{(m)}$ at every other iteration step starting from $m=0$.

The recovered initial data, $\left(\rho_0^{(167)}(x),u_0^{(167)}(x),p_0^{(167)}(x)\right)$, and the corresponding solution of (4.4) are shown in figure 4.19. In figure 4.20, we plot the cost functional and $L^1$-errors for $m=1,\ldots,491$ iterations. The obtained results clearly demonstrate how powerful the filtering mechanism can be. Even though the recovered solution plotted in figure 4.19 (right) is not better than its counterpart from Example 5a, the recovered initial data are almost perfect; see figure 4.19 (left). We would like to emphasize that the recovered initial data obtained in Example 5b are better than the corresponding optimization result in Example 5a just in the sense that it has a much smaller total variation: this is achieved using the nonlinear filter that affects the local extrema and thus reduced the total variation of the recovered initial data. Notice that the number of iterations has increased and that the iteration process now stops based on the first stopping criterion in Step 4 of Algorithm 3.1 (see figure 4.20 (left)), and hence the accuracy is improved. Moreover, as one can see from the plot of the cost functional, the accuracy can be further improved. However, neither the solution nor the recovered initial data will change qualitatively. Also notice that unlike in Example 5a, neither the cost functional (4.3) nor the $L^1$-errors of the recovered solutions exhibit monotone behavior when the filter is applied. On one hand, this slows down the convergence of the proposed iteration process, but on the other hand, it helps to further minimize the cost functional (4.3).

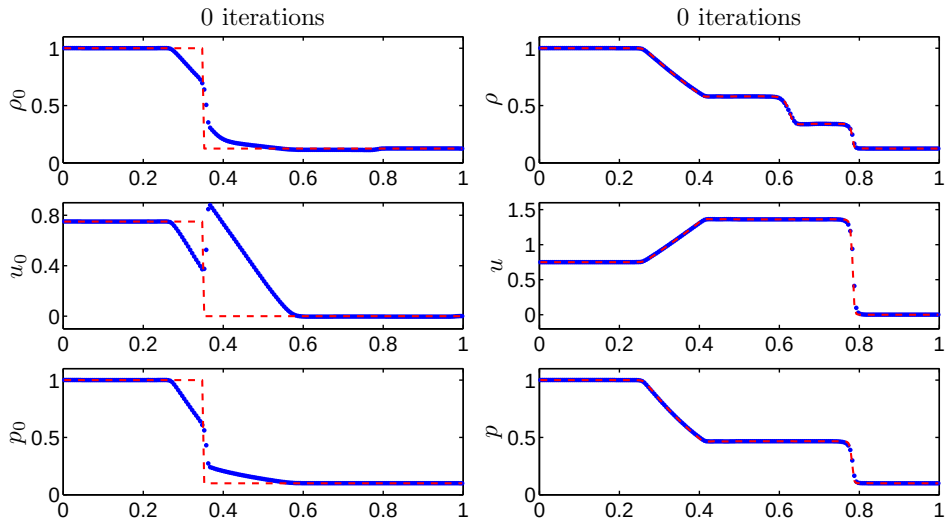


FIG. 4.21. *Example 6: Left: Initial guess* $\left(\rho_0^{(0)}(x),u_0^{(0)}(x),p_0^{(0)}(x)\right)$ *(plotted with points) and the initial data (4.14) (dashed line); Right: Recovered solution* $\left(\rho^{(0)}(x,0.2),u^{(0)}(x,0.2),p^{(0)}(x,0.2)\right)$ *(plotted with points) and the terminal state* $\left(\rho_d(x),u_d(x),p_d(x)\right)$ *(dashed line).*

**Example 6: $\boldsymbol{w}_d$ is a discontinuous solution of the Euler equations of gas dynamics.** In this example, we again consider the optimization problem (1.1), (4.3), (4.4), which is solved in the interval $[0,1]$ using a uniform grid with $\Delta x=1/200$. The terminal state $\boldsymbol{w}_d(x)$, prescribed at $T=0.2$, is a solution of (4.4) subject to the following Riemann initial data:

$$(\rho(x,0),u(x,0),p(x,0))=\begin{cases}(1.000,0.75,1.0), & \text{if } x\le 0.35,\\ (0.125,0.00,0.1), & \text{if } x>0.35.\end{cases} \tag{4.14}$$

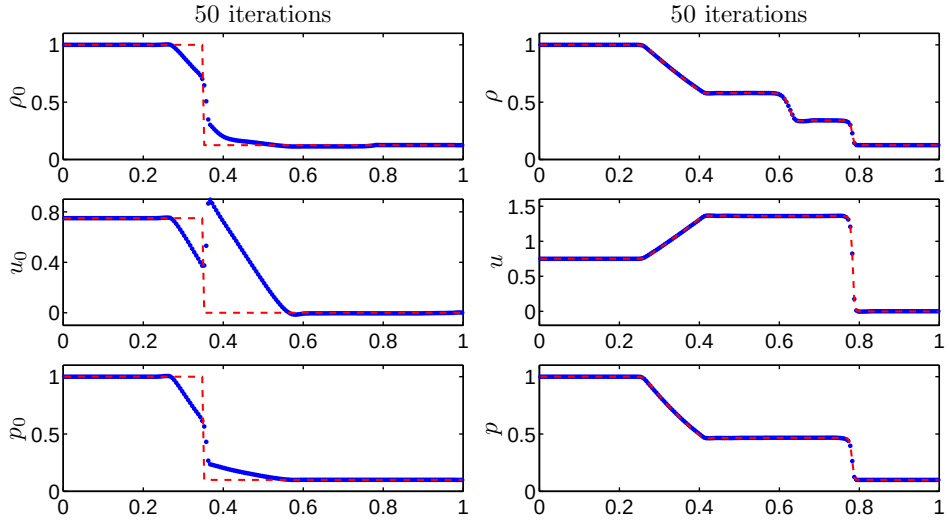


FIG. 4.22. *Example 6a: Left: Recovered initial data* $\big(\rho_0^{(50)}(x),u_0^{(50)}(x),p_0^{(50)}(x)\big)$ *(plotted with points) and the initial data (4.14) (dashed line); Right: Recovered solution* $\big(\rho^{(50)}(x,0.2),u^{(50)}(x,0.2),p^{(50)}(x,0.2)\big)$ *(plotted with points) and the terminal state* $\big(\rho_d(x),u_d(x),p_d(x)\big)$ *(dashed line).*

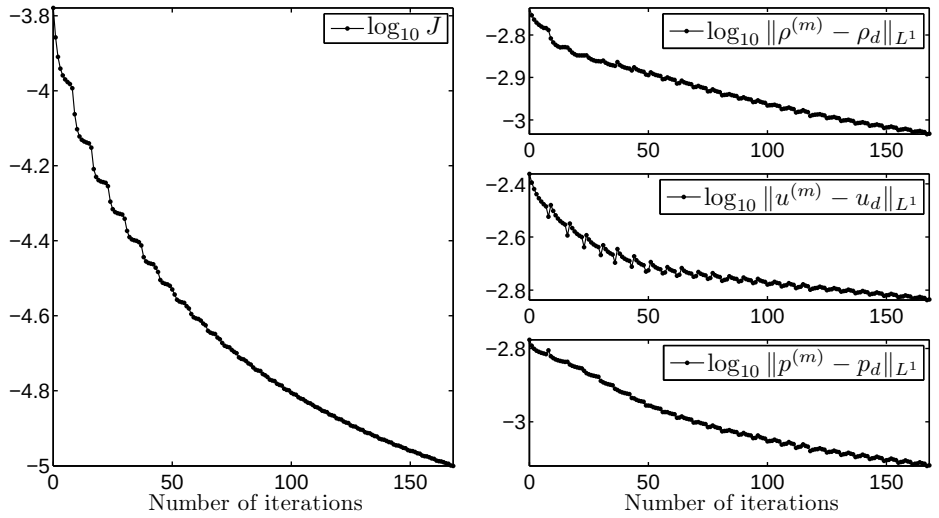


FIG. 4.23. *Example 6a: Dependence of the logarithm of the cost functional (4.3) (left) and the* $L^1$*-errors (right) on the number of iterations.*

We compute $\boldsymbol{w}_d(x)$ numerically using the second-order semi-discrete central-upwind scheme (Section 3.1) on a uniform grid with $\Delta x=1/200$. The initial guess, $\big(\rho_0^{(0)}(x),u_0^{(0)}(x),p_0^{(0)}(x)\big)$, chosen according to (3.3), and the corresponding solution, $\big(\rho^{(0)}(x,0.2),u^{(0)}(x,0.2),p^{(0)}(x,0.2)\big)$, are shown in figure 4.21. As one can see, this initial guess leads to a quite accurate recovered solution (see figure 4.21, right).

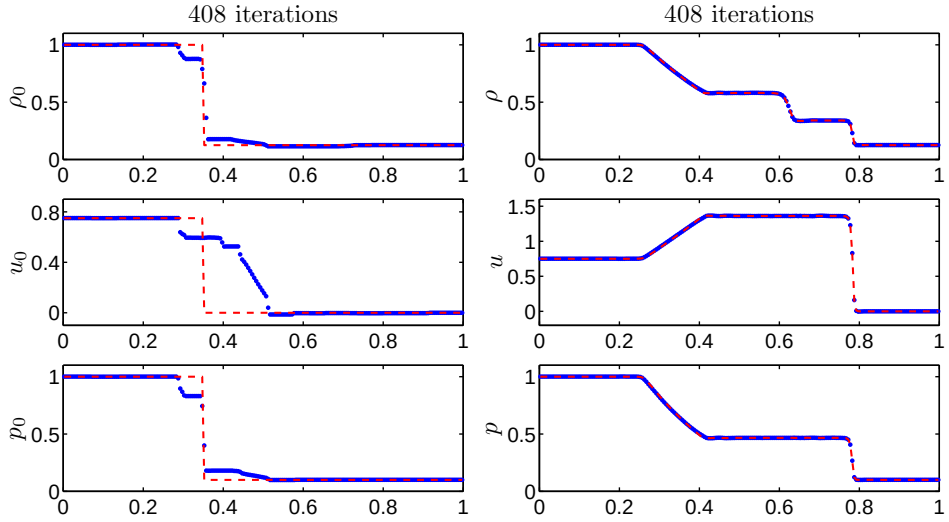


FIG. 4.24. *Example 6b: Left: Recovered initial data* $\left(\rho_0^{(408)}(x),u_0^{(408)}(x),p_0^{(408)}(x)\right)$ *(plotted with points) and the initial data (4.14) (dashed line); Right: Recovered solution* $\left(\rho^{(408)}(x,0.2),u^{(408)}(x,0.2),p^{(408)}(x,0.2\right)$ *(plotted with points) and the terminal state* $\left(\rho_d(x),u_d(x),p_d(x)\right)$ *(dashed line).*

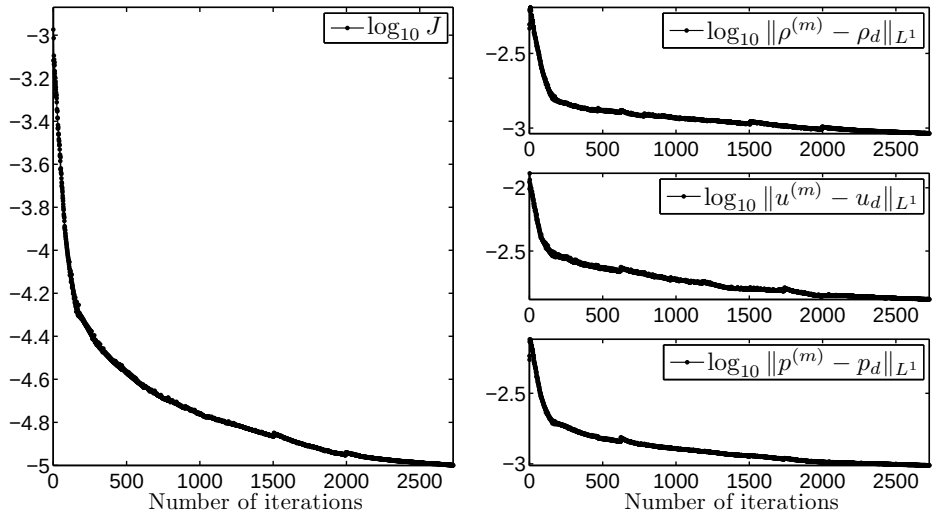


FIG. 4.25. *Example 6b: Dependence of the logarithm of the cost functional (4.3) (left) and the* $L^1$*-errors (right) on the number of iterations.*

In order to further improve the result, we proceed according to the optimization Algorithm 3.1 with $\varepsilon_J = 3\cdot 10^{-5}$ and $\varepsilon_{\Delta J} = 10^{-12}$ in the stopping criteria in Step 4 of the algorithm. In this example, we solve the backward problem (3.15) using the semi-discrete scheme (3.21), (3.22). Although, this scheme is more computationally expensive for a single iteration than its simplified analog, (3.23), (3.22) with $P_{j+\frac{1}{2}} \equiv I_3$,

it leads to a faster convergence in terms of number of iterations because the gradient direction is found more accurately. In addition, the use of a more accurate scheme (3.21), (3.22) allows one to achieve smaller values of the cost functional on a given computational grid as is clearly confirmed by our numerical experiments, which are not reported here for the sake of brevity.

*Example 6a:* First, Algorithm 3.1 is implemented with the filter being switched off, that is, Step 6 is not included. The recovered initial data $\big(\rho_0^{(50)}(x),u_0^{(50)}(x),p_0^{(50)}(x)\big)$ and the corresponding recovered solution are shown in figure 4.22. In figure 4.23, we show the behavior of the cost functional (4.3) and the $L^1$-errors for $m=1,\dots,168$ iterations using a logarithmic scale. As one can see, in this case, the use of Algorithm 3.1 without the nonlinear filter does not lead to any qualitative change in either the recovered initial data or recovered solution in comparison to $\boldsymbol{w}_0^0(x)$ or $\boldsymbol{w}^0(x,0.2)$, respectively.

*Example 6b:* We now switch the nonlinear filter in Step 6 of Algorithm 3.1 on. The filter is implemented with $\ell=10$ and it is applied to the variables $\rho_0^{(m)}$, $u_0^{(m)}$, and $p_0^{(m)}$ at every second iteration step starting from $m=0$.

The recovered initial data $\big(\rho_0^{(408)}(x),u_0^{(408)}(x),p_0^{(408)}(x)\big)$ are shown in figure 4.24 (left) and the corresponding recovered solution is plotted in figure 4.24 (right). As one can see, the solution is still accurately recovered (see figure 4.24, right) and is very similar to the one obtained in Example 6a without the filter (see figure 4.22, right). However, the computed optimal controls are very different—the recovered initial data in Example 6b are nearly monotone and thus have much smaller total variation than the recovered initial data obtained in Example 6a (compare the left graphs in figures 4.24 and 4.22).

In figure 4.25, we plot the cost functional (4.3) and the $L^1$-errors for $m=1,\cdots,2728$ iterations using a logarithmic scale. As in the previous examples, the use of the filter leads to a nonmonotone behavior of these quantities and thus to somewhat slower convergence.

**Example 7: $\boldsymbol{w}_d$ is a perturbed solution of the Sod shock tube problem.** Here, we again consider the optimization problem (1.1), (4.3), (4.4) and solve it in the interval $[-1,2]$ using a uniform grid with $\Delta x=1/200$. To obtain the terminal state $\boldsymbol{w}_d(x)=(\rho_d(x),\mathcal{M}_d(x),E_d(x))^T$ prescribed at time $T=0.16$, we numerically solve the system (4.4) subject to the Sod initial data (4.13) on $[-1,2]$ using the second-order semi-discrete central-upwind scheme (Section 3.1) on a uniform grid with $\Delta x=1/200$, and then add independent random variables $\delta\rho_j$, $\delta u_j$, and $\delta p_j$ to the density, velocity, and pressure fields of the computed solution of the Sod shock tube problem, respectively. The noise,

$$\delta\rho_j,\delta u_j,\delta p_j\propto\mathcal{N}(0,0.04),$$

where $\mathcal{N}(\mu,\sigma)$ states for the normal distribution with mean $\mu$ and variance $\sigma^2$, is added on the interval $x\in[0,1]$ only.

In this example, we minimize a smoothed version (defined in (3.4)) of the functional (4.3):

$$\begin{aligned}J_\delta(\boldsymbol{w}(\cdot,T);\boldsymbol{w}_d(\cdot))=\frac{1}{2}\int\limits_{\mathbb{R}}\Big|\big((\rho*\varphi_\delta)(x,T),(\mathcal{M}*\varphi_\delta)(x,T),(E*\varphi_\delta)(x,T)\big)^T\\ -\big((\rho_d*\varphi_\delta)(x),(\mathcal{M}_d*\varphi_\delta))(x),(E_d*\varphi_\delta))(x)\big)^T\Big|^2dx,\end{aligned}\tag{4.15}$$

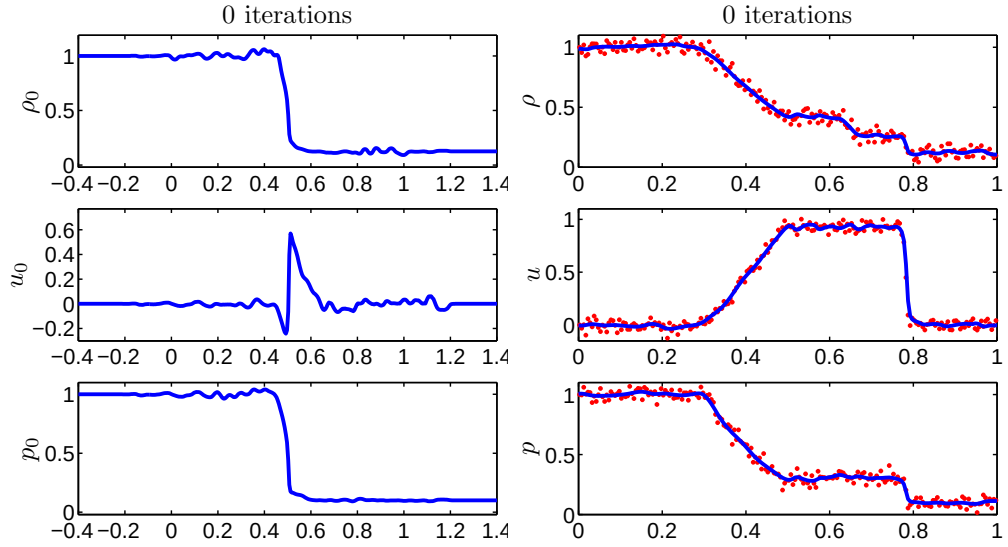


FIG. 4.26. *Example 7: Left: Initial guess* $\left(\rho_0^{(0)}(x),u_0^{(0)}(x),p_0^{(0)}(x)\right)$*; Right: Recovered solution* $\left(\rho^{(0)}(x,0.16),u^{(0)}(x,0.16),p^{(0)}(x,0.16)\right)$ *(solid line) and the terminal state* $\left(\rho_d(x),u_d(x),p_d(x)\right)$ *(plotted with points).*

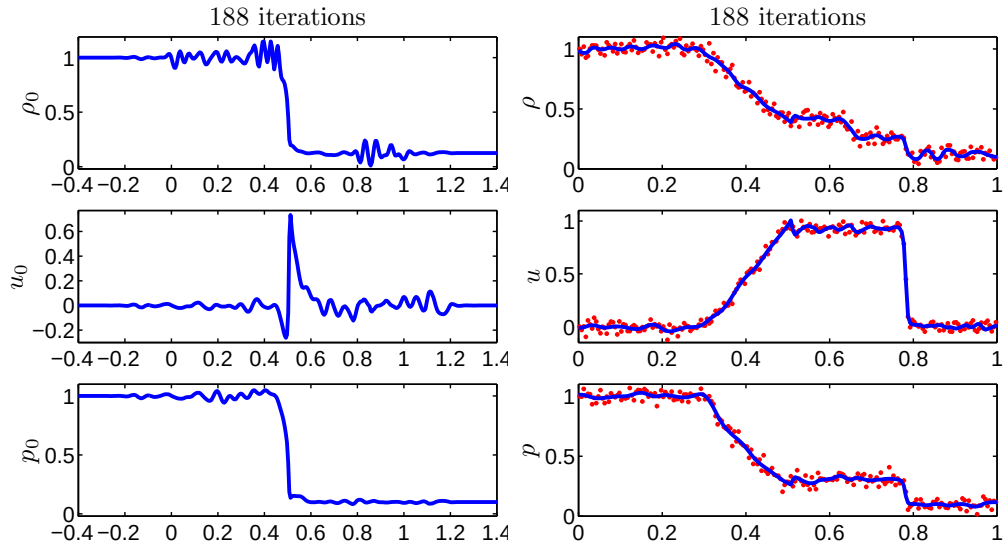


FIG. 4.27. *Example 7: Left: Recovered initial data* $\left(\rho_0^{(188)}(x),u_0^{(188)}(x),p_0^{(188)}(x)\right)$*; Right: Recovered solution* $\left(\rho^{(188)}(x,0.16),u^{(188)}(x,0.16),p^{(188)}(x,0.16)\right.$ *(solid line) and the terminal state* $\left(\rho_d(x),u_d(x),p_d(x)\right)$ *(plotted with points).*

where we take $\delta = 0.02$. We use Algorithm 3.1 with $J$ replaced with $J_\delta$ and the filter in Step 6 switched off. We use $\varepsilon_J = 3\cdot 10^{-5}$ and $\varepsilon_{\Delta J} = 10^{-12}$ in Step 4 of Algorithm 3.1.

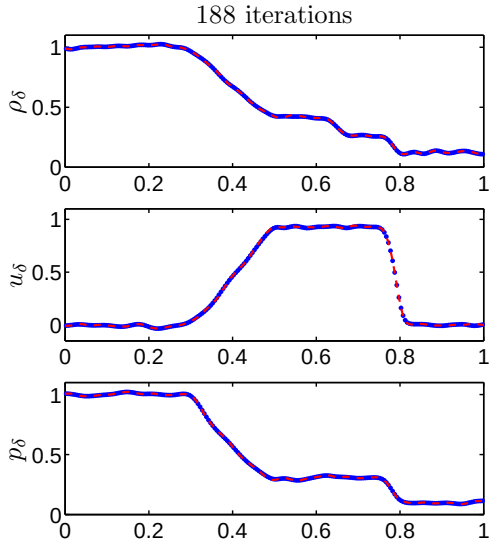


FIG. 4.28. *Example 7: Smoothed recovered solution* $\big(\rho_\delta^{(188)}(x,0.16),u_\delta^{(188)}(x,0.16),p_\delta^{(188)}(x,0.16)\big)$ *(plotted with points) and smoothed terminal state* $\big(\rho_{d,\delta}(x),u_{d,\delta}(x),p_{d,\delta}(x)\big)$ *(dashed line).*

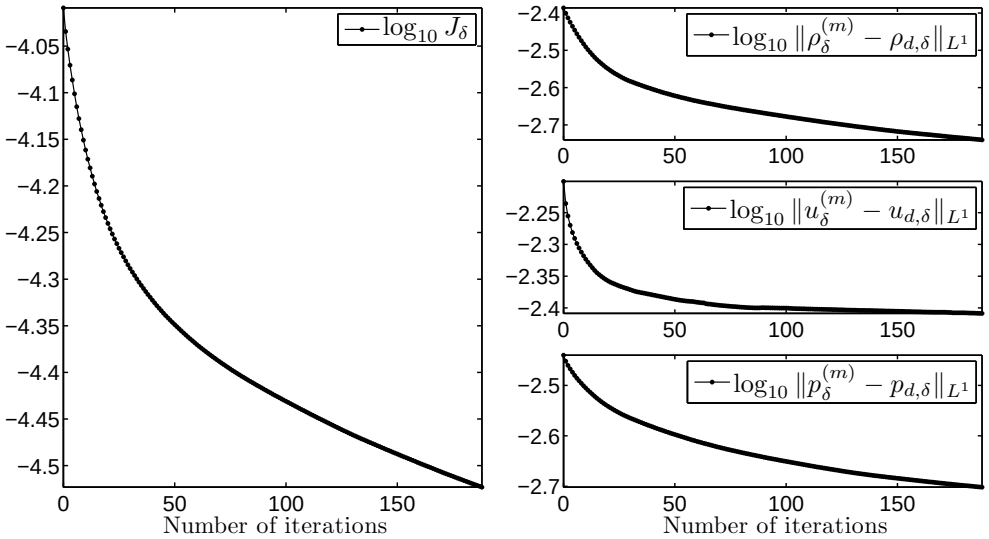


FIG. 4.29. *Example 7: Dependence of the logarithm of the cost functional (4.15) (left) and the smoothed* $L^1$*-errors (right) on the number of iterations.*

Due to the randomness in the terminal data, the solution of this optimization problem is not expected to be fully recovered for any initial guess. The initial guess, $\big(\rho_0^{(0)}(x),u_0^{(0)}(x),p_0^{(0)}(x)\big)$, chosen according to (3.3), and the corresponding solution,

$\big(\rho^{(0)}(x,0.16),u^{(0)}(x,0.16),p^{(0)}(x,0.16)\big)$, are shown in figure 4.26. As one can see, this initial guess results as usual in a quite good recovered solution $\boldsymbol{w}^{(0)}$ (figure 4.26, right) which is, however, not nearly as oscillatory as $\boldsymbol{w}_d$ is.

We implement Algorithm 3.1 and achieve the desired tolerance after 188 iterations. The obtained optimization results are shown in figure 4.27. As one can see, both the recovered initial data, $\big(\rho_0^{(188)}(x),u_0^{(188)}(x),p_0^{(188)}(x)\big)$, plotted on the left, and the corresponding solution, $\big(\rho^{(188)}(x,0.16),u^{(188)}(x,0.16),p^{(188)}(x,0.16)$, plotted on the right, are much more oscillatory than the initial approximation shown in figure 4.26, but the recovered solution is still not nearly as oscillatory as the terminal state is. On the other hand, the smoothed recovered solution is very close to the smoothed terminal state, as it is demonstrated in figure 4.28. This is also confirmed in figure 4.29, where we show the behavior of the cost functional (4.15) and the smoothed $L^1$-errors using a logarithmic scale.

## 5. Conclusions

In this paper, we have proposed a numerical method (described in Section 3) for solving the optimal control problem (1.1), (1.2) with the cost functional given by either (1.3) or its smoothed version (3.4). The method has been tested on a number of numerical examples presented in Section 4. The obtained results demonstrate high accuracy of the developed algorithm both in smooth and nonsmooth cases.

In the case of sufficiently smooth solutions, the proposed Algorithm 3.1 without filtering (that is, with Step 6 omitted) corresponds to a block Gauß-Seidel iteration applied to the system of necessary first-order optimality conditions (1.1)–(1.3), (2.2)–(2.4). The adjoint equation (2.2) is a system of nonconservative hyperbolic equations with variable, possibly discontinuous coefficients. In Section 3.2, we have developed a new upwind scheme for (2.2), (2.3). The new scheme is an extension of the upwind scheme proposed in [21] for scalar transport equations with variable coefficients.

While a proof of convergence of the proposed optimization method for systems of PDEs seems to be out of reach at this stage, the convergence in the scalar case will be studied in the future work. A rigorous proof of convergence including nonsmooth solutions usually relies on sufficiently smooth cost functionals. This is typically achieved by a convolution (see Remark 3.2). Applying the filtering in Step 6 of the proposed algorithm serves a similar purpose: It reduces small oscillations in both the reconstructed control and the solution.

**Acknowledgment.** The work of M. Herty was supported by the DAAD 55866082, BMBF KinOpt, and the cluster of excellence EXC128. The work of A. Kurganov was supported in part by the NSF Grant DMS-1115718. The authors also acknowledge the support by the KI-Net research network, NSF RNMS grant DMS-1107444.

## REFERENCES

[1] M. Banda and M. Herty, *Adjoint IMEX-based schemes for control problems governed by hyperbolic conservation laws*, Comput. Optim. Appl., 51(2), 909–930, 2012.

[2] S. Bianchini, *On the shift differentiability of the flow generated by a hyperbolic system of conservation laws*, Disc. Cont. Dyn. Sys., 6, 329–350, 2000.

[3] A. Bressan and G. Guerra, *Shift-differentiability of the flow generated by a conservation law*, Disc. Cont. Dyn. Sys., 3, 35–58, 1997.

[4] A. Bressan and M. Lewicka, *Shift differentials of maps in BV spaces*, in Nonlinear Theory of Generalized Functions (Vienna, 1997), Chapman & Hall/CRC Res. Notes Math., Chapman & Hall/CRC, Boca Raton, FL, 401, 47–61, 1999.

[5] A. Bressan and A. Marson, *A variational calculus for discontinuous solutions to conservation laws*, Commun. Part. Diff. Eqs., 20, 1491–1552, 1995.
[6] A. Bressan and W. Shen, *Optimality conditions for solutions to hyperbolic balance laws*, Control methods in PDE-dynamical systems, Contemp. Math., 426, 129–152, 2007.
[7] P. Calamai and J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Program., 39, 93–116, 1987.
[8] C. Castro, F. Palacios, and E. Zuazua, *An alternating descent method for the optimal control of the inviscid Burgers equation in the presence of shocks*, Math. Models Meth. Appl. Sci., 18, 369–416, 2008.
[9] A. Chertock, M. Herty, and A. Kurganov, *An Eulerian-Lagrangian method for optimization problems governed by nonlinear hyperbolic PDEs*, Comput. Optim. Appl., accepted, 2014.
[10] C. Dafermos, *Hyperbolic Conservation Laws in Continuum Physics*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, Third Edition, 325, 2010.
[11] B. Engquist, P. Lötstedt, and B. Sjögreen, *Nonlinear filters for efficient shock computation*, Math. Comput., 52, 509–537, 1989.
[12] M. Giles, *Analysis of the accuracy of shock–capturing in the steady quasi 1d–Euler equations*, Int. J. Comput. Fluid Dynam., 5, 247–258, 1996.
[13] M. Giles, *Discrete adjoint approximations with shocks*, in Hyperbolic Problems: Theory, Numerics, Applications, Springer, Berlin, 185–194, 2003.
[14] M. Giles and N. Pierce, *Analytic adjoint solutions for the quasi-one-dimensional Euler equations*, J. Fluid Mech., 426, 327–345, 2001.
[15] M. Giles and N. Pierce, *Adjoint error correction for integral outputs*, in Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics, Lect. Notes Comput. Sci. Eng., Springer, Berlin, 25, 47–95, 2003.
[16] M. Giles and E. Süli, *Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality*, Acta Numer., 11, 145–236, 2002.
[17] M. Giles and S. Ulbrich, *Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. Part 1: Linearized approximations and linearized output functionals*, SIAM J. Numer. Anal., 48, 882–904, 2010.
[18] M. Giles and S. Ulbrich, *Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. Part 2: Adjoint approximations and extensions*, SIAM J. Numer. Anal., 48, 905–921, 2010.
[19] S. Gottlieb, D. Ketcheson, and C.-W. Shu, *Strong stability preserving Runge-Kutta and multistep time discretizations*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2011.
[20] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43, 89–112, 2001.
[21] F. James and M. Sepúlveda, *Convergence results for the flux identification in a scalar conservation law*, SIAM J. Control Optim., 37, 869–891, (electronic) 1999.
[22] C. Kelley, *Iterative Methods for Optimization*, Frontiers in Applied Mathematics, Philadelphia, PA: Society for Industrial and Applied Mathematics. xv, 1999.
[23] A. Kurganov and C.T. Lin, *On the reduction of numerical dissipation in central-upwind schemes*, Commun. Comput. Phys., 2, 141–163, 2007.
[24] A. Kurganov, S. Noelle, and G. Petrova, *Semi-discrete central-upwind scheme for hyperbolic conservation laws and Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 23, 707–740, 2001.
[25] A. Kurganov and E. Tadmor, *New high resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys., 160, 241–282, 2000.
[26] P. LeFloch, *Hyperbolic Systems of Conservation Laws*, Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 2002.
[27] K.A. Lie and S. Noelle, *On the artificial compression method for second-order nonoscillatory central difference schemes for systems of conservation laws*, SIAM J. Sci. Comput., 24, 1157–1174, 2003.
[28] H. Nessayahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87, 408–463, 1990.
[29] D. Serre, *Systems of Conservation Laws. 1*, Cambridge University Press, Cambridge, 1999.
[30] P. Spellucci, *Numerical Methods of Nonlinear Optimization (Numerische Verfahren der Nichtlinearen Optimierung)*, ISNM Lehrbuch. Basel: Birkhäuser Verlag. 576 S., 1993.
[31] P.K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal., 21, 995–1011, 1984.
[32] S. Ulbrich, *Optimal Control of Nonlinear Hyperbolic Conservation Laws with Source Terms*, Habilitation Thesis, Technische Universitaet Muenchen, 2001.

[33] S. Ulbrich, *Adjoint-based derivative computations for the optimal control of discontinuous solutions of hyperbolic conservation laws*, Syst. Control Lett., 48, 313–328, 2003.

[34] S. Ulbrich, *On the superlinear local convergence of a filer-SQP method*, Math. Prog. Ser. B, 100, 217–245, 2004.

[35] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, J. Comput. Phys., 32, 101–136, 1979.