Constraint Splitting and Projection Methods for Optimal Control

Variational Analysis and Optimization Webinar
Mathematics of Computation and Optimization (MoCaO)
Adelaide, Australia

30 September 2020

C. Yalçın Kaya
University of South Australia

Joint work with

Heinz H. Bauschke
University of British Columbia

Regina S. Burachik
University of South Australia

Early part based on

- H. H. Bauschke, R. S. Burachik, C. Y. Kaya, Constraint splitting and projection methods for optimal control of double integrator. Chapter in Springer Edited Book "Splitting Algorithms, Modern Operator Theory, and Applications," pp. 45–68, 2019. (arXiv:1804.03767, 2018).

Outline

1 Cubic curves in $\mathbb{R}^n$: *Minimum-energy control of the double integrator*

2 Constraint splitting and projections

3 Best approx. algorithms: *Dykstra* | *MAP* | *DR* | *AAC* | *FISTA*

4 Numerical experiments: parametric behaviour | error analysis

Motivation

- Douglas–Rachford splitting method applied to *discrete-time* optimal control problems.
  (O'Donoghue–Stathopoulos–Boyd 2013)
  Also see (Eckstein–Ferris 1998).

- No known example of application of best approximation algorithms to *continuous-time* optimal control problems – except Bauschke–Burachik–K (2019).

- Minimum-energy control of the double integrator – building block for cubic splines.

Cubic Curves in $\mathbb{R}^n$

$$
(\mathrm{P}) \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 \|u(t)\|_2^2\, dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\ & \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,. \end{cases}
$$

$x_1(t), x_2(t) \in \mathbb{R}^n$: state variable vectors

$u(t) \in \mathbb{R}^n$: control variable vector

$x_i(t) = (x_{i,1}(t), \ldots, x_{i,n}(t)),\ i = 1, 2; \quad u(t) = (u_1(t), \ldots, u_n(t))$

$n = 1$: Min.-energy control of the the double integrator

$n = 2, 3$: Spatial curves with minimum ave. acceleration

Cubic Curves in $\mathbb{R}^n$

$$
(\mathrm{P}) \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 \|u(t)\|_2^2\, dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\ & \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,, \\ & \|u(t)\|_2 \le a\,. \quad \text{(constrained acceleration)} \end{cases}
$$

$x_1(t), x_2(t) \in \mathbb{R}^n$: state variable vectors

$u(t) \in \mathbb{R}^n$: control variable vector

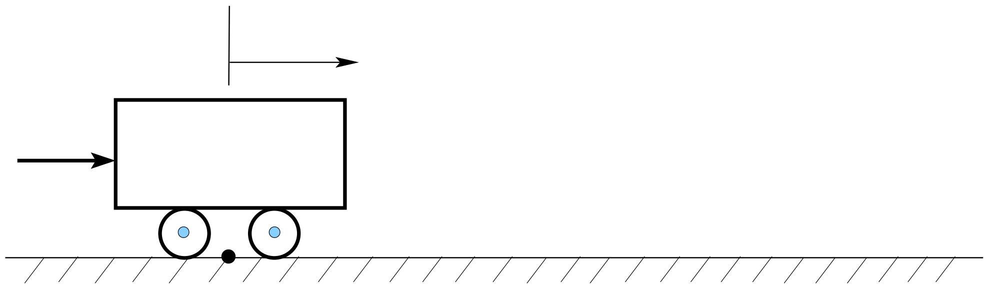
$x_i(t) = (x_{i,1}(t), \ldots, x_{i,n}(t)),\ i = 1, 2; \quad u(t) = (u_1(t), \ldots, u_n(t))$

$n = 1$: Min.-energy control of the the double integrator

$n = 2, 3$: Spatial curves with minimum ave. acceleration

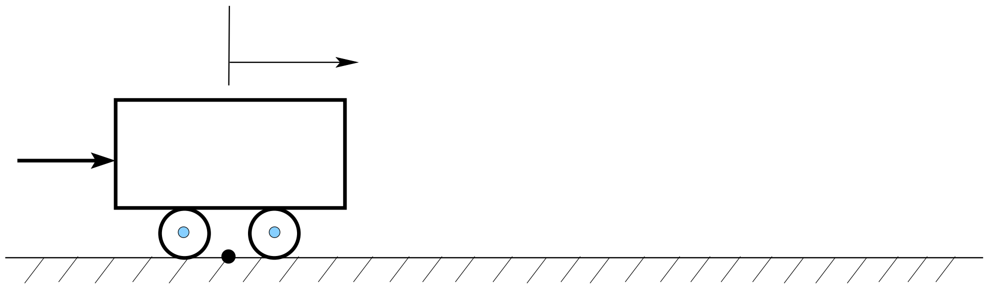
Examples with $n = 1, 2, 3$

$n = 1:$

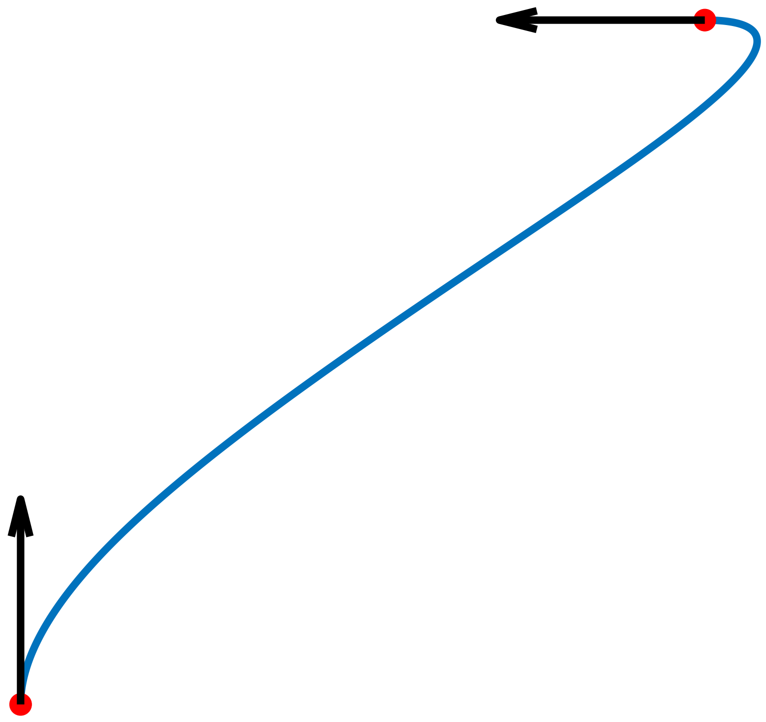


Examples with $n = 1, 2, 3$

$n = 1:$

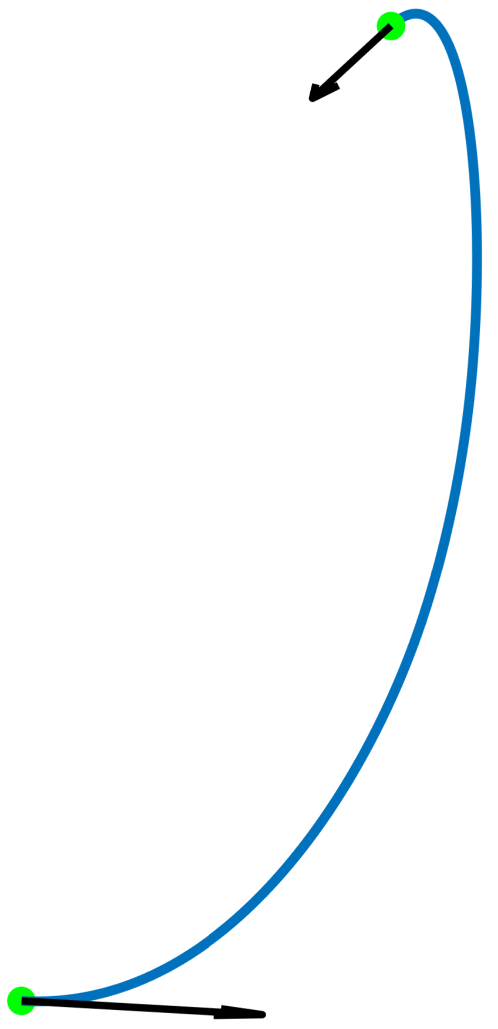


$n = 2:$

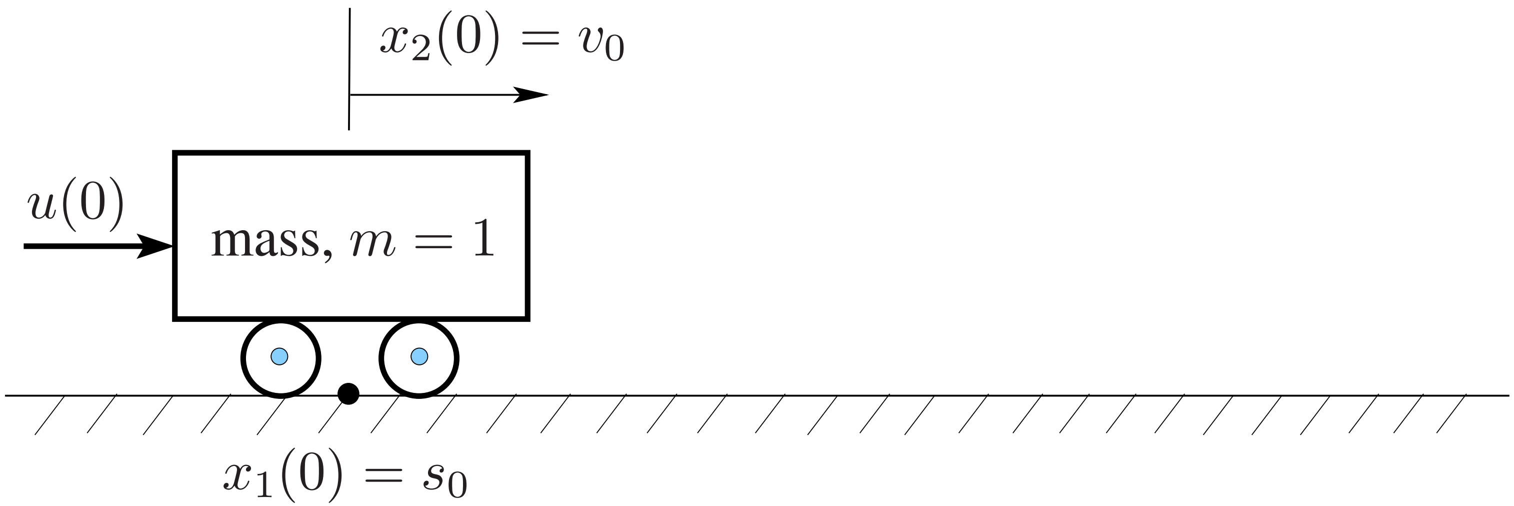


Examples with $n = 1, 2, 3$

$n = 3$:



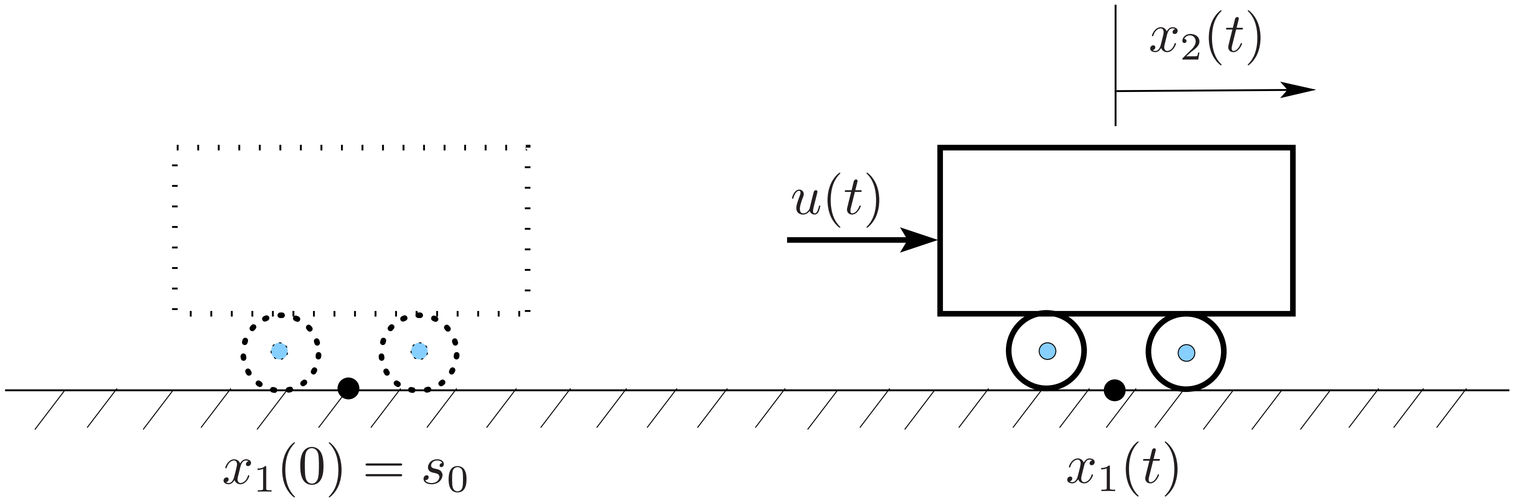
Min.-energy Control of Double Integrator



$$
\text{(P0)} \quad \begin{cases} \min & \displaystyle\frac{1}{2}\int_0^1 u^2(t)\,dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\ & \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,. \end{cases}
$$

$x_1(t)$, $x_2(t)$: state variables; $u(t)$: control variable.

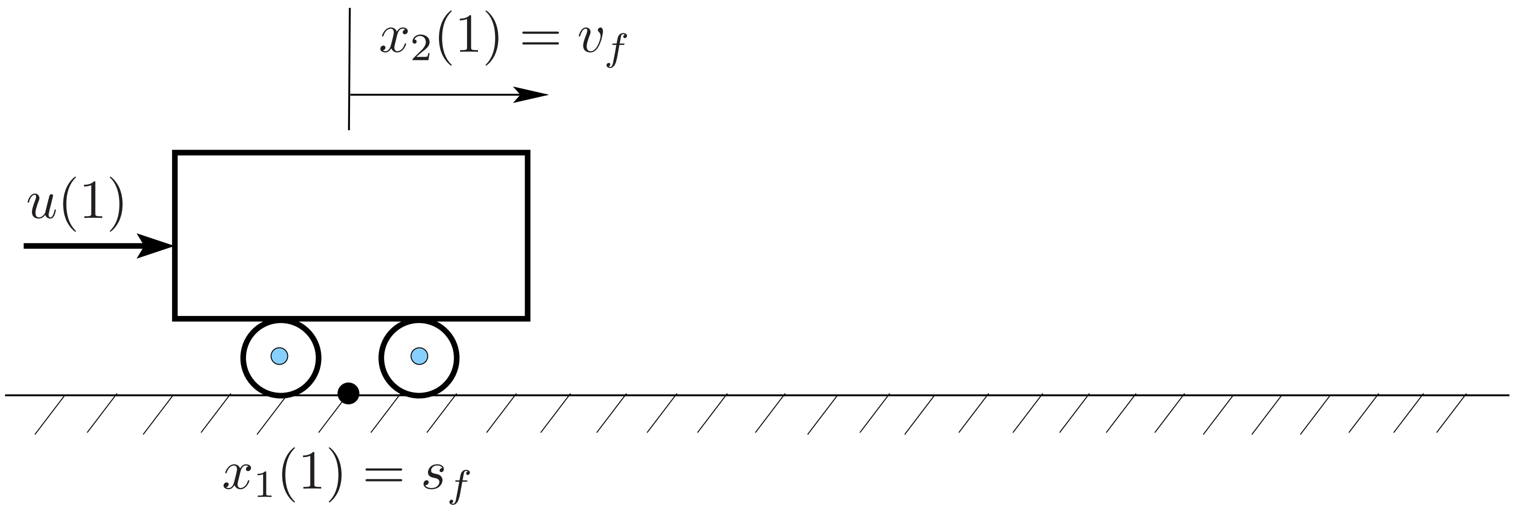
Min.-energy Control of Double Integrator



$$
\text{(P0)} \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 u^2(t)\,dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\ & \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,. \end{cases}
$$

$x_1(t)$, $x_2(t)$: state variables; $u(t)$: control variable.

Min.-energy Control of Double Integrator



$$
(\mathrm{P0}) \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 u^2(t)\,dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\ & \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,. \end{cases}
$$

$x_1(t)$, $x_2(t)$: state variables; $u(t)$: control variable.

Min.-energy Control of Double Integrator

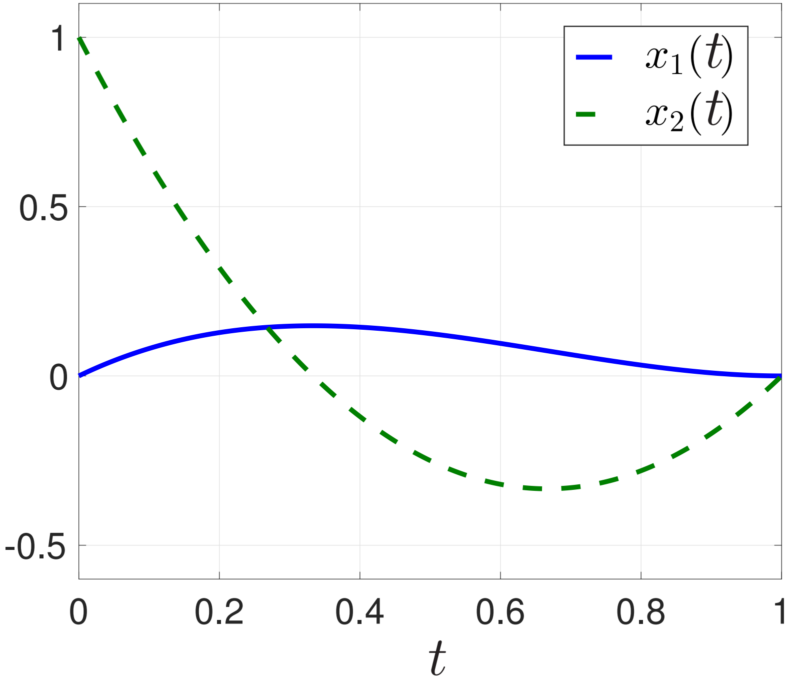
$$\begin{aligned} u(t) &= c_1\, t + c_2\,, \\ x_1(t) &= \frac{1}{6}\, c_1\, t^3 + \frac{1}{2}\, c_2\, t^2 + v_0\, t + s_0\,, \\ x_2(t) &= \frac{1}{2}\, c_1\, t^2 + c_2\, t + v_0\,, \end{aligned}$$

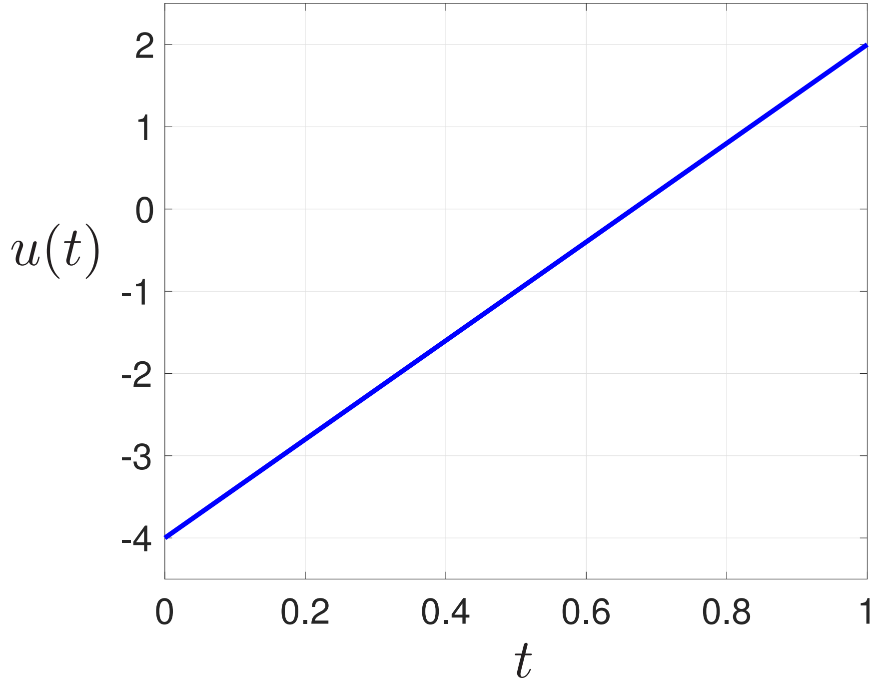
for all $t \in [0, 1]$, where

$$\begin{aligned} c_1 &= -12\,(s_f - s_0) + 6\,(v_0 + v_f)\,, \\ c_2 &= 6\,(s_f - s_0) - 2\,(2\,v_0 + v_f)\,. \end{aligned}$$

Min.-energy Control of Double Integrator

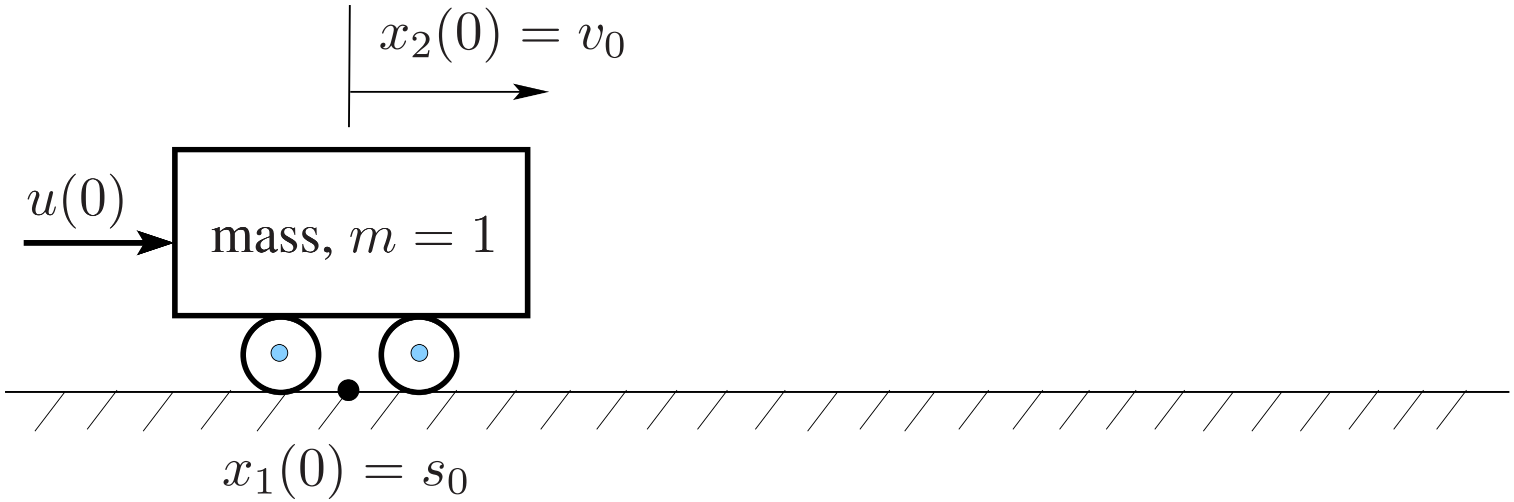
Solution with $s_0 = 0$, $s_f = 0$, $v_0 = 1$, $v_f = 0$:





$$
\begin{aligned}
u(t) &= 6\,t - 4; \\
x_1(t) &= t^3 - 2\,t^2 + t \\
x_2(t) &= 3\,t^2 - 4\,t + 1
\end{aligned}
$$

Min.-energy Control with Constraints



$$
(\mathrm{P}) \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 u^2(t)\,dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\ & \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,, \\ & -a \le u(t) \le a\,. \end{cases}
$$

$a > 0$ some real constant.

Min.-energy Control with Constraints

Maximum Principle

Define the *Hamiltonian function* :

$$H(x_1, x_2, u, \lambda_1, \lambda_2) := \frac{1}{2}u^2 + \lambda_1 x_2 + \lambda_2 u\,,$$

where the *adjoint variables* $\lambda_1(t)$ and $\lambda_2(t)$ satisfy

$$\dot{\lambda}_1 = -\partial H/\partial x_1 \quad \text{and} \quad \dot{\lambda}_2 = -\partial H/\partial x_2\,,$$

i.e.,

$$\lambda_1(t) = c_1 \quad \text{and} \quad \lambda_2(t) = -c_1\,t - c_2\,,$$

c_1, c_2 real constants.

Optimality Conditions (DI)

Maximum Principle

If u is an optimal control for Problem (P), then there exists a continuously differentiable vector of adjoint variables λ, as defined before, such that $\lambda(t) \neq 0$ for all $t \in [0, t_f]$, and that, for a.e. $t \in [0, t_f]$,

$$u(t) = \operatorname*{argmin}_{v \in [-a,a]} H(x, v, \lambda(t)) \,,$$

i.e.,

$$u(t) = \operatorname*{argmin}_{v \in [-a,a]} \frac{1}{2} v^2 + \lambda_2(t)\, v \,.$$

Optimality Conditions (DI)

Optimal control

$$u(t) = \begin{cases} -\lambda_2(t)\,, & \text{if } -a \le \lambda_2(t) \le a\,, \\ a\,, & \text{if } \lambda_2(t) \le -a\,, \\ -a\,, & \text{if } \lambda_2(t) \ge a\,. \end{cases}$$

Note that the optimal control u for Problem (P) is continuous.

Numerical Solution Techniques

Three approaches:

I. (*First-*)*discretize-then-optimize*

1. Discretize Problem (P) over a partition of the time horizon $[0, 1]$.
2. Use some (large-scale) finite-dimensional optimization software (e.g. AMPL + Ipopt) to get a *discrete* (finite-dimensional) *approximation* for the state and control variables $x(t)$ and $u(t)$.

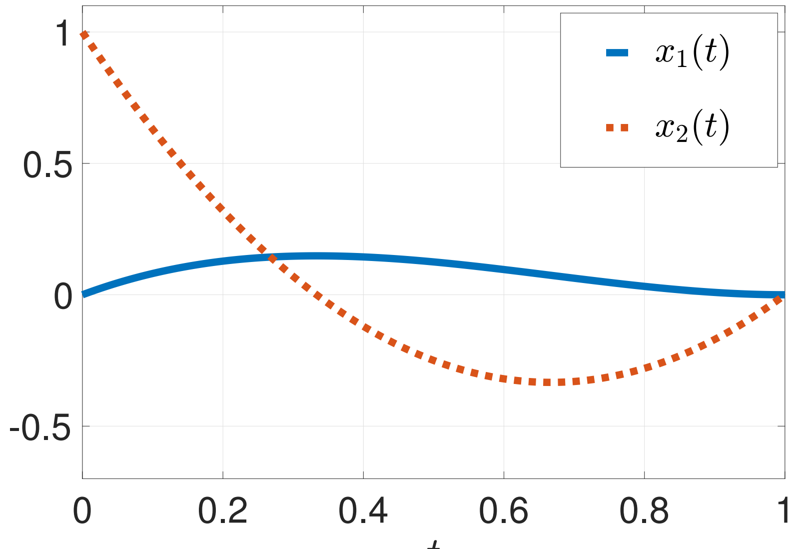
II. (*First-*)*optimize-then-discretize*

1. Write down conditions of optimality.
2. Solve the optimality conditions by using discretized functions.

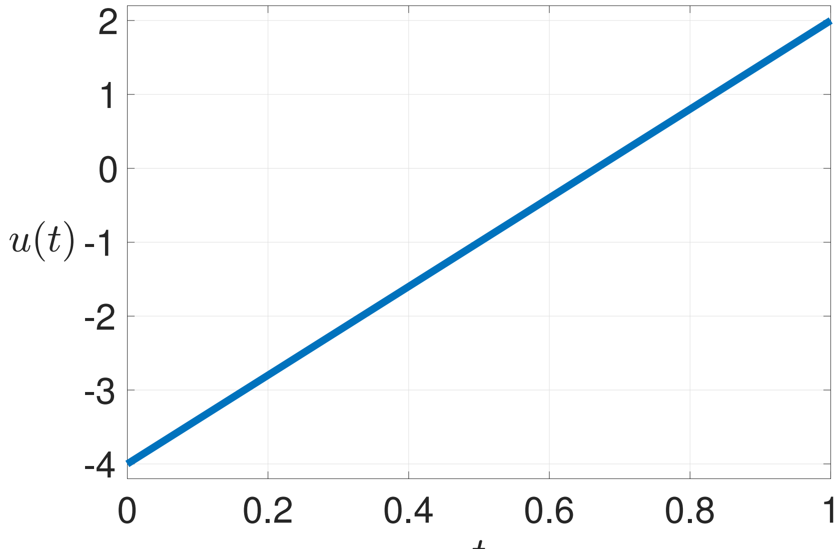
III. *Arc parameterization*

1. Parameterize w.r.t. a concatenation of $(u(t) = a)$-, $(u(t) = -a)$- and $(u(t) = -\lambda_2(t))$-arcs over intervals $[t_{i-1}, t_i]$, t_i unknown, $i = 1, \ldots, N$.
2. Use some finite-dimensional optimization software (e.g. AMPL + Ipopt) to find the unknown t_i, $i = 1, \ldots, N$.

Analytical Solution (DI)



(a) Optimal state variables

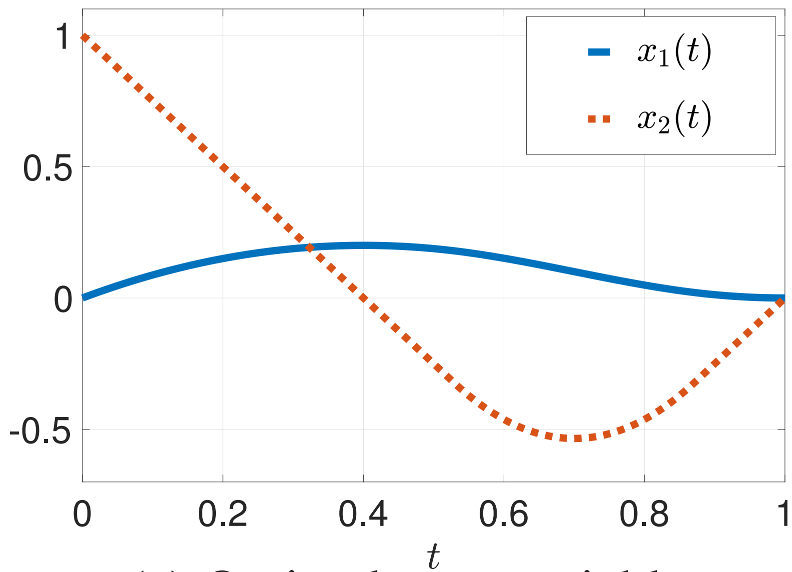


(b) Optimal control variable

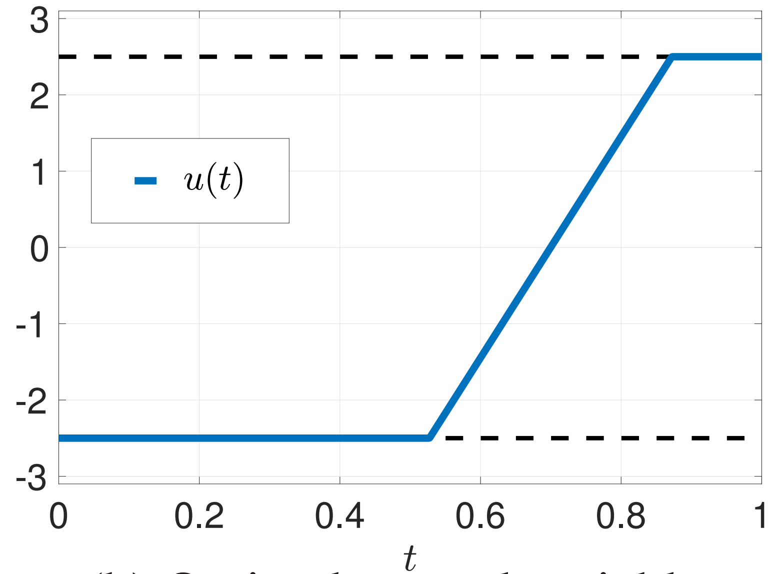
$s_0 = 0$, $s_f = 0$, $v_0 = 1$, $v_f = 0$

$a = \infty$

Numerical Solution (DI)

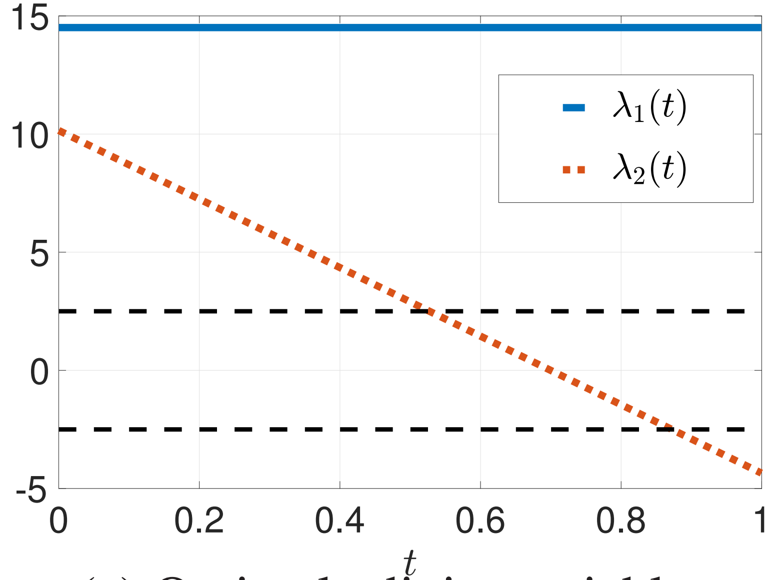


(a) Optimal state variables



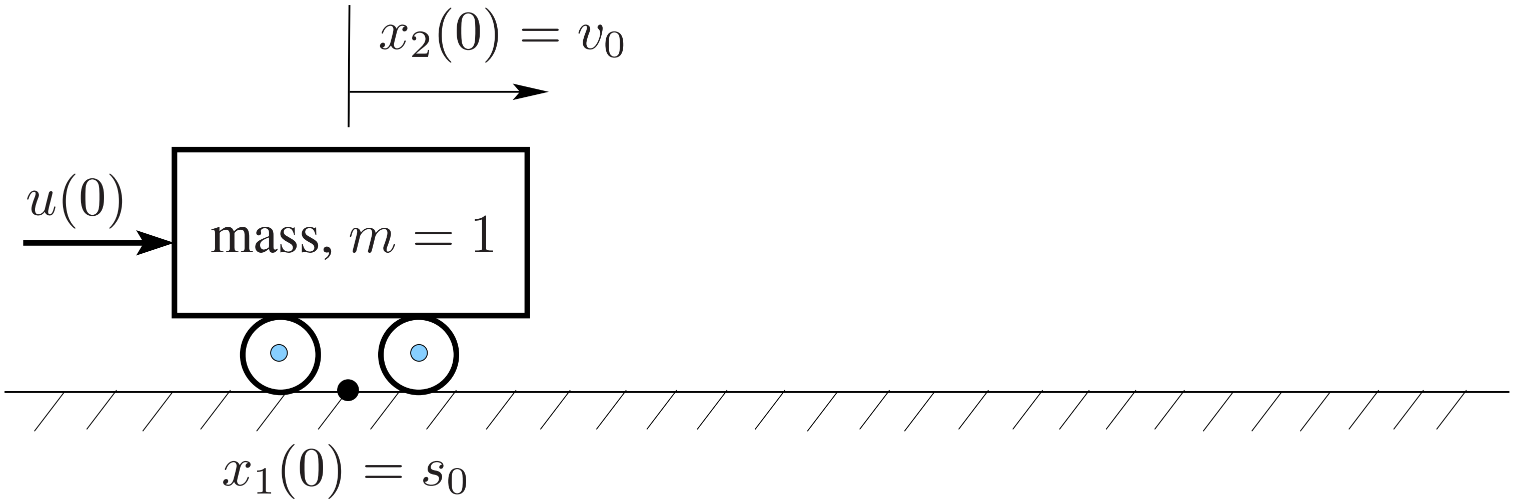
(b) Optimal control variable

$$s_0 = 0,\, s_f = 0,\, v_0 = 1,\, v_f = 0$$
$$a = 2.5$$



(c) Optimal adjoint variables

Constraint Splitting (DI)



$$
(\mathrm{P}) \left\{ \begin{array}{ll}
\min & \displaystyle \frac{1}{2}\int_0^1 u^2(t)\,dt \\
\text{subject to} & \dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,, \quad x_1(1) = s_f\,, \\
& \dot{x}_2(t) = u(t)\,, \quad x_2(0) = v_0\,, \quad x_2(1) = v_f\,, \\
& -a \le u(t) \le a\,.
\end{array} \right.
$$

Constraint Splitting (DI in $\mathbb{R}^n$)

$$
\text{(Pc)}\quad \begin{cases}
\quad\text{min} & \displaystyle\frac{1}{2}\int_0^1 \|u(t)\|_2^2\, dt \\
\text{subject to} & \dot{x}_1(t) = x_2(t)\,,\ \ x_1(0) = s_0\,,\ \ x_1(1) = s_f\,, \\
 & \dot{x}_2(t) = u(t)\,,\ \ \ x_2(0) = v_0\,,\ \ x_2(1) = v_f\,, \\
 & \|u(t)\|_2 \le a\,.
\end{cases}
$$

Constraint Splitting

$$\mathcal{A} := \{u \in L^2(0,1;\mathbb{R}^n) \mid \exists x_i \in W^{1,2}(0,1;\mathbb{R}^n), i = 1,2, \text{ which solve}$$

$$\dot{x}_1(t) = x_2(t)\,,\; x_1(0) = s_0\,,\; x_1(1) = s_f\,,$$

$$\dot{x}_2(t) = u(t)\,,\;\; x_2(0) = v_0\,,\; x_2(1) = v_f\,,$$

$$\forall t \in [0,1]\}\,,$$

$$\mathcal{B} := \{u \in L^2(0,1;\mathbb{R}^n) \mid \|u(t)\|_2 \le a\,, \text{ for all } t \in [0,1]\}\,.$$

$\mathcal{A}$ is an ***affine subspace*** and $\mathcal{B}$ a ***ball***.

Projections

The projection onto $\mathcal{A}$ from a current iterate u^- is u which solves

$$\text{(P1)} \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 \|u(t) - u^-(t)\|_2^2 \, dt \\ \text{subject to} & u \in \mathcal{A}\,. \end{cases}$$

The projection onto $\mathcal{B}$ from a current iterate u^- is u which solves

$$\text{(P2)} \begin{cases} \min & \dfrac{1}{2}\displaystyle\int_0^1 \|u(t) - u^-(t)\|_2^2 \, dt \\ \text{subject to} & u \in \mathcal{B}\,. \end{cases}$$

Projections

Proposition 1 (Projection onto $\boldsymbol{\mathcal{A}}$). *The projection $P_{\mathcal{A}}$ of $u^- \in L^2(0,1;\mathbb{R}^n)$ onto the constraint set $\mathcal{A}$, as the solution of Problem* (P1)*, is given by*

$$P_{\mathcal{A}}(u^-)(t) = u^-(t) + c_1\, t + c_2\,,$$

for all $t \in [0,1]$, where

$$c_1 = 12\,(x_1(1) - s_f) - 6\,(x_2(1) - v_f)\,,$$

$$c_2 = -6\,(x_1(1) - s_f) + 2\,(x_2(1) - v_f)\,,$$

and $x_1(1)$ and $x_2(1)$ are obtained by solving the IVP

$$\dot{x}_1(t) = x_2(t)\,, \quad x_1(0) = s_0\,,$$

$$\dot{x}_2(t) = u^-(t)\,, \quad x_2(0) = v_0\,,$$

for all $t \in [0,1]$.

Projections

Proposition 2 (Projection onto $\mathcal{B}$)**.** *The projection* $P_{\mathcal{B}}$ *of* $u^- \in L^2(0,1;\mathbb{R}^n)$ *onto the constraint set* $\mathcal{B}$*, as the solution of Problem* (P2)*, is given by*

$$P_{\mathcal{B}}(u^-)(t) = \begin{cases} u^-(t)\,, & \text{if } \|u^-(t)\|_2 \leq a\,, \\ a\,\dfrac{u^-(t)}{\|u^-(t)\|_2}\,, & \text{if } \|u^-(t)\|_2 > a\,, \end{cases}$$

for all $t \in [0,1]$.

Best Approximation Algorithms

X is a real Hilbert space

with inner product $\langle \cdot, \cdot \rangle$, induced norm $\| \cdot \|$.

A is a closed affine subspace of X, and B is a nonempty closed convex subset of X.

Given $z \in X$, our aim is to find

$$P_{A \cap B}(z),$$

the projection of z onto the intersection $A \cap B \neq \emptyset$.

Best Approximation Algorithms

We test five methods when $X = L^2(0, 1; \mathbb{R}^n)$, $A = \mathcal{A}$, $B = \mathcal{B}$, and $z = 0$:

- Dykstra's Algorithm [strongly convergent]
  (Boyle–Dykstra 1985)
- Method of Alternating Projections (MAP) [weakly convergent]
  (von Neumann 1948, Bregman 1965)
- Douglas–Rachford (DR) Algorithm [weakly convergent]
  (Douglas–Rachford 1956, Lions–Mercier 1979, Eckstein–Bertsekas 1992)
- Aragón Artacho–Campoy (AAC) Algorithm [strongly convergent]
  (Aragón Artacho–Campoy 2018, Alwadani–Bauschke–Moursi–Wang 2018)
- Fast Iterative Shrinkage-thresholding Algorithm (FISTA) [strong. conv.]
  (Beck–Teboulle 2009, Attouch–Cabot 2018, Bauschke–Bui–Wang 2019)

Best Approximation Algorithms

Algorithm 1 (Dykstra)

Step 1 (*Initialization*) Choose the initial iterates $u^0 = 0$ and $q^0 = 0$. Choose a small parameter $\varepsilon > 0$, and set $k = 0$.

Step 2 (*Projection onto* $\mathcal{B}$) Set $u^- = u^k + q^k$. Compute $\widetilde{u} = P_{\mathcal{B}}(u^-)$.

Step 3 (*Projection onto* $\mathcal{A}$) Set $u^- := \widetilde{u}$. Compute $\widehat{u} = P_{\mathcal{A}}(u^-)$.

Step 4 (*Update*) Set $u^{k+1} := \widehat{u}$ and $q^{k+1} := u^k + q^k - \widetilde{u}$.

Step 5 (*Stopping criterion*) If $\|u^{k+1} - u^k\|_{L^\infty} \leq \varepsilon$, then return $\widetilde{u}$ and stop. Otherwise, set $k := k + 1$ and go to Step 2.

Best Approximation Algorithms

Algorithm 2 (MAP)

Step 1 (*Initialization*) Choose the initial iterate $u^0 = 0$.
Choose a small parameter $\varepsilon > 0$, and set $k = 0$.

Step 2 (*Projection onto $\mathcal{B}$*) Set $u^- = u^k$. Compute $\widetilde{u} = P_{\mathcal{B}}(u^-)$.

Step 3 (*Projection onto $\mathcal{A}$*) Set $u^- := \widetilde{u}$. Compute $\widehat{u} = P_{\mathcal{A}}(u^-)$.

Step 4 (*Update*) Set $u^{k+1} := \widehat{u}$.

Step 5 (*Stopping criterion*) If $\|u^{k+1} - u^k\|_{L^\infty} \leq \varepsilon$, then return $\widetilde{u}$ and stop.
Otherwise, set $k := k + 1$ and go to Step 2.

Best Approximation Algorithms

Algorithm 3 (DR)

Step 1 (*Initialization*) Choose a parameter $\lambda \in \,]0,1[$ and the initial iterate u^0 arbitrarily. Choose a small parameter $\varepsilon > 0$, and set $k = 0$.

Step 2 (*Projection onto $\mathcal{B}$*) Set $u^- = \lambda u^k$. Compute $\widetilde{u} = P_{\mathcal{B}}(u^-)$.

Step 3 (*Projection onto $\mathcal{A}$*) Set $u^- := 2\widetilde{u} - u^k$. Compute $\widehat{u} = P_{\mathcal{A}}(u^-)$.

Step 4 (*Update*) Set $u^{k+1} := u^k + \widehat{u} - \widetilde{u}$.

Step 5 (*Stopping criterion*) If $\|u^{k+1} - u^k\|_{L^\infty} \leq \varepsilon$, then return $\widetilde{u}$ and stop. Otherwise, set $k := k + 1$ and go to Step 2.

Best Approximation Algorithms

Algorithm 4 (AAC)

Step 1 (*Initialization*) Choose the initial iterate u^0 arbitrarily. Choose a small parameter $\varepsilon > 0$, two parameters[1] α and β in $]0, 1[$, and set $k = 0$.

Step 2 (*Projection onto $\mathcal{B}$*) Set $u^- = u^k$. Compute $\widetilde{u} = P_{\mathcal{B}}(u^-)$.

Step 3 (*Projection onto $\mathcal{A}$*) Set $u^- = 2\beta\widetilde{u} - u^k$. Compute $\widehat{u} = P_{\mathcal{A}}(u^-)$.

Step 4 (*Update*) Set $u^{k+1} := u^k + 2\alpha\beta(\widehat{u} - \widetilde{u})$.

Step 5 (*Stopping criterion*) If $\|u^{k+1} - u^k\|_{L^\infty} \leq \varepsilon$, then return $\widetilde{u}$ and stop. Otherwise, set $k := k + 1$ and go to Step 2.

---

[1] Aragón Artacho and Campoy recommend $\alpha = 0.9$ and $\beta \in [0.7, 0.8]$ in their paper .

Best Approximation Algorithms

Algorithm 5 (FISTA)

Step 1 (*Initialization*) Choose $\widehat{u}_1 = \widehat{u}_2 = v = 0$, $t_0 = 1$. Choose a small parameter $\varepsilon > 0$, Lipschitz const. $L = 2$ for ℓ_2-norm, and $k = 0$.

Step 2 (*Projection onto* $\mathcal{B}$) Set $u^- := v - L\,\widehat{u}_1$. Compute $\widetilde{u}_1^{k+1} = \widehat{u}_1 - (v - P_{\mathcal{B}}(u^-))/L$.

Step 3 (*Projection onto* $\mathcal{A}$) Set $u^- := v - L\,\widehat{u}_2$. Compute $\widetilde{u}_2^{k+1} = \widehat{u}_2 - (v - P_{\mathcal{A}}(u^-))/L$.

Step 4 (*Update*) Set $u^{k+1} = \widetilde{u}_1^{k+1} + \widetilde{u}_2^{k+1}$.

Step 5 (*Stopping criterion*) If $\|u^{k+1} - u^k\|_{L^\infty} \leq \varepsilon$, then return $\widetilde{u}$ and stop. Otherwise,

set: $t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$, $\widehat{u}_i := \widetilde{u}_i^{k+1} + \frac{t_k - 1}{t_{k+1}}\left(\widetilde{u}_i^{k+1} - \widetilde{u}_i^k\right)$, $i = 1, 2$ (Nesterov 1983)

OR $\widehat{u}_i := \widetilde{u}_i^{k+1} + \frac{1-\alpha}{k+1}\left(\widetilde{u}_i^{k+1} - \widetilde{u}_i^k\right)$, $i = 1, 2$, $\alpha > 3$ (Attouch–Cabot 2018)

OR $\widehat{u}_i := \widetilde{u}_i^{k+1} + \frac{1 - \ln^\theta(k+1)}{k+1}\left(\widetilde{u}_i^{k+1} - \widetilde{u}_i^k\right)$, $i = 1, 2$, $\theta > 0$ (Attouch–Cabot 2018)

Set $v := \widehat{u}_1 + \widehat{u}_2$, $k := k + 1$ and go to Step 2.

Numerical Experiments

- Algorithms 1–5 carry out iterations with functions.

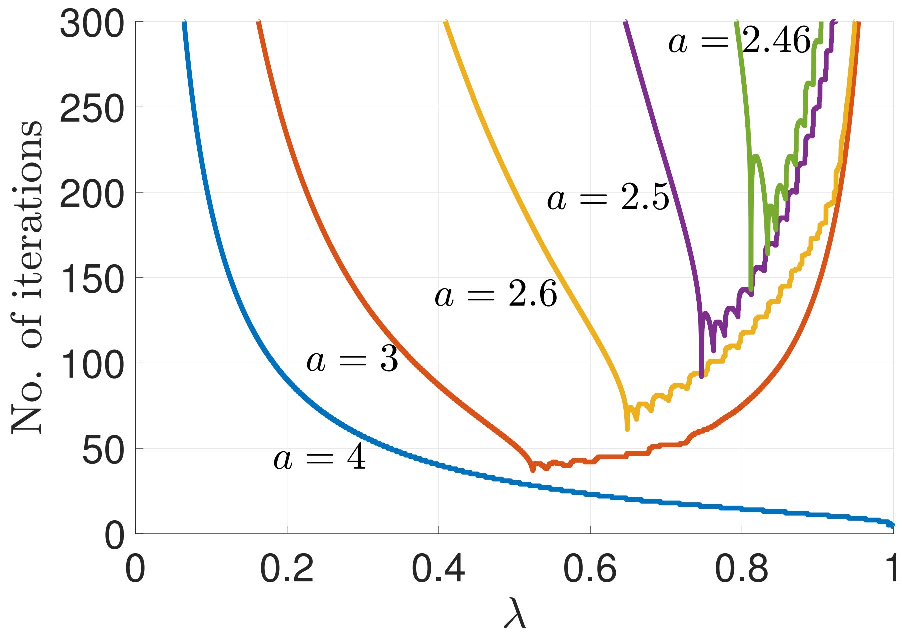
- Use discrete approximations of the functions over the partition $0 = t_0 < t_1 < \ldots < t_N = 1$. For the IVP in computing $P_{\mathcal{A}}$, use Euler's method over the same partition. (Could use any other ODE solver—interested only in $x_i(1)$)

- Define

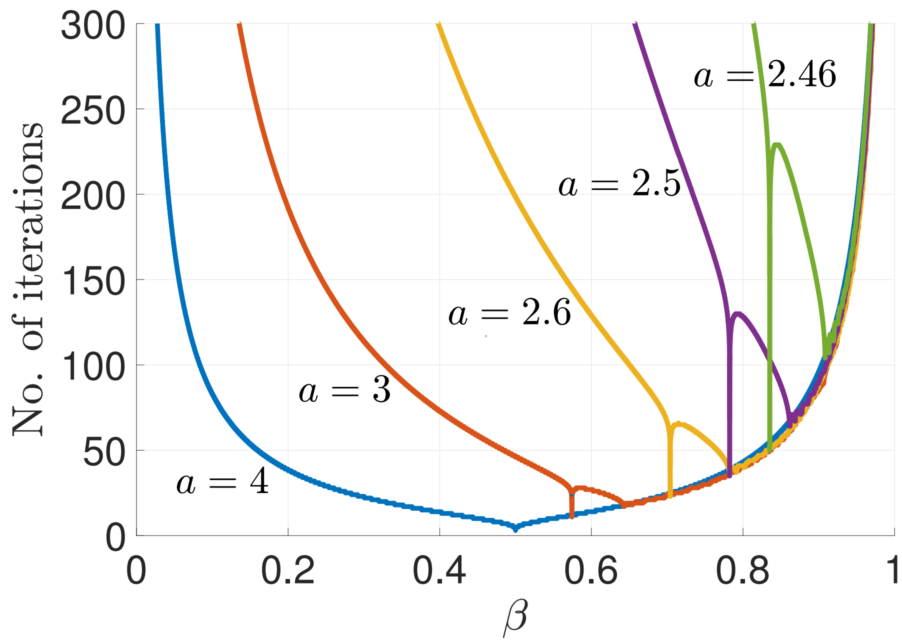
$$\sigma_u^k := \max_{0 \le i \le N-1} |u_i^k - u^*(t_i)| \quad \text{and} \quad \sigma_x^k := \max_{0 \le i \le N} ||x_i^k - x^*(t_i)||_\infty \,.$$

Numerical Experiments ($n = 1$)

Parametric Behaviour



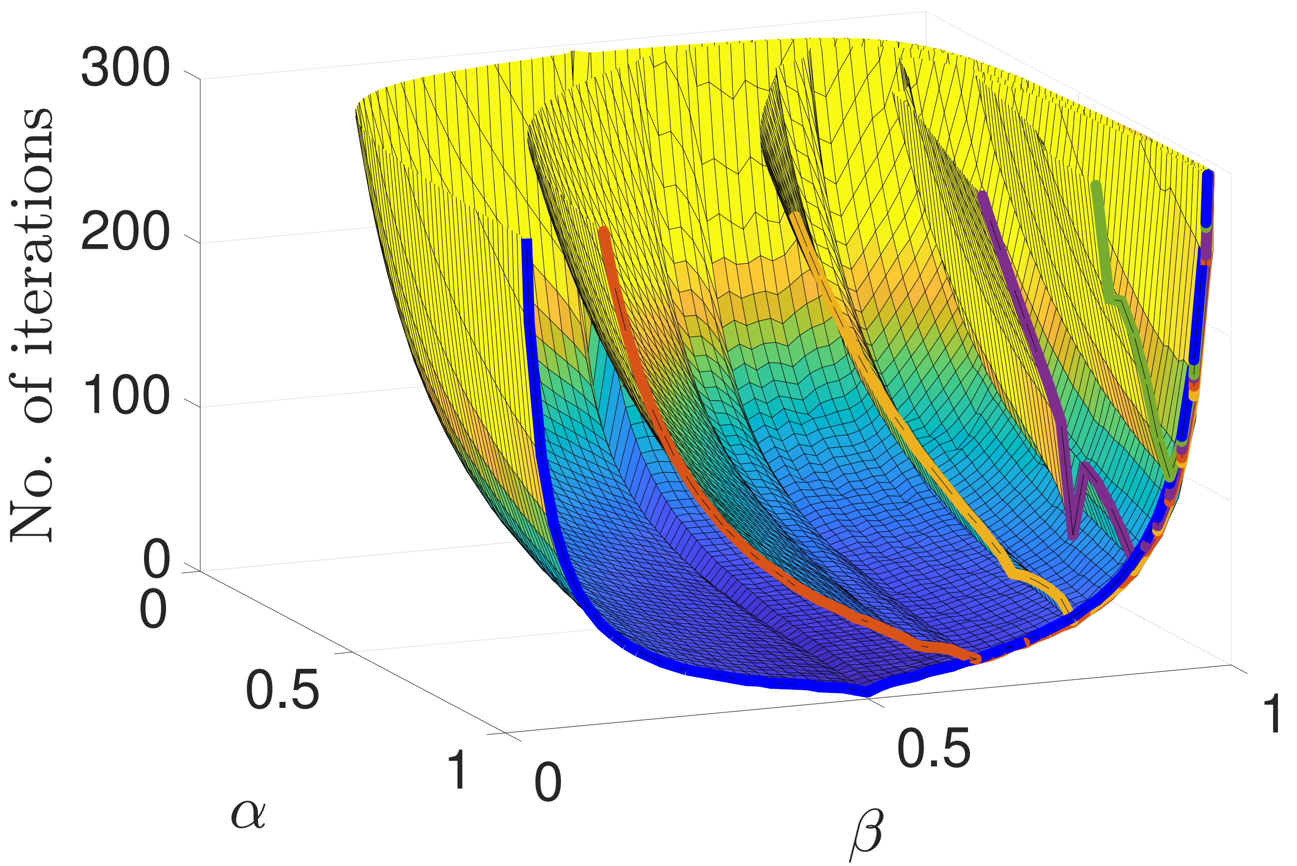
(a) Algorithm 3 (DR)



(b) Algorithm 4 (AAC)
$\alpha = 1$

Numerical Experiments ($n = 1$)

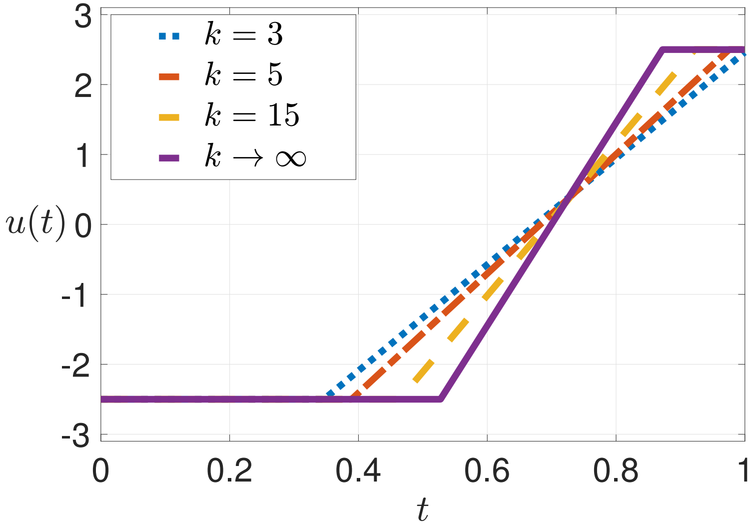
Parametric Behaviour



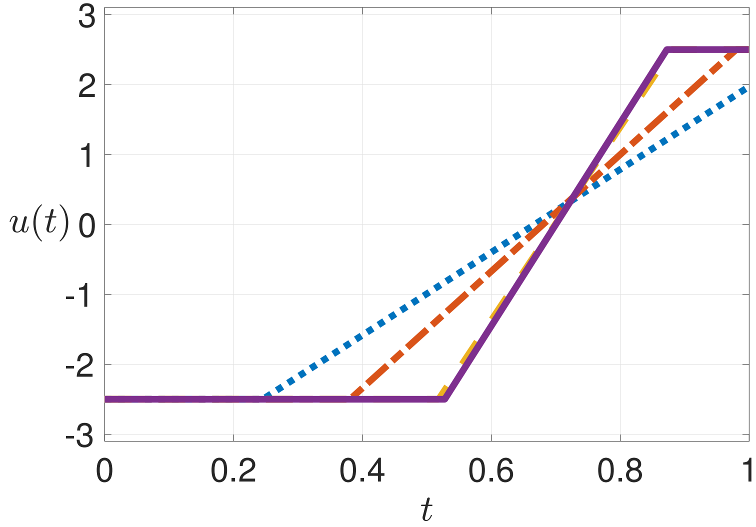
(c) Algorithm 4 (AAC)

Numerical Experiments ($n = 1$)

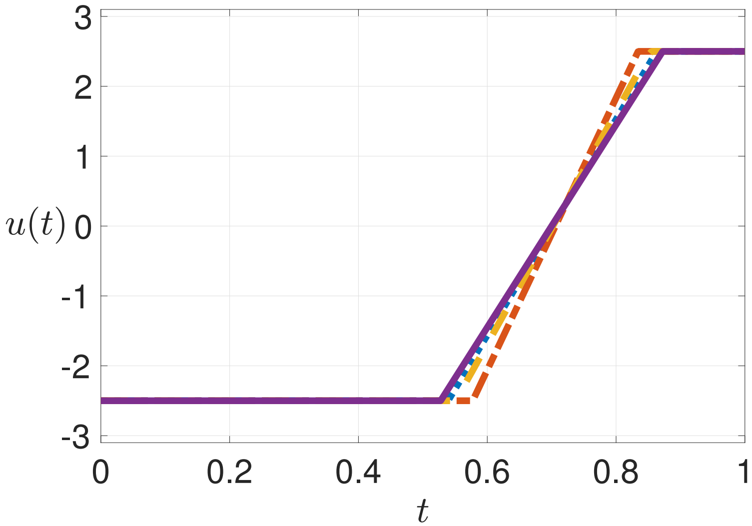
Behaviour in Early Iterations ($N = 2 \times 10^3$)



(a) Algorithm 1 (Dykstra).



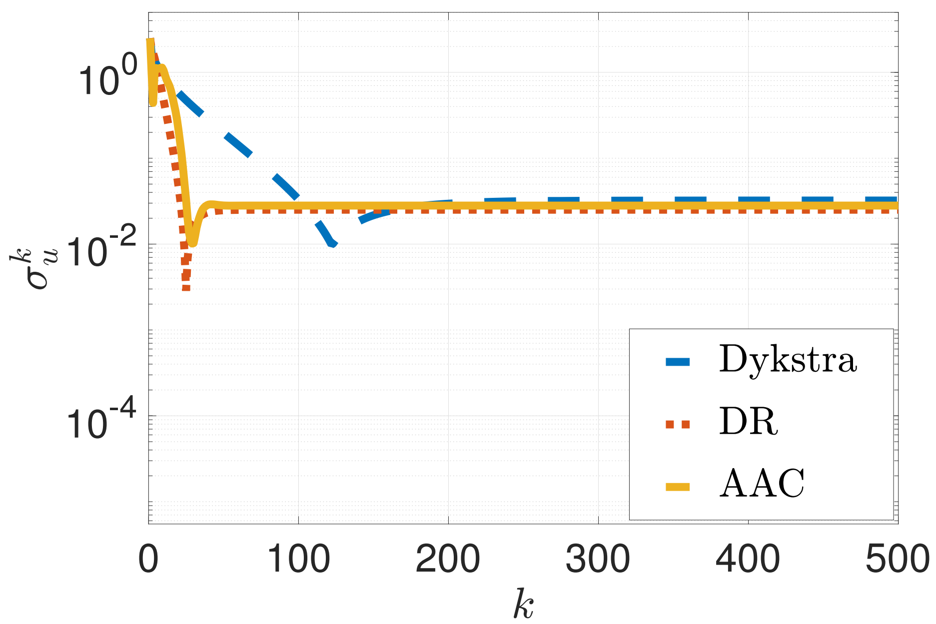
(b) Algorithm 3 (DR, $\lambda = 0.7466$).



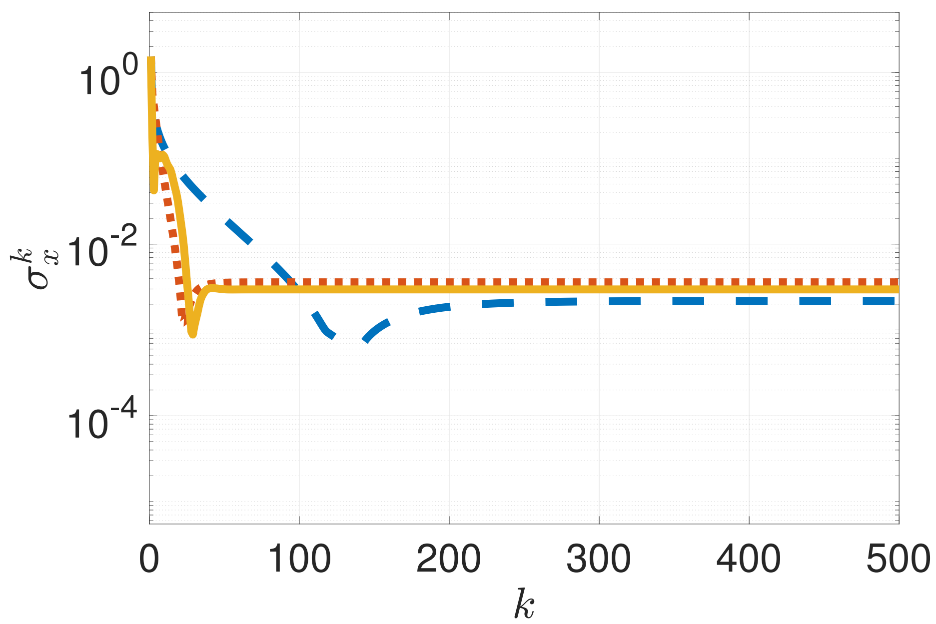
(c) Algorithm 4 (AAC, $\alpha = 1$, $\beta = 0.8617$).

Numerical Experiments ($n = 1$)

Error in Each Iteration



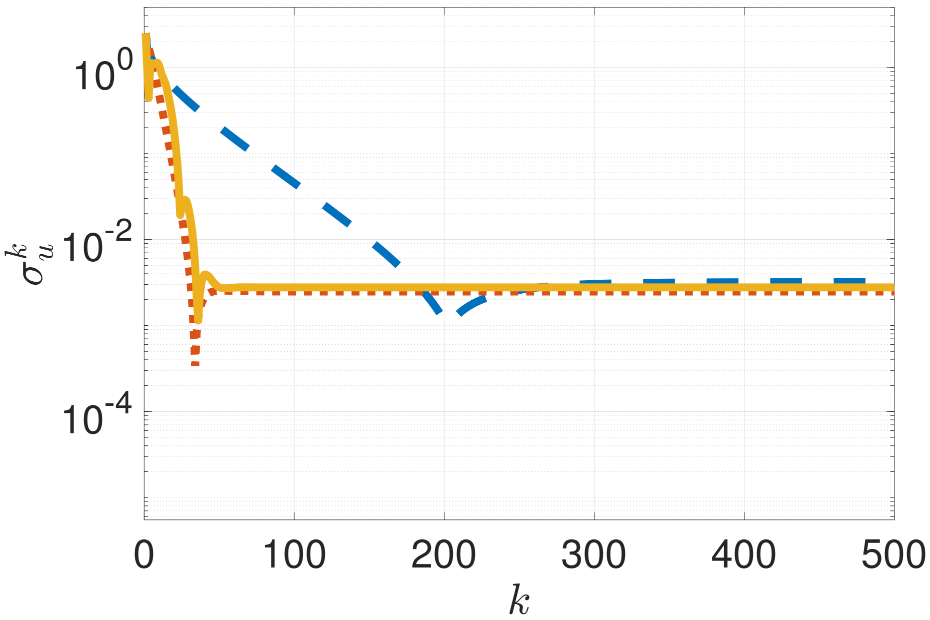
(a) L^∞-error in control with $N = 10^3$.



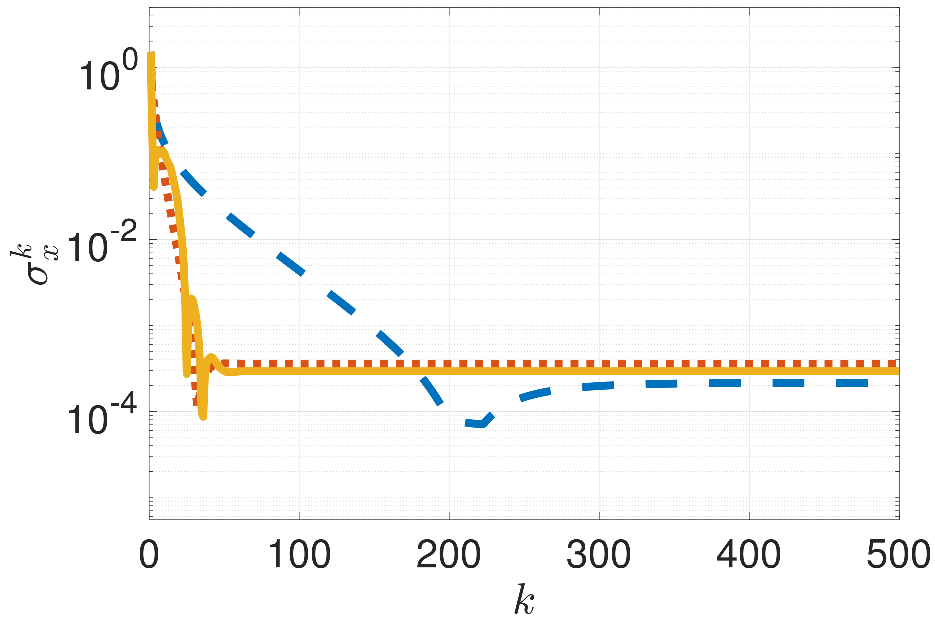
(b) L^∞-error in states with $N = 10^3$.

Numerical Experiments ($n = 1$)

Error in Each Iteration



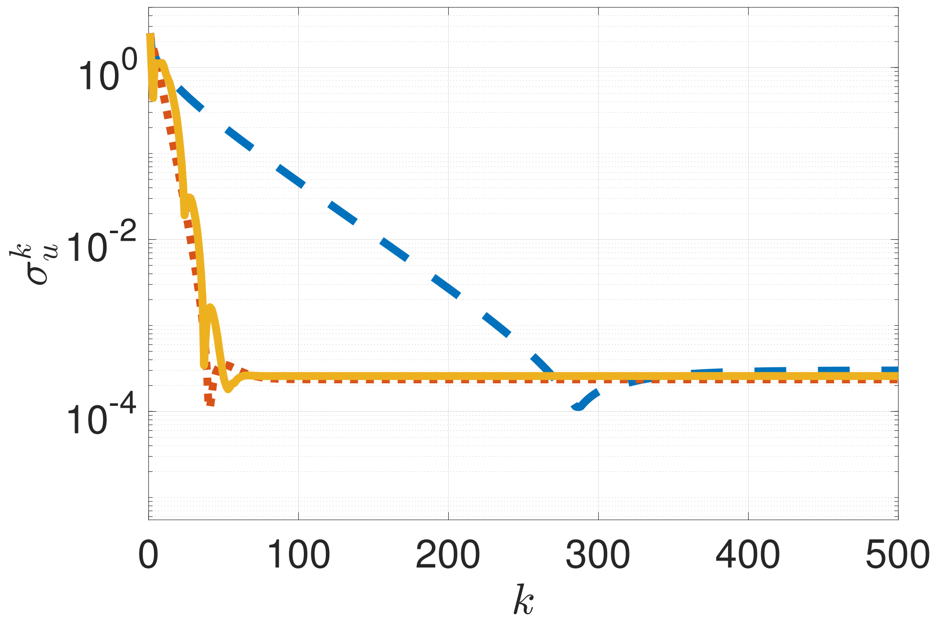
(a) L^∞-error in control with $N = 10^4$.



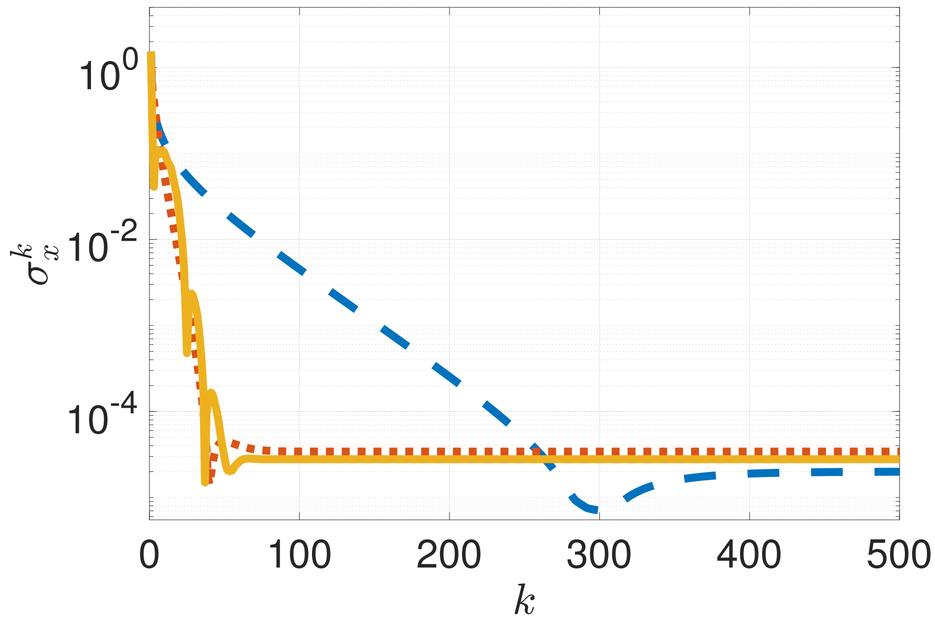
(b) L^∞-error in states with $N = 10^4$.

Numerical Experiments ($n = 1$)

Error in Each Iteration



(a) L^∞-error in control with $N = 10^5$.



(b) L^∞-error in states with $N = 10^5$.

Numerical Experiments ($n = 1$)

N	Dykstra	DR	AAC	Ipopt
10^3	3.2×10^{-2}	2.5×10^{-2}	2.8×10^{-2}	3.2×10^{-2}
10^4	3.2×10^{-3}	2.5×10^{-3}	2.8×10^{-3}	7.7×10^{-3}
10^5	3.0×10^{-4}	2.4×10^{-4}	2.6×10^{-4}	1.6×10^{-2}

(a) L^∞-error in control, σ_u^k .

N	Dykstra	DR	AAC	Ipopt
10^3	2.2×10^{-3}	3.6×10^{-3}	3.0×10^{-3}	2.2×10^{-3}
10^4	2.1×10^{-4}	3.6×10^{-4}	2.9×10^{-4}	2.3×10^{-4}
10^5	2.0×10^{-5}	3.4×10^{-5}	2.8×10^{-5}	8.7×10^{-5}

(b) L^∞-error in states, σ_x^k .

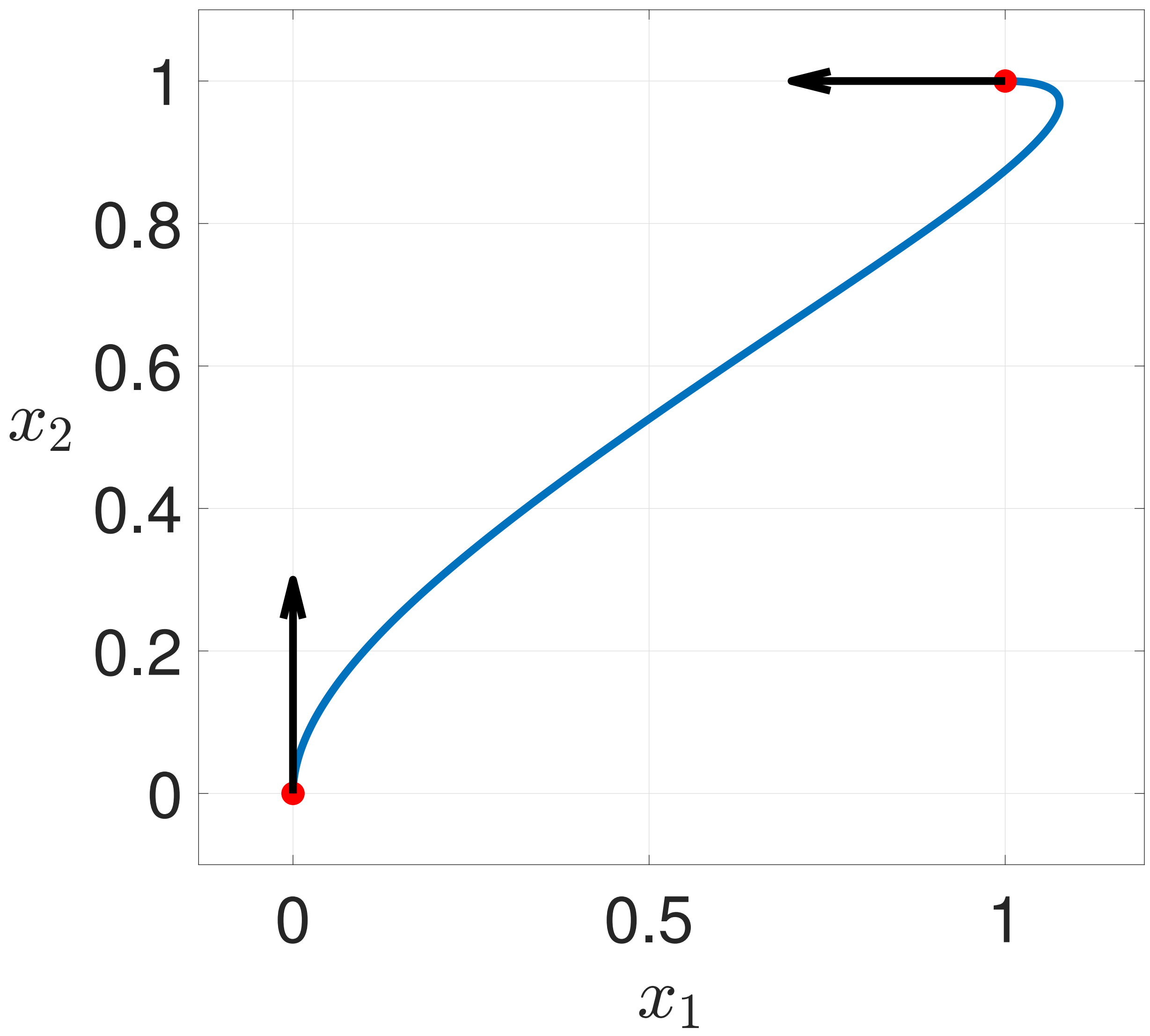
Table 1: Least errors by Algorithms 1, 3–4 and Ipopt ($\varepsilon = 10^{-8}$)

Numerical Experiments ($n = 1$)

N	Dykstra	DR	AAC	Ipopt
10^3	0.03	0.01	0.01	0.08
10^4	0.16	0.05	0.05	0.71
10^5	1.6	0.41	0.28	7.3

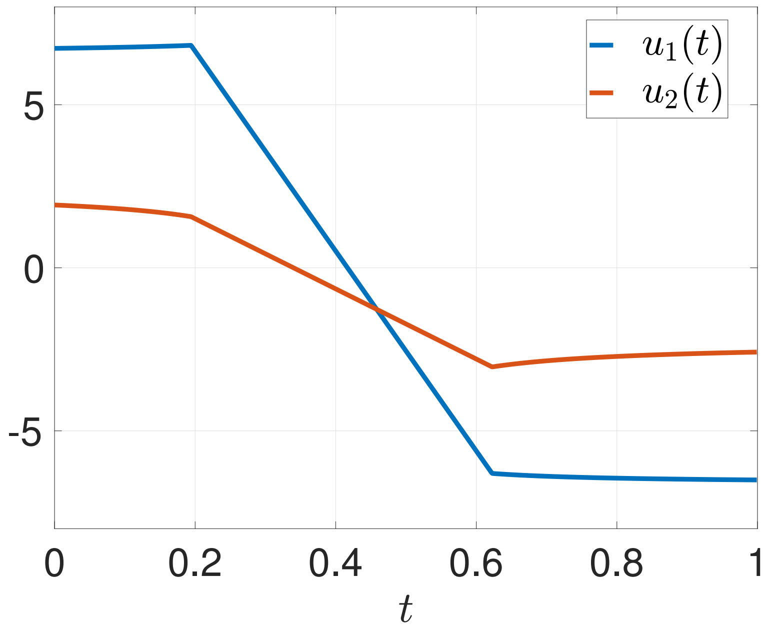
Table 2: CPU times taken by Algorithms 1, 3–4 and Ipopt.

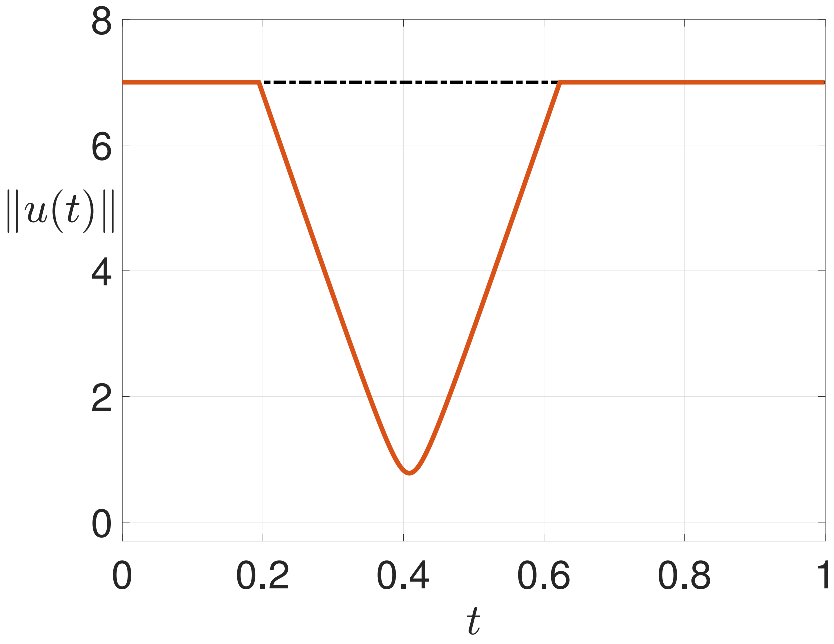
Numerical Experiments ($n = 2$)

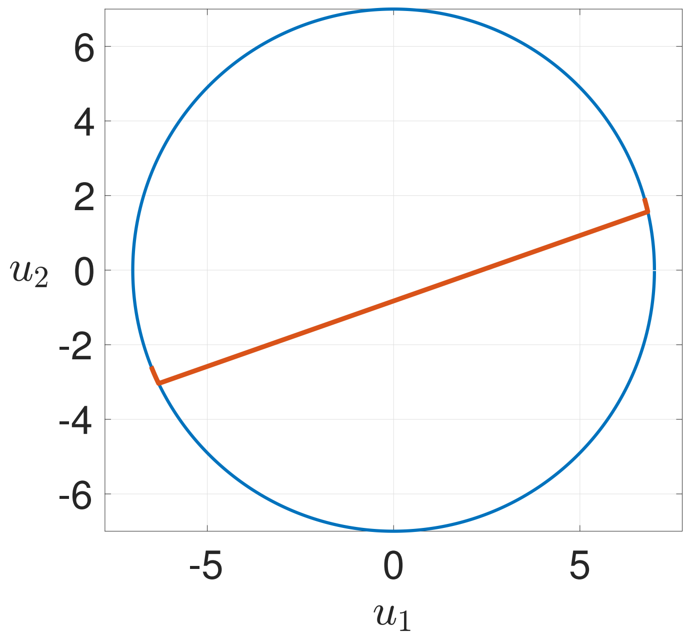


$s_0 = (0, 0), \; v_0 = (0, 1), \quad s_f = (1, 1), \; v_f = (-1, 0)$

Numerical Experiments ($n = 2$)

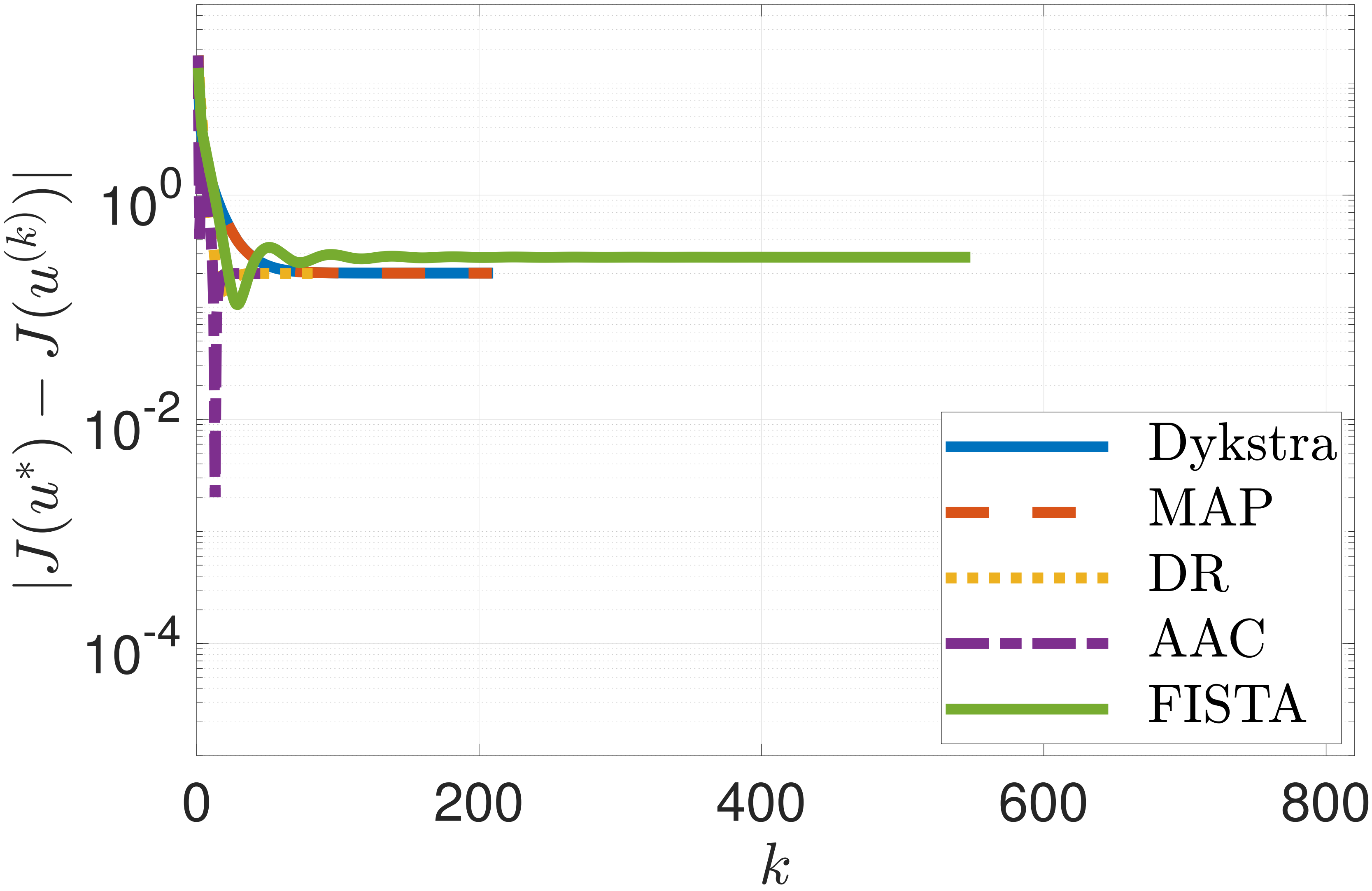






Numerical Experiments ($n = 2$)

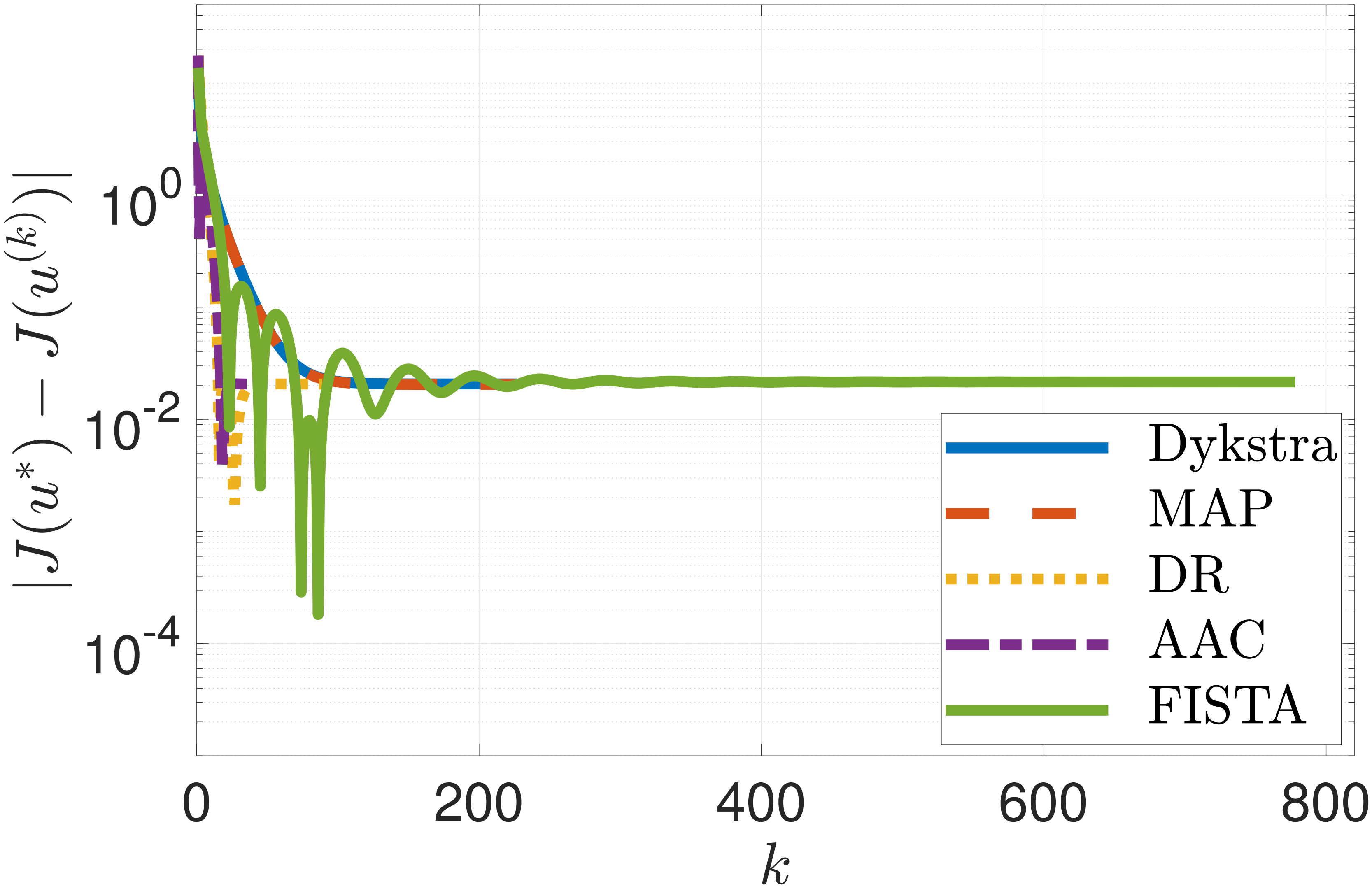
Objective Functional Error in Each Iteration



$N = 10^2$

Numerical Experiments ($n = 2$)

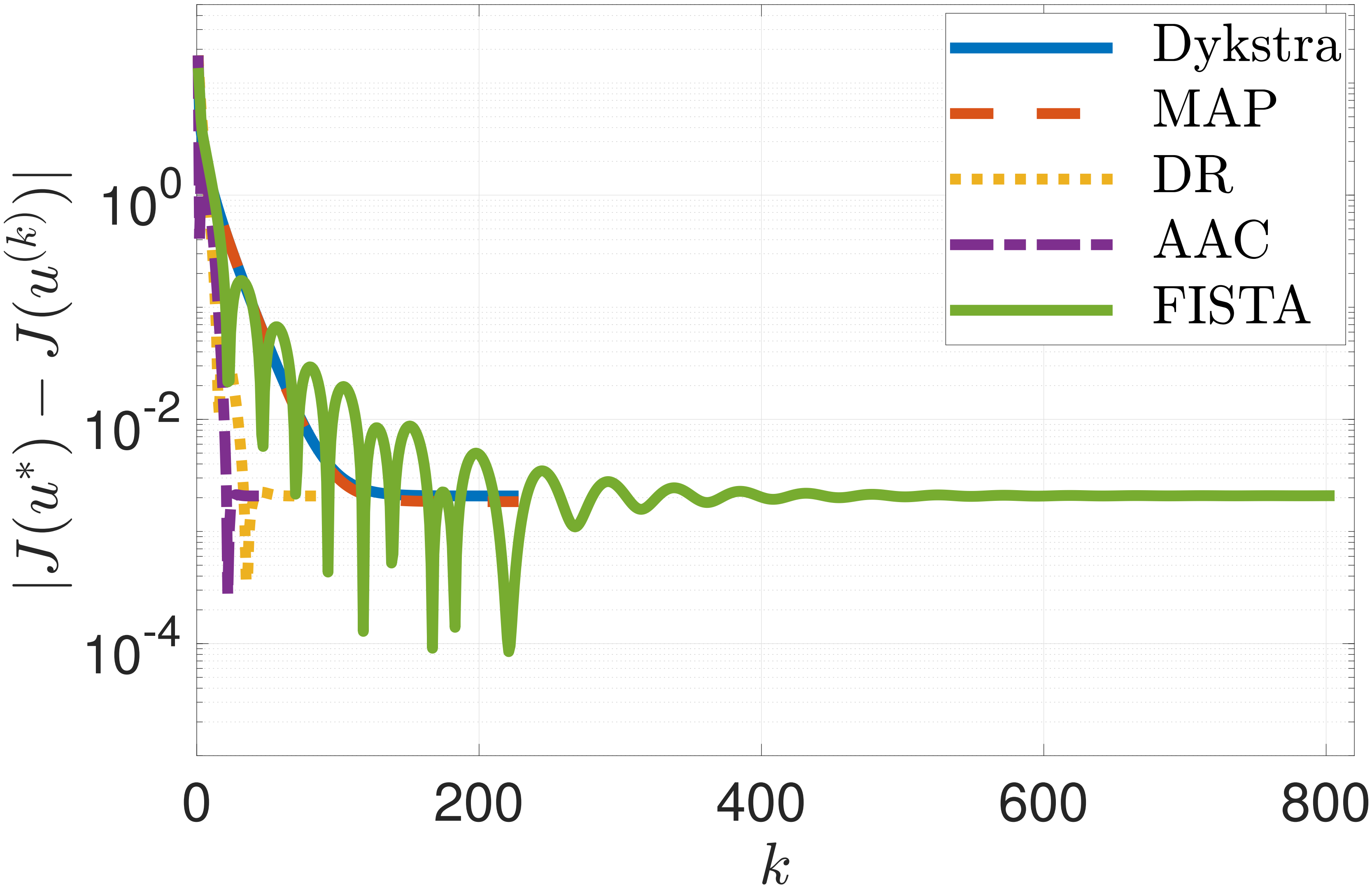
Objective Functional Error in Each Iteration



$N = 10^3$

Numerical Experiments ($n = 2$)

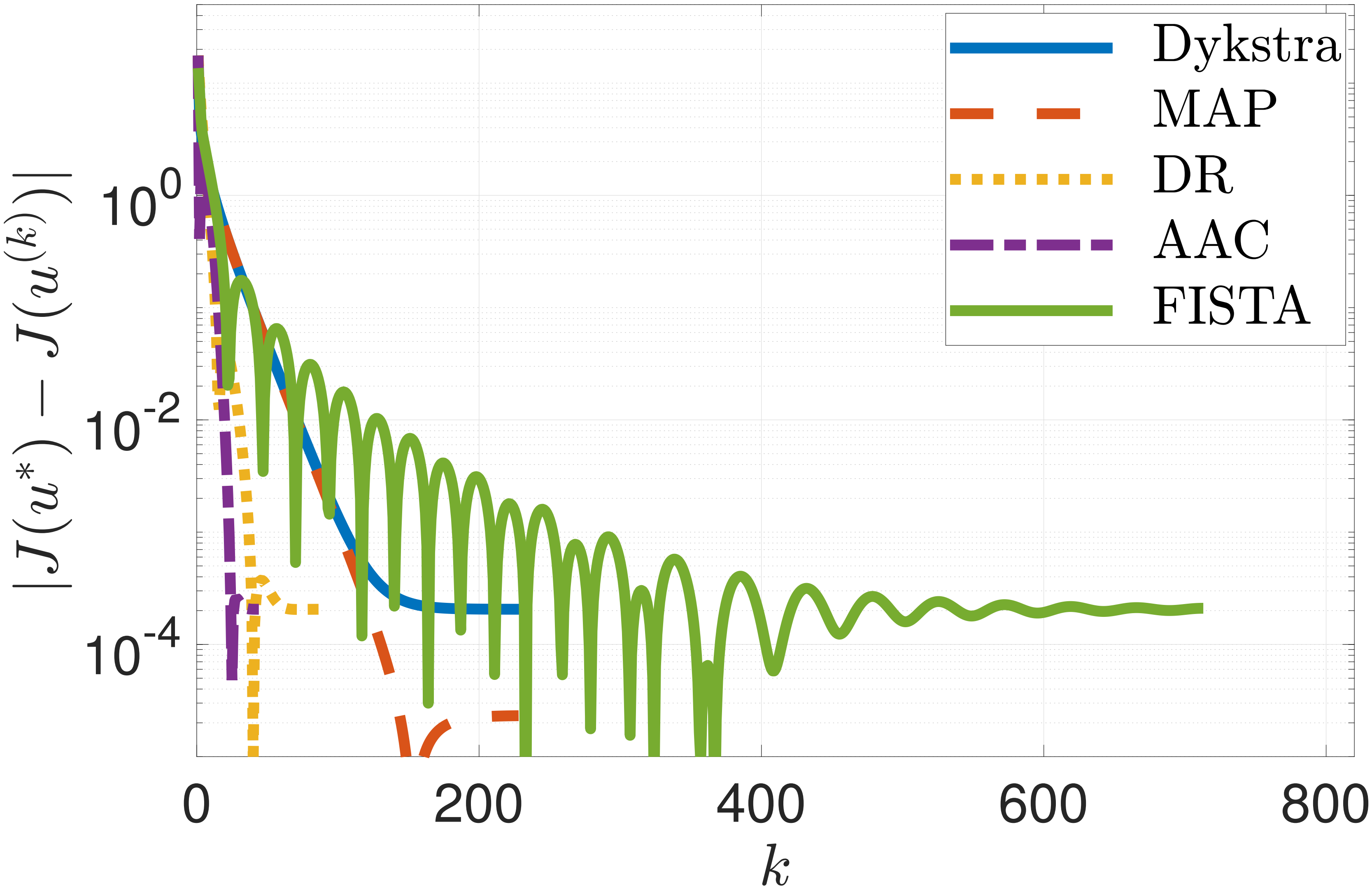
Objective Functional Error in Each Iteration



$N = 10^4$

Numerical Experiments ($n = 2$)

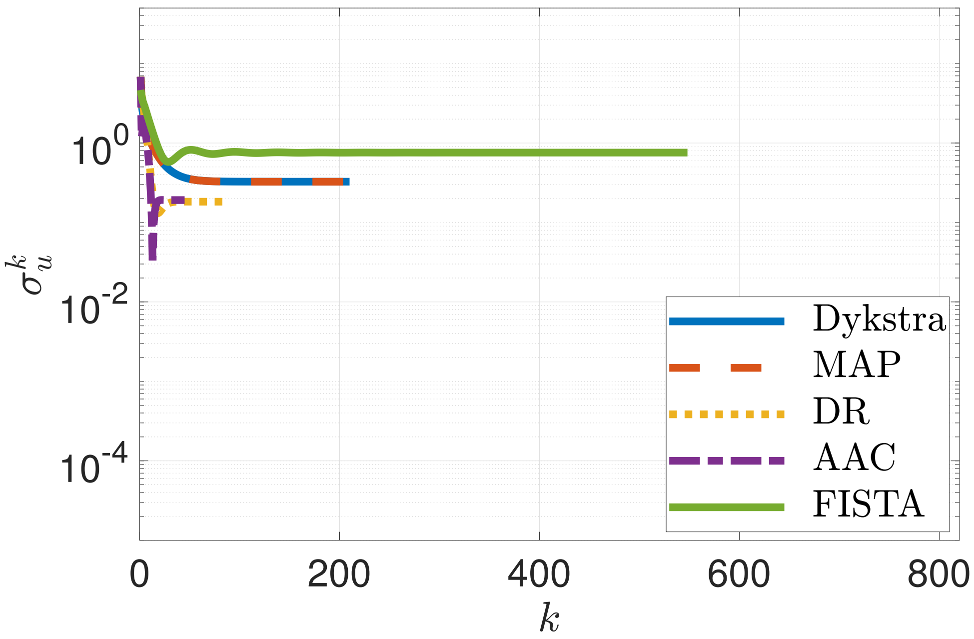
Objective Functional Error in Each Iteration



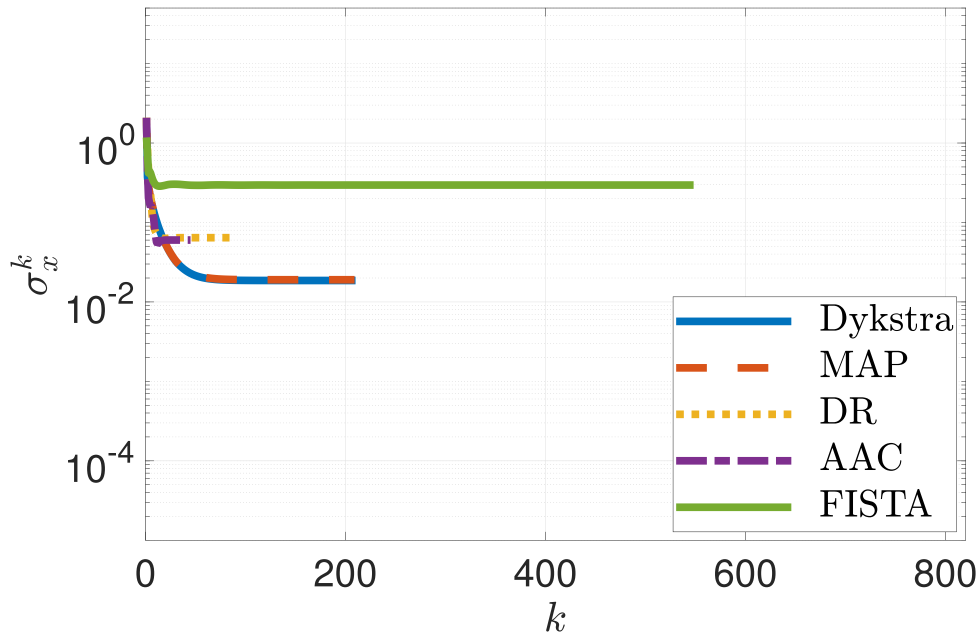
$N = 10^5$

Numerical Experiments ($n = 2$)

State and Control Error in Each Iteration



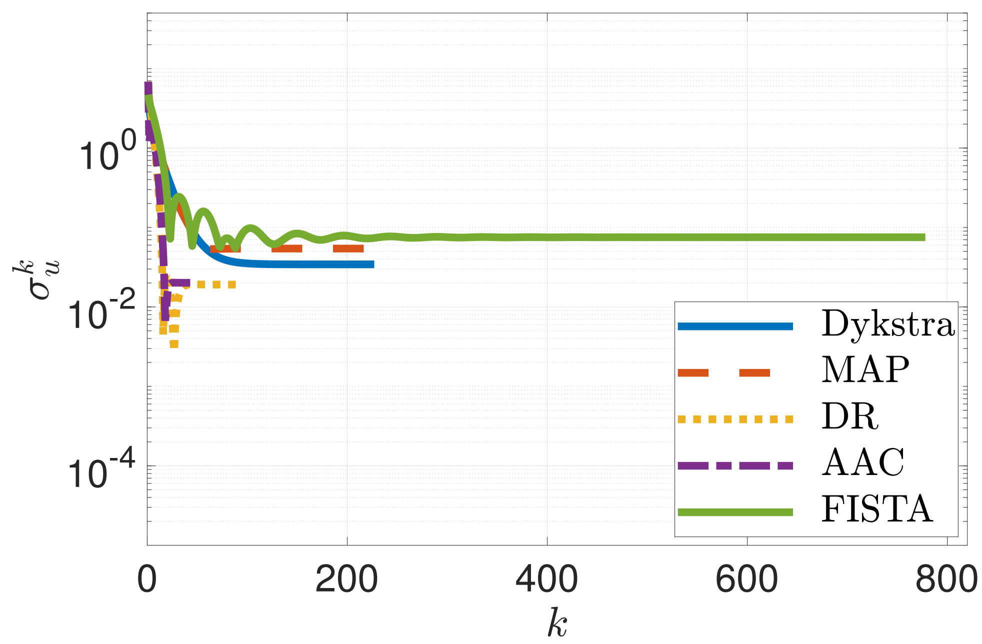
(a) L^∞-error in control with $N = 10^2$.



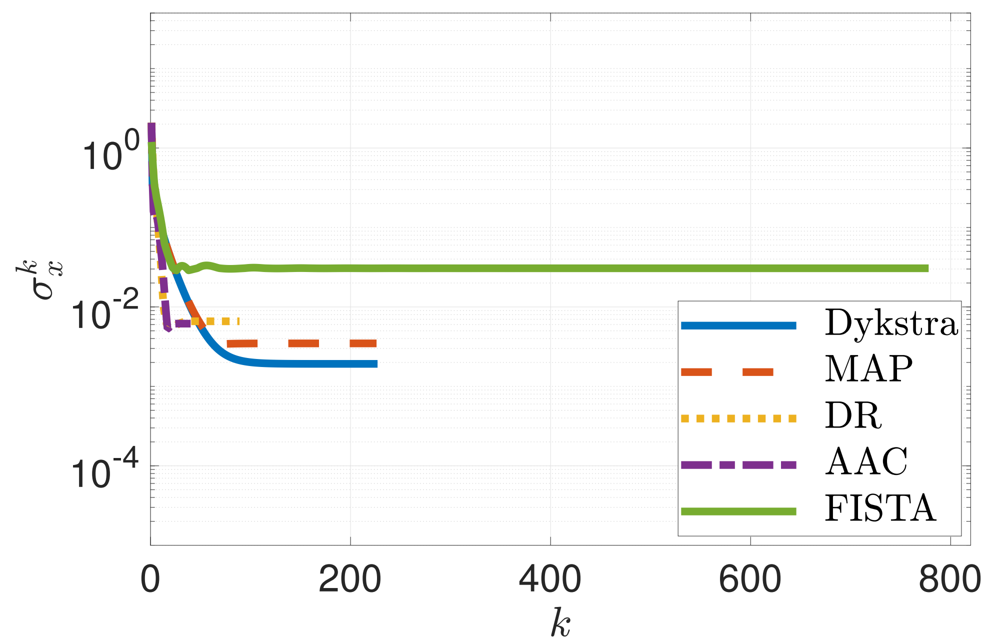
(b) L^∞-error in states with $N = 10^2$.

Numerical Experiments ($n = 2$)

State and Control Error in Each Iteration



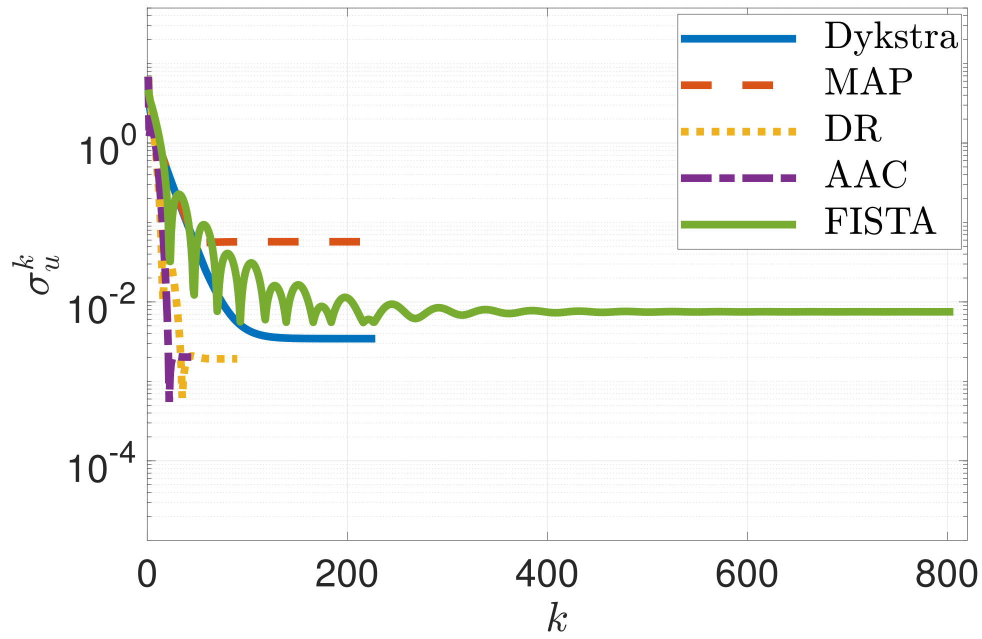
(a) L^∞-error in control with $N = 10^3$.



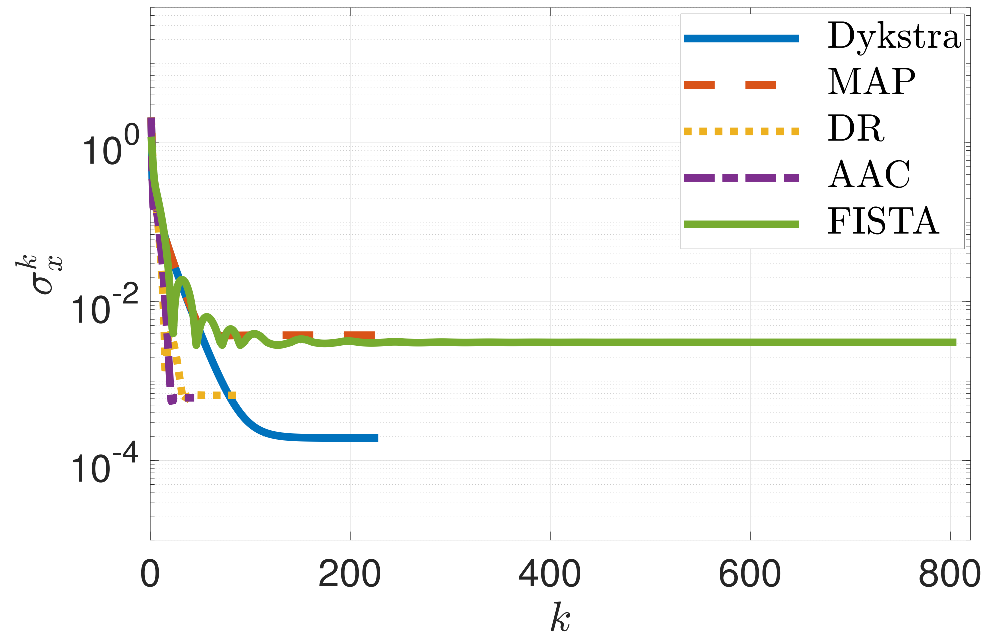
(b) L^∞-error in states with $N = 10^3$.

Numerical Experiments ($n = 2$)

State and Control Error in Each Iteration



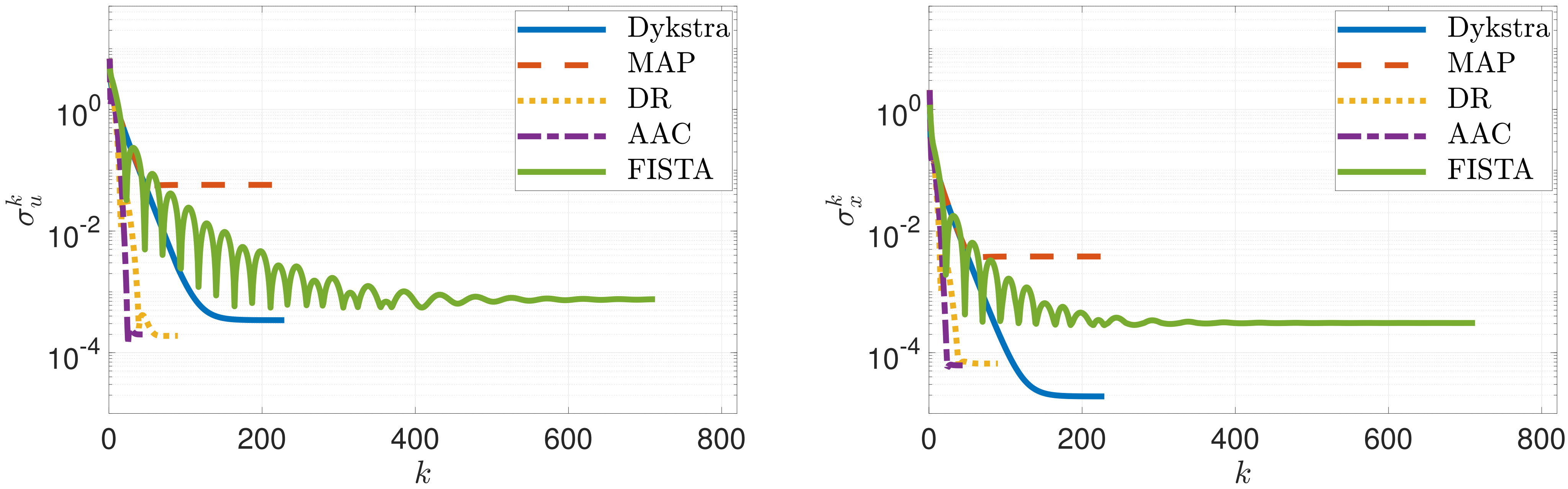
(a) L^∞-error in control with $N = 10^4$.



(b) L^∞-error in states with $N = 10^4$.

Numerical Experiments ($n = 2$)

State and Control Error in Each Iteration



(a) L^∞-error in control with $N = 10^5$.

(b) L^∞-error in states with $N = 10^5$.

Numerical Experiments ($n = 2$)

N	Dykstra	MAP	DR	AAC	FISTA
10^2	3.3×10^{-1}	3.3×10^{-1}	1.8×10^{-1}	1.9×10^{-1}	7.6×10^{-1}
10^3	3.4×10^{-2}	5.4×10^{-2}	1.9×10^{-2}	2.0×10^{-2}	7.5×10^{-2}
10^4	3.4×10^{-3}	5.7×10^{-2}	1.9×10^{-3}	2.0×10^{-3}	7.5×10^{-3}
10^5	3.4×10^{-4}	5.8×10^{-2}	1.9×10^{-4}	2.0×10^{-4}	7.6×10^{-4}

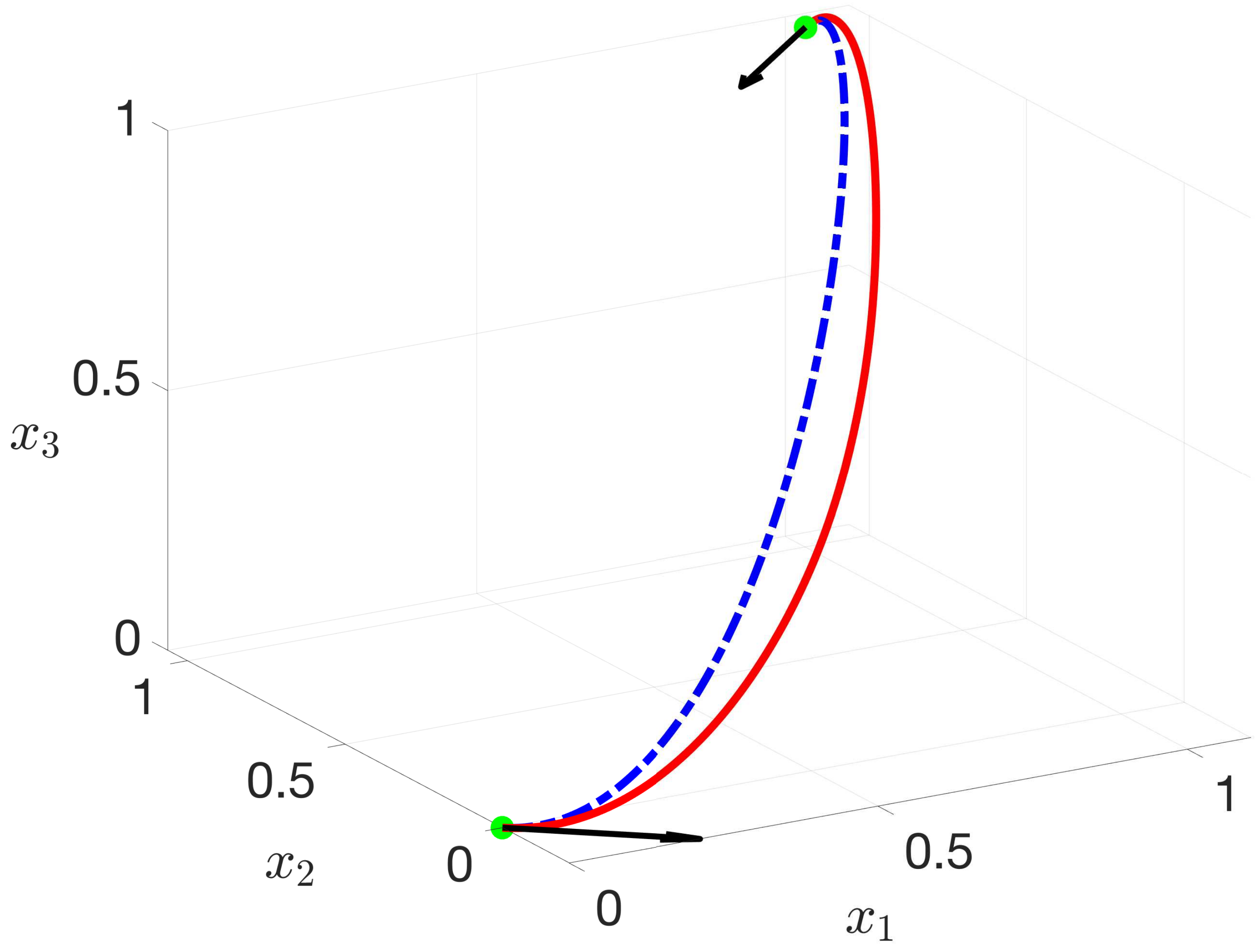
(a) L^∞-error in control, σ_u^k.

N	Dykstra	MAP	DR	AAC	FISTA
10^2	1.9×10^{-2}	1.9×10^{-2}	6.4×10^{-2}	6.0×10^{-2}	3.0×10^{-1}
10^3	1.9×10^{-3}	3.5×10^{-3}	6.6×10^{-3}	6.2×10^{-3}	3.1×10^{-2}
10^4	1.9×10^{-4}	3.8×10^{-3}	6.6×10^{-4}	6.2×10^{-4}	3.1×10^{-3}
10^5	1.9×10^{-5}	3.8×10^{-3}	6.6×10^{-5}	6.2×10^{-5}	3.1×10^{-4}

(b) L^∞-error in states, σ_x^k.

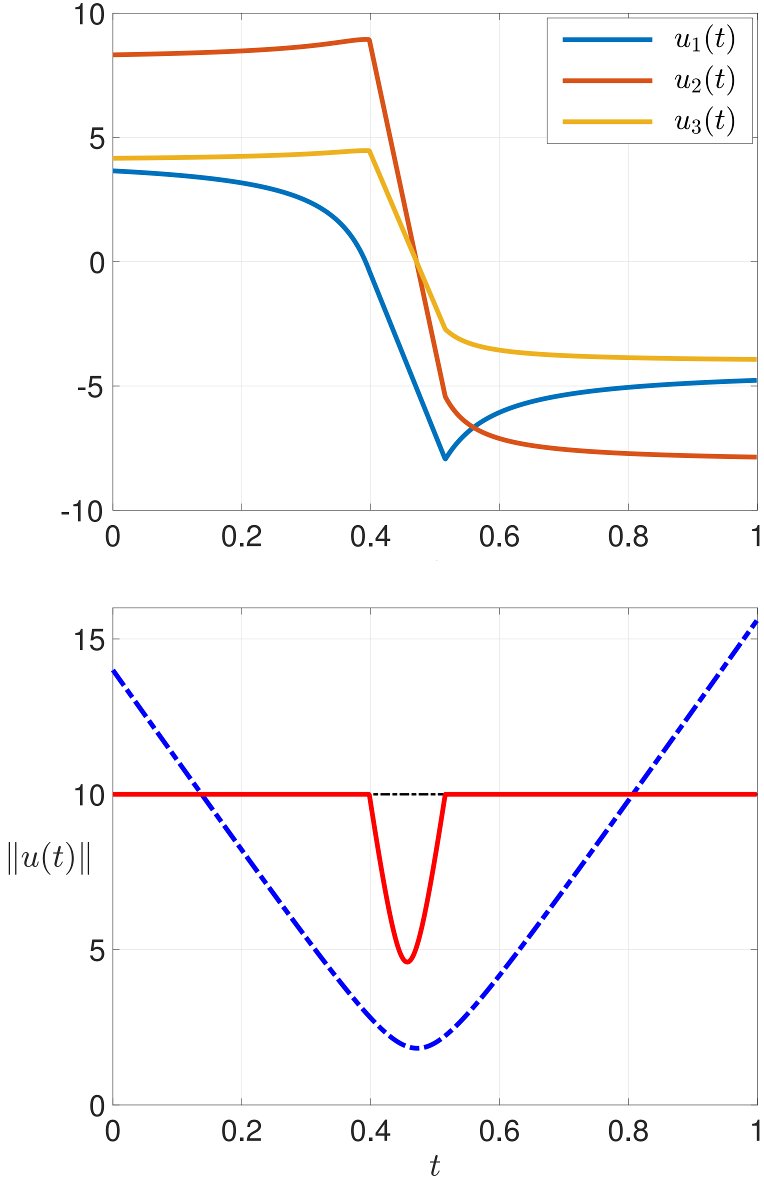
Table 3: Least errors by Algorithms 1–5 ($\varepsilon = 10^{-8}$)

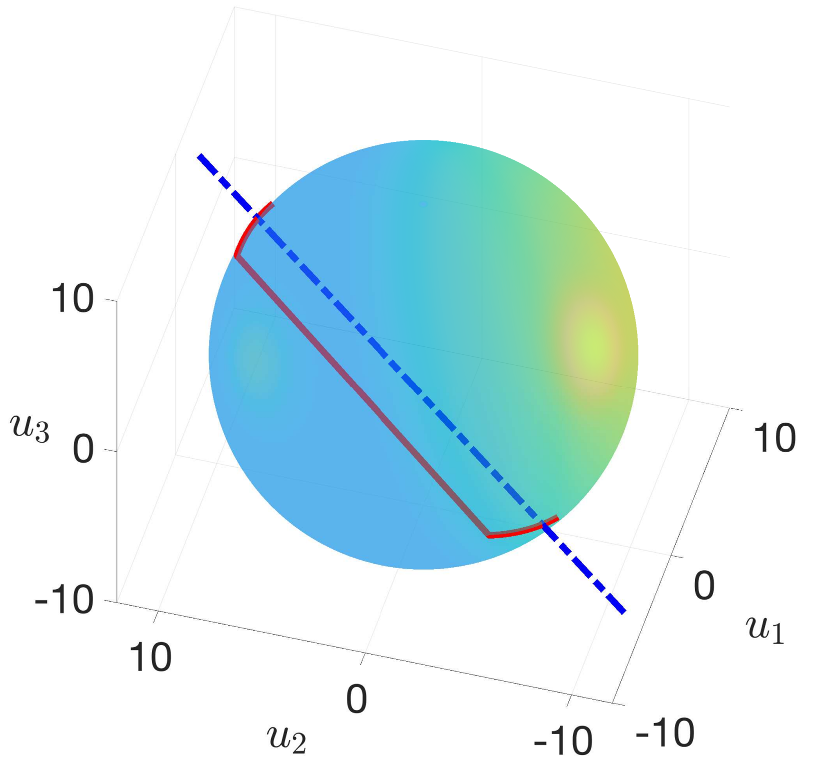
Numerical Experiments ($n = 3$)



$$s_0 = (0, 0, 0), \ \ v_0 = (1, -1, 0), \quad s_f = (1, 1, 1), \ \ v_f = (-1, -1, 0)$$

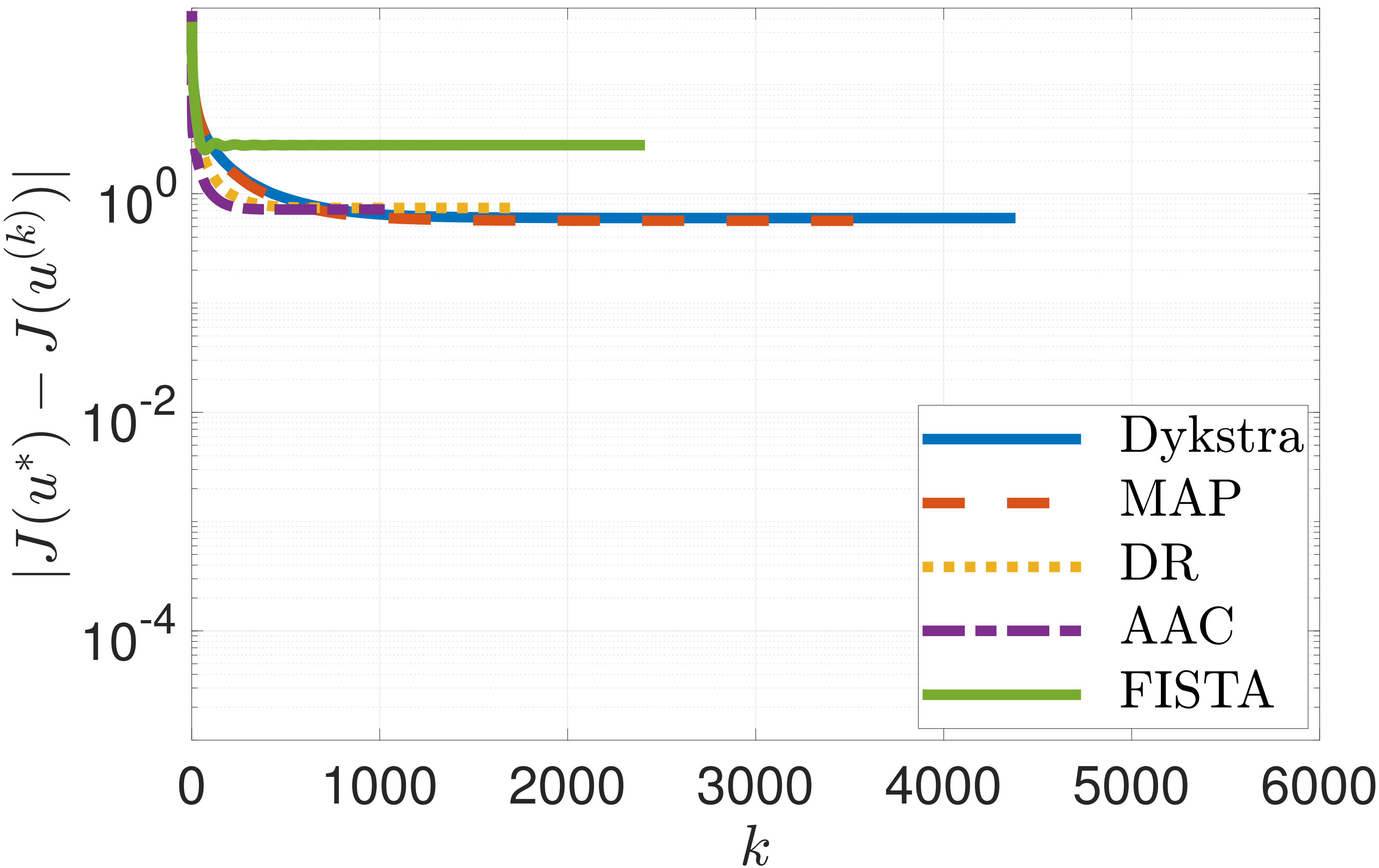
Numerical Experiments ($n = 3$)





Numerical Experiments ($n = 3$)

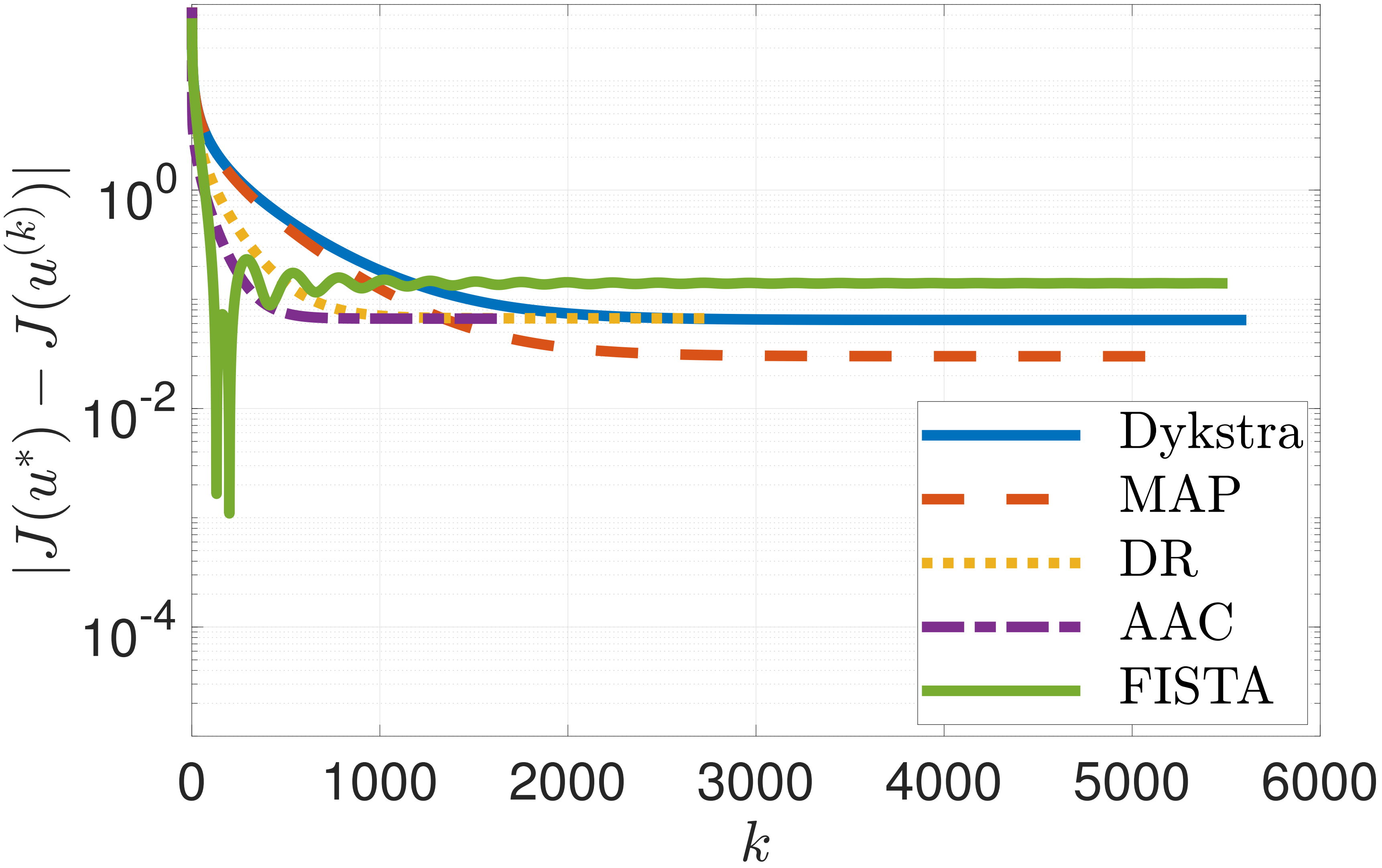
Objective Functional Error in Each Iteration



$N = 10^2$

Numerical Experiments ($n = 3$)

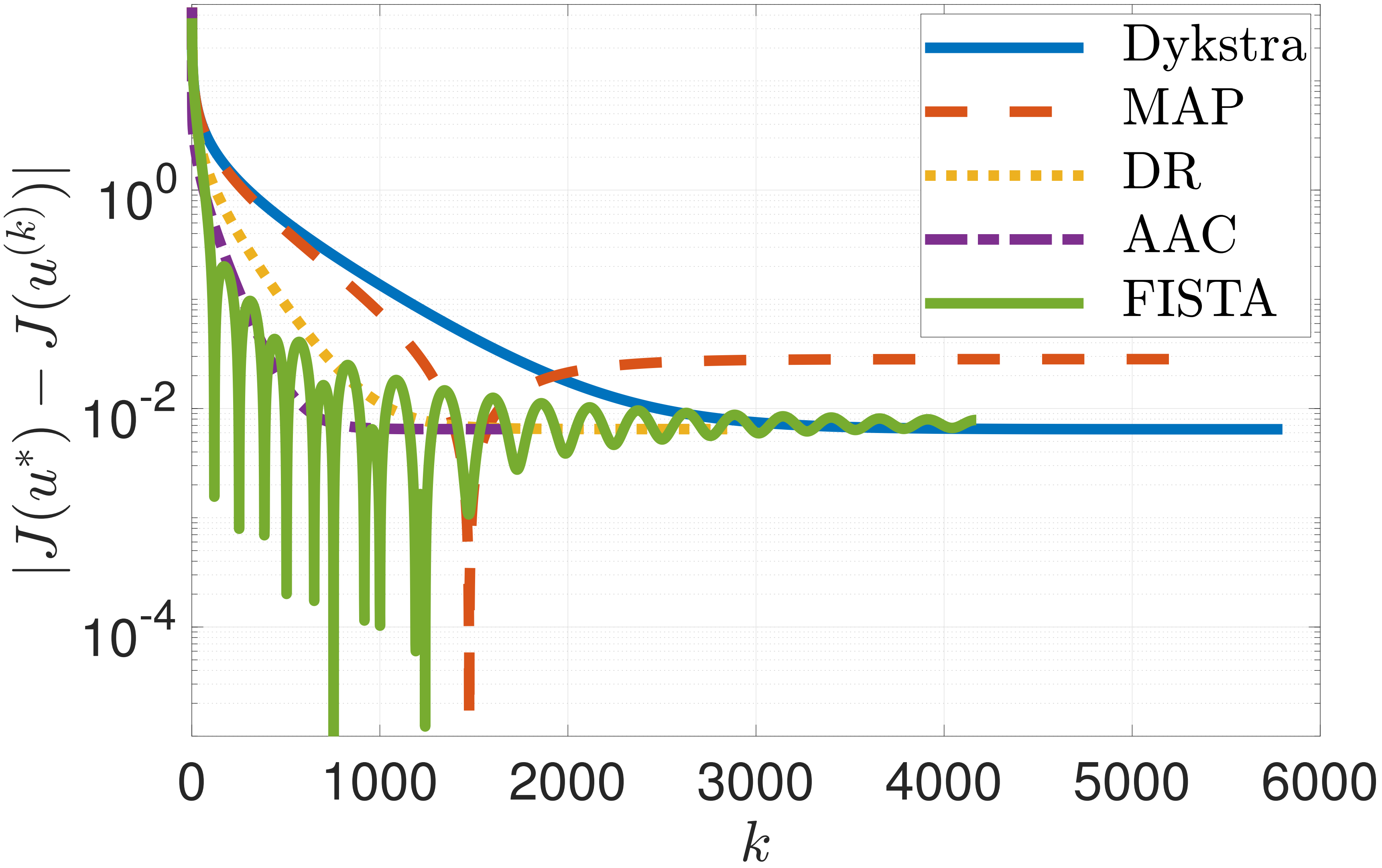
Objective Functional Error in Each Iteration



$N = 10^3$

Numerical Experiments ($n = 3$)

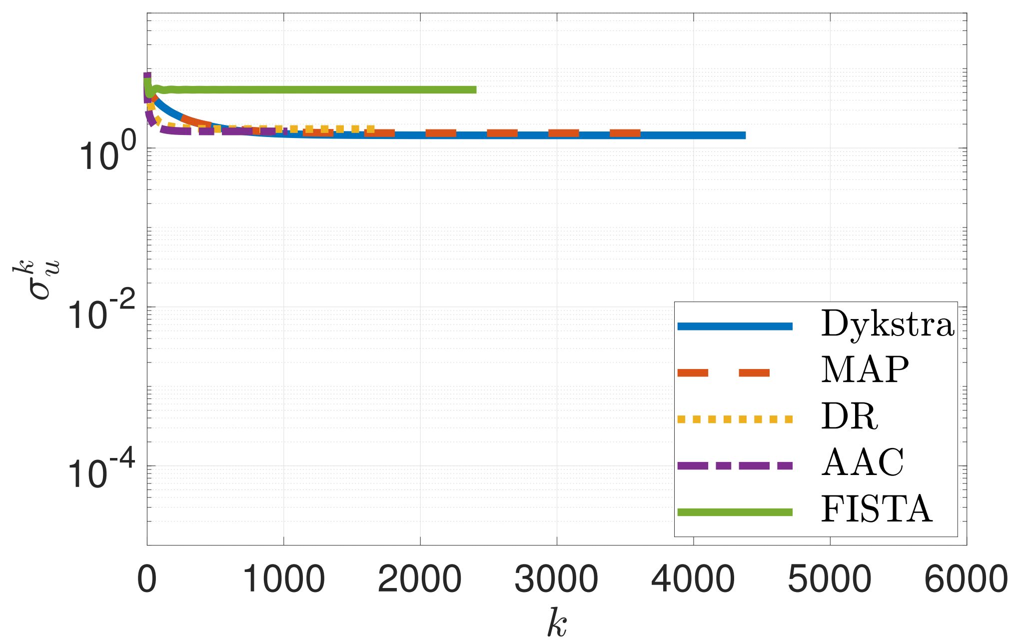
Objective Functional Error in Each Iteration



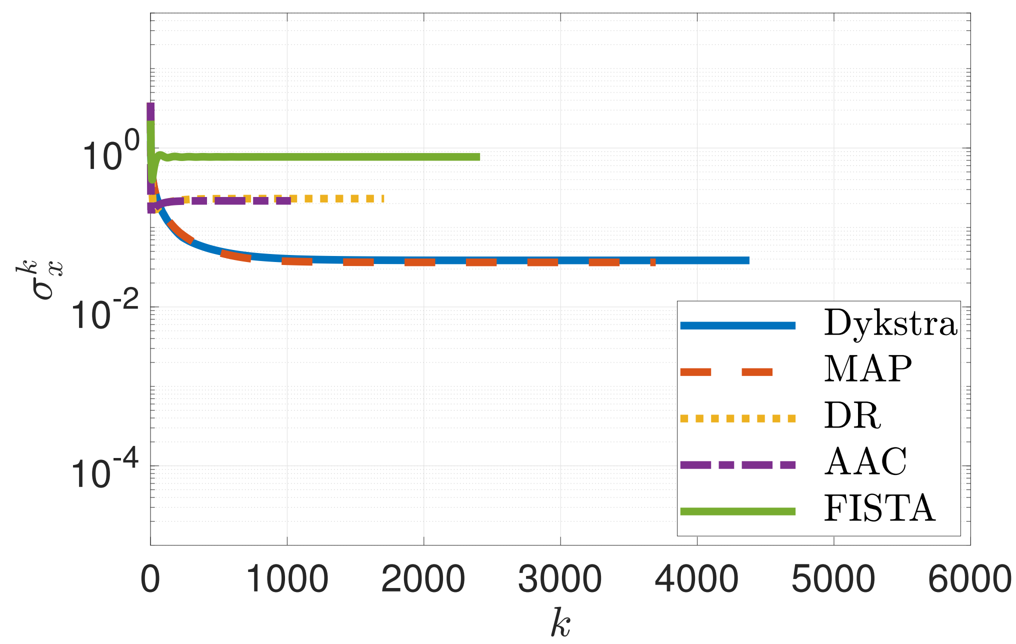
$N = 10^4$

Numerical Experiments ($n = 3$)

State and Control Error in Each Iteration



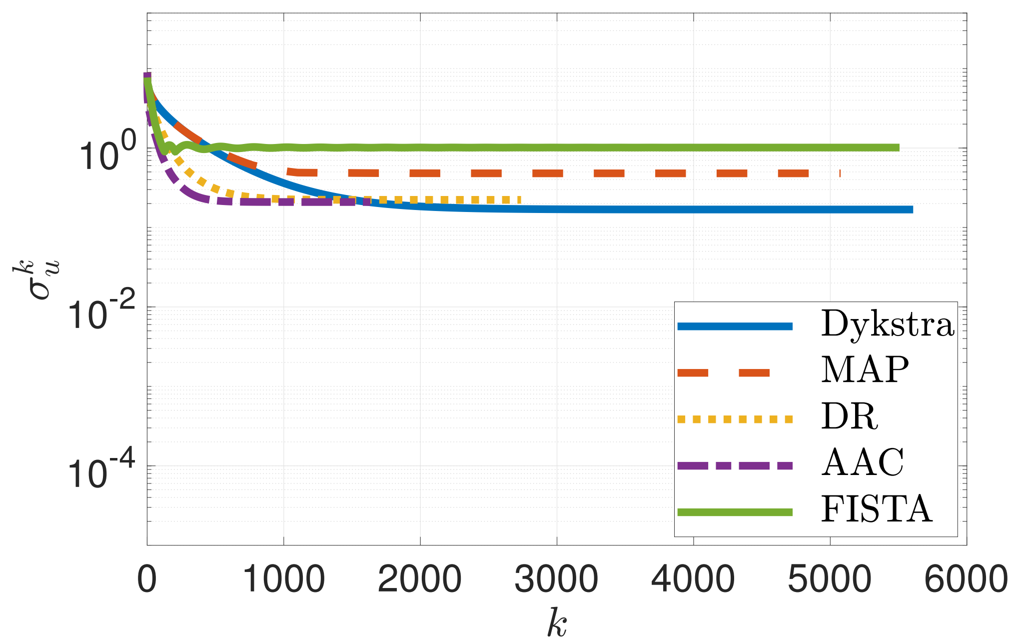
(a) L^∞-error in control with $N = 10^2$.



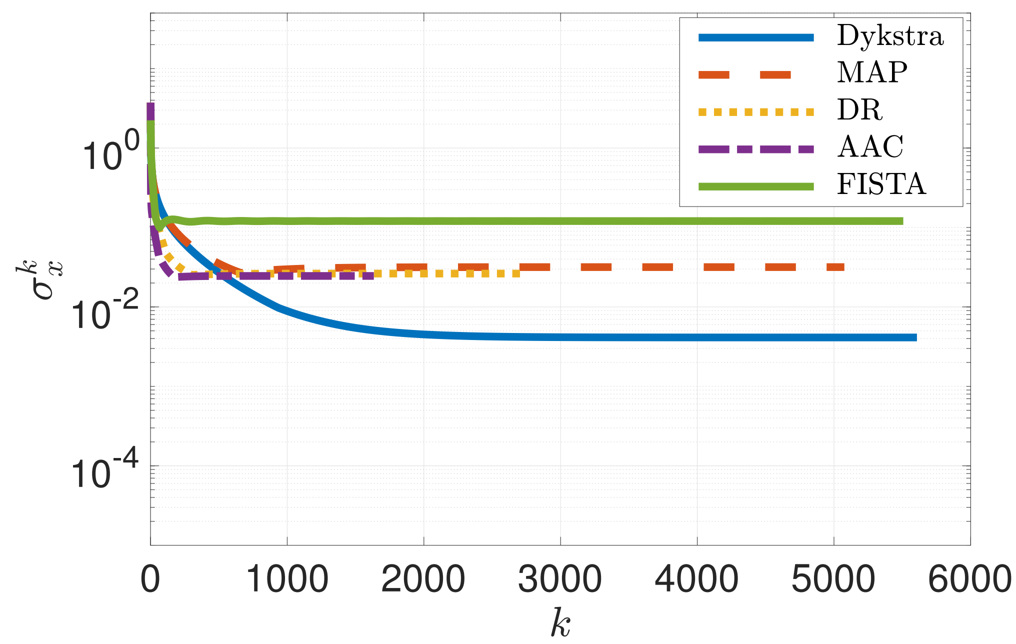
(b) L^∞-error in states with $N = 10^2$.

Numerical Experiments ($n = 3$)

State and Control Error in Each Iteration



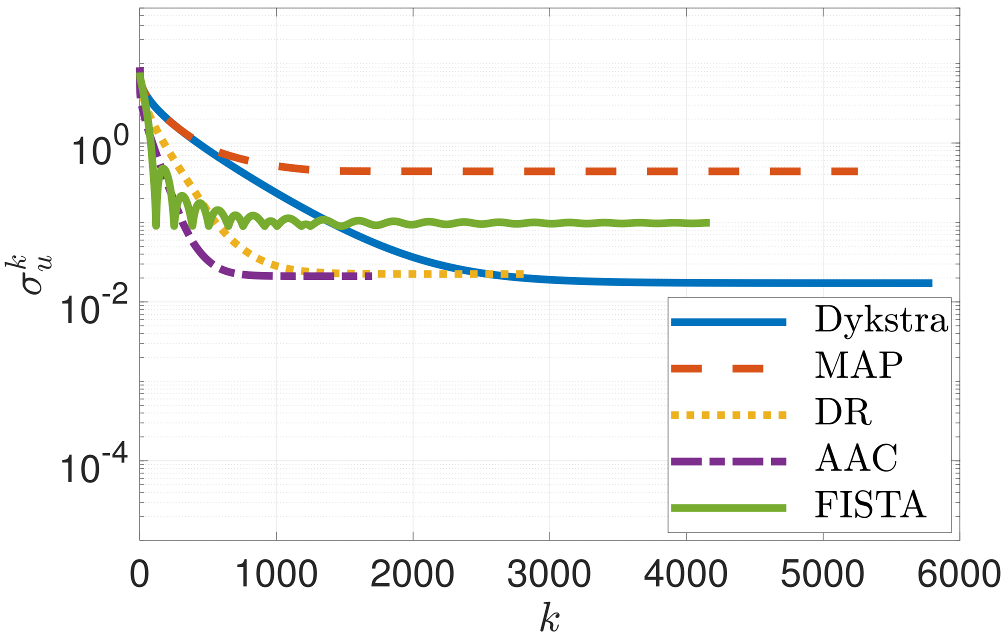
(a) L^∞-error in control with $N = 10^3$.



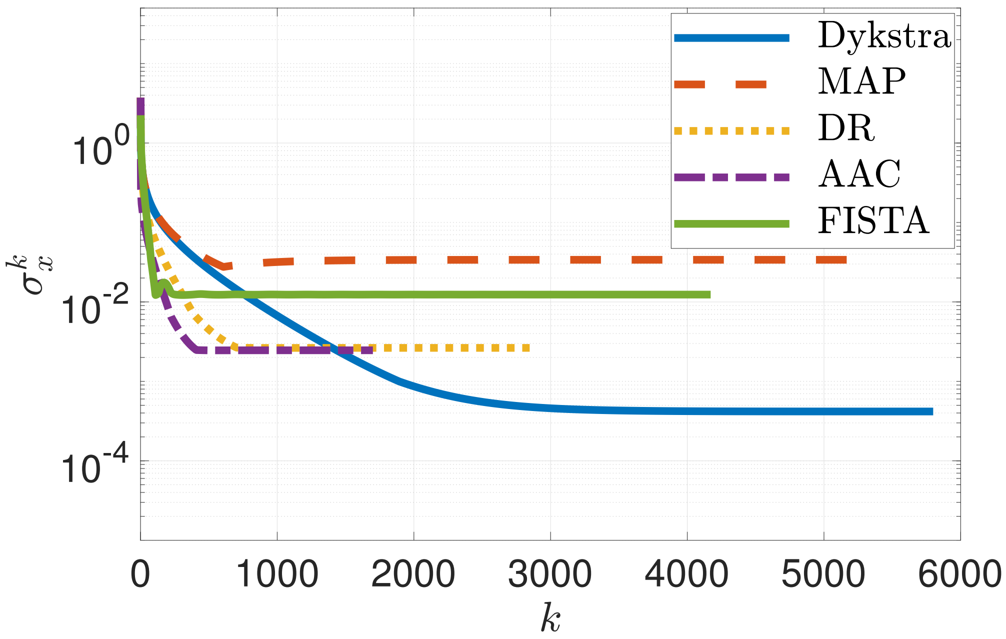
(b) L^∞-error in states with $N = 10^3$.

Numerical Experiments ($n = 3$)

State and Control Error in Each Iteration



(a) L^∞-error in control with $N = 10^4$.



(b) L^∞-error in states with $N = 10^4$.

Numerical Experiments ($n = 3$)

N	Dykstra	MAP	DR	AAC	FISTA
10^2	1.4×10^1	1.5×10^1	1.7×10^1	1.6×10^1	5.4×10^1
10^3	1.7×10^{-1}	4.8×10^{-1}	2.2×10^{-1}	2.1×10^{-1}	1.0×10^1
10^4	1.7×10^{-2}	4.4×10^{-1}	2.2×10^{-3}	2.1×10^{-3}	9.9×10^{-2}

(a) L^∞-error in control, σ_u^k .

N	Dykstra	MAP	DR	AAC	FISTA
10^2	3.9×10^{-2}	3.6×10^{-2}	2.3×10^{-1}	2.2×10^{-1}	7.7×10^{-1}
10^3	4.1×10^{-3}	3.2×10^{-2}	2.6×10^{-2}	2.5×10^{-2}	1.2×10^{-1}
10^4	4.2×10^{-4}	3.4×10^{-2}	2.6×10^{-3}	2.5×10^{-3}	1.2×10^{-2}

(b) L^∞-error in states, σ_x^k .

Table 4: Least errors by Algorithms 1–5 ($\varepsilon = 10^{-8}$)

Conclusion and Open Problems

We observe and note that

- Dykstra, DR, AAC (Algorithms 1, 3 and 4) are the most successful. Dykstra is best in generating optimal states (position and velocity).
- Projection methods are better than the standard discretization approach.
- MAP is observed to converge only weakly for $n = 2$ and 3.
- Models and algorithms here are prototypes for future extensions.

Conclusion and Open Problems

We observe and note that

- Dykstra, DR, AAC (Algorithms 1, 3 and 4) are the most successful. Dykstra is best in generating optimal states (position and velocity).
- Projection methods are better than the standard discretization approach.
- MAP is observed to converge only weakly for $n = 2$ and 3.
- Models and algorithms here are prototypes for future extensions.

Future work

- If $u^-(t)$ is piecewise linear then its projection is piecewise linear. This might simplify further the projection expressions.
- Extension to general control-constrained linear-quadratic problems.
- Extension to nonconvex optimal control problems.