Oaxaca September 2019

An SDDP-like algorithm for infinite horizon multistage stochastic programmes

Regan Baucke

CERMICS, Ecole des Ponts
regan.baucke@enpc.fr

We are familiar with the deterministic control problem

$$\begin{aligned} \min_{u} \quad & \sum_{t=0}^{T} C_t(x_t, u_t) + V_T(x_T) \\ \text{s.t.} \quad & u_t \in \mathcal{U}_t(x_t),\ \forall t, \\ & x_t \in \mathcal{X}_t,\ \forall t, \\ & x_{t+1} = f_{t+1}(x_t, u_t),\ \forall t. \end{aligned}$$

We are familiar with the deterministic control problem

$$
\begin{aligned}
\min_{u} \quad & \sum_{t=0}^{T} C_t(x_t, u_t) + V_T(x_T) \\
\text{s.t.} \quad & u_t \in \mathcal{U}_t(x_t), \ \forall t, \\
& x_t \in \mathcal{X}_t, \ \forall t, \\
& x_{t+1} = f_{t+1}(x_t, u_t), \ \forall t.
\end{aligned}
$$

We are familiar with the deterministic control problem

$$
\begin{aligned}
\min_{u} \quad & \sum_{t=0}^{T} C_t(x_t, u_t) + V_T(x_T) \\
\text{s.t.} \quad & u_t \in \mathcal{U}_t(x_t),\ \forall t, \\
& x_t \in \mathcal{X}_t,\ \forall t, \\
& x_{t+1} = f_{t+1}(x_t, u_t),\ \forall t.
\end{aligned}
$$

We are familiar with the deterministic control problem

$$
\begin{aligned}
\min_{u} \quad & \sum_{t=0}^{T} C_t(x_t, u_t) + V_T(x_T) \\
\text{s.t.} \quad & u_t \in \mathcal{U}_t(x_t),\ \forall t, \\
& x_t \in \mathcal{X}_t,\ \forall t, \\
& x_{t+1} = f_{t+1}(x_t, u_t),\ \forall t.
\end{aligned}
$$

We are familiar with the deterministic control problem

$$
\begin{aligned}
\min_{u} \quad & \sum_{t=0}^{T} C_t(x_t, u_t) + V_T(x_T) \\
\text{s.t.} \quad & u_t \in \mathcal{U}_t(x_t),\ \forall t, \\
& x_t \in \mathcal{X}_t,\ \forall t, \\
& x_{t+1} = f_{t+1}(x_t, u_t),\ \forall t.
\end{aligned}
$$

We are familiar with the deterministic control problem

$$
\begin{aligned}
\min_{u} \quad & \sum_{t=0}^{T} C\left(x_t, u_t\right) + V_T(x_T) \\
\text{s.t.} \quad & u_t \in \mathcal{U}\left(x_t\right),\ \forall t, \\
& x_t \in \mathcal{X}\ ,\ \forall t, \\
& x_{t+1} = f \quad (x_t, u_t),\ \forall t.
\end{aligned}
$$

We are familiar with the deterministic control problem

$$
\begin{aligned}
\min_{u} \quad & \sum_{t=0}^{\infty} \gamma^t C\,(x_t, u_t), \quad \gamma \in [0,1), \\
\text{s.t.} \quad & u_t \in \mathcal{U}\,(x_t),\ \forall t, \\
& x_t \in \mathcal{X}\,,\ \forall t, \\
& x_{t+1} = f \quad (x_t, u_t),\ \forall t.
\end{aligned}
$$

1. The naïve approach
2. Lipschitz considerations
3. Our algorithm
4. To the stochastic case

We can approximate our infinite horizon problem with:

$$\begin{aligned} \min_{u} \quad & \sum_{t=0}^{T} \gamma^t C(x_t, u_t), \quad \gamma \in [0,1), \\ \text{s.t.} \quad & u_t \in \mathcal{U}(x_t),\ \forall t, \\ & x_t \in \mathcal{X},\ \forall t, \\ & x_{t+1} = f(x_t, u_t),\ \forall t. \end{aligned}$$

We define the following Bellman operator

$$\mathbb{B}_t[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} \gamma^t C(x,u) + V(f(x,u)),$$

and "future-state" operator

$$\mathbb{F}_t[V](x) := f(x, u^{\#}(x)).$$

We can now write out our optimisation problem in terms of these Bellman operators.

$$\begin{cases} V_T = 0, \\ V_t = \mathbb{B}_t[V_{t+1}], \ \forall t < T. \end{cases}$$

We wish to compute V_0 for an initial state x_0.

We define the following Bellman operator

$$\mathbb{B}_t[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} \gamma^t C(x,u) + V(f(x,u)),$$

and "future-state" operator

$$\mathbb{F}_t[V](x) := f(x, u^{\#}(x)).$$

We can now write out our optimisation problem in terms of these Bellman operators.

$$\begin{cases} V_T = 0, \\ V_t = \mathbb{B}_t[V_{t+1}], \ \forall t < T. \end{cases}$$

We wish to compute V_0 for an initial state x_0.

We define the following Bellman operator

$$\mathbb{B}_t[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} \gamma^t C(x,u) + V(f(x,u)),$$

and "future-state" operator

$$\mathbb{F}_t[V](x) := f(x, u^{\#}(x)).$$

We can now write out our optimisation problem in terms of these Bellman operators.

$$\begin{cases} V_T = 0, \\ V_t = \mathbb{B}_t[V_{t+1}], \ \forall t < T. \end{cases}$$

We wish to compute V_0 for an initial state x_0.

Negatives:

- Can be difficult to control the truncation error
- Failure to exploit self-similarity of the formulation

Positives:

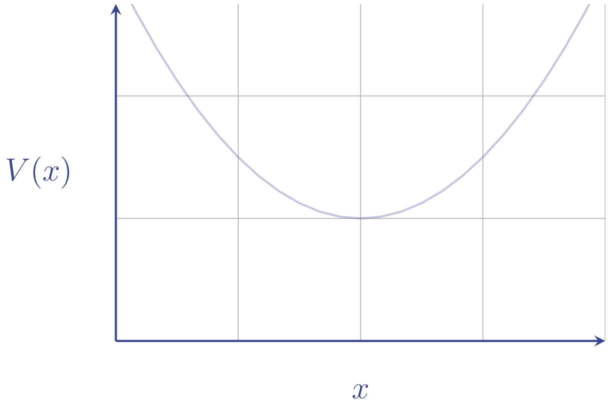
- Direct compatibility with existing SDDP algorithms

Negatives:

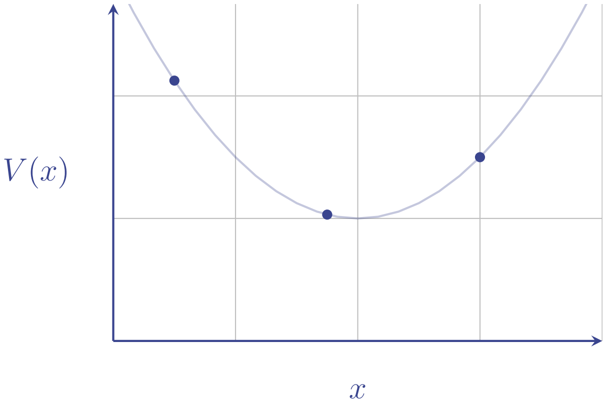
- Can be difficult to control the truncation error
- Failure to exploit self-similarity of the formulation

Positives:

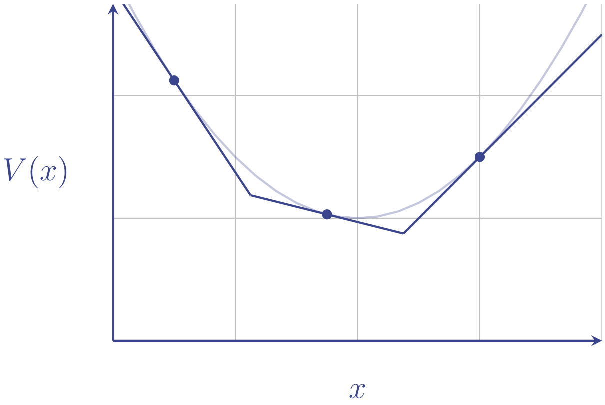
- Direct compatibility with existing SDDP algorithms



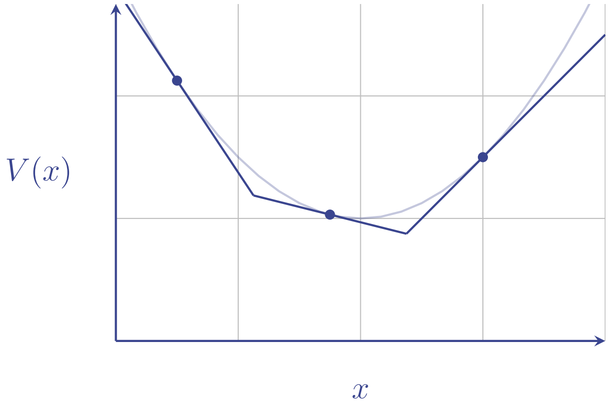
$V(x)$
x



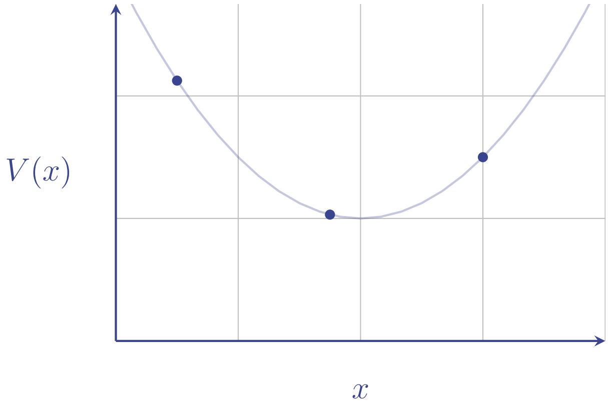
$V(x)$
x

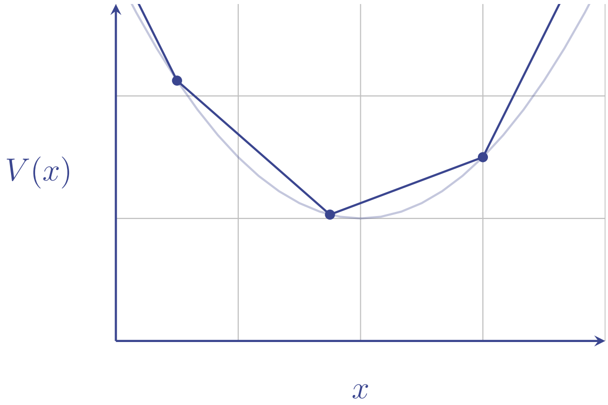


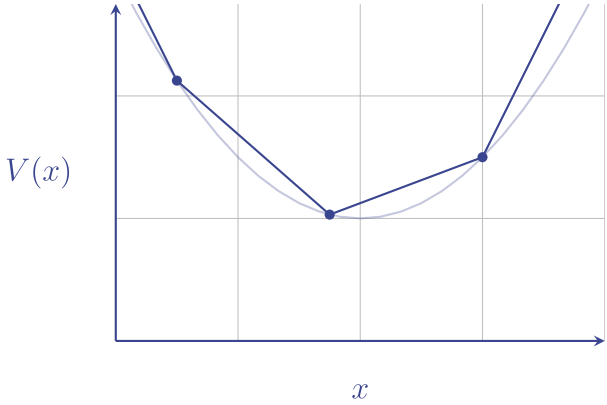
$V(x)$
x



$$\underline{V}(x) = \min_{\mu \in \mathbb{R}} \quad \mu$$
$$\text{s.t.} \quad \mu \geq V(\hat{x}) + \langle d_{\hat{x}}, x - \hat{x} \rangle, \ \forall \hat{x},$$


$V(x)$
x


$V(x)$
x



$$\bar{V}(x) = \min_{\mu \in \mathbb{R}, \lambda \in \mathbb{R}^x} \quad \mu + \langle \lambda, x \rangle$$
$$\text{s.t.} \quad \mu + \langle \lambda, \hat{x} \rangle \geq V(\hat{x}), \ \forall \hat{x}.$$
$$||\lambda||_* \leq \alpha.$$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k)$, $\forall t$
4. set $k \rightarrow k + 1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k)$, $\forall t$
4. set $k \to k + 1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$

2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$

3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$

4. set $k \to k+1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \le V_t \le \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k+1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k+1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \le V_t \le \bar{V}_t^k$

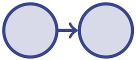
For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k+1$

Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

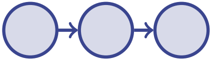
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

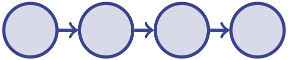
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \le V_t \le \bar{V}_t^k$

For a given iteration k:

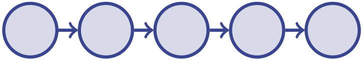
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \le V_t \le \bar{V}_t^k$

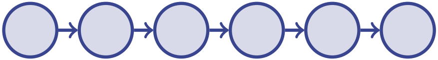
For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$

2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$

3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$

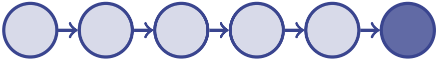
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

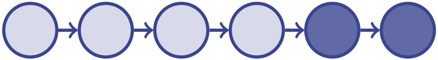
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

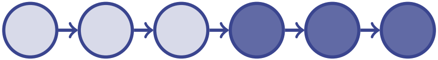
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

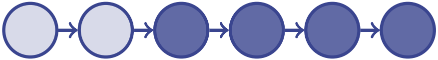
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

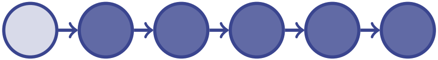
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \rightarrow k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

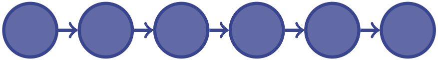
1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k + 1$



Outline of DDP, which refines bounding functions $\underline{V}_t^k \leq V_t \leq \bar{V}_t^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T$ using $x_{t+1}^k \in \mathbb{F}_t[\underline{V}_{t+1}^{k-1}](x_t^k)$
2. update $\underline{V}_t^k$ using value of $\mathbb{B}_t[\underline{V}_{t+1}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}_{t+1}^k]}{\partial x}\right|_{x_t^k}, \ \forall t$
3. update $\bar{V}_t^k$ using value of $\mathbb{B}_t[\bar{V}_{t+1}^k](x_t^k), \ \forall t$
4. set $k \to k+1$



Theorem

$$\lim_{k\to\infty} \bar{V}_t^k(x_t^k) - \underline{V}_t^k(x_t^k) = 0,\ \forall t \leq T$$

Proof.

1. Value functions are Lipschitz-continuous (relies on a finiteness of T)
2. Bounding functions have nice properties

□

Theorem

$$\lim_{k\to\infty} \bar{V}_t^k(x_t^k) - \underline{V}_t^k(x_t^k) = 0, \ \forall t \leq T$$

Proof.

1. Value functions are Lipschitz-continuous (relies on a finiteness of T)
2. Bounding functions have nice properties

$\square$

We define the following Bellman operator

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

and "future-state" operator

$$\mathbb{F}[V](x) := f(x, u^{\#}(x)).$$

We can now write out our optimisation problem in terms of these Bellman operators.

$$V = \mathbb{B}[V].$$

We wish to compute V for an initial state x_0.

We define the following Bellman operator

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

and "future-state" operator

$$\mathbb{F}[V](x) := f(x, u^{\#}(x)).$$

We can now write out our optimisation problem in terms of these Bellman operators.

$$V = \mathbb{B}[V].$$

We wish to compute V for an initial state x_0.

We define the following Bellman operator

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

and "future-state" operator

$$\mathbb{F}[V](x) := f(x, u^{\#}(x)).$$

We can now write out our optimisation problem in terms of these Bellman operators.

$$V = \mathbb{B}[V].$$

We wish to compute V for an initial state x_0.

If we hope to use a cutting plane algorithm to solve this optimisation problem, we need to ensure that V is Lipschitz continuous!

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

It is reasonable to assume that if all the components of the Bellman operator are Lipschitz in some sense, that V ought to be Lipschitz also.

If we hope to use a cutting plane algorithm to solve this optimisation problem, we need to ensure that V is Lipschitz continuous!

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

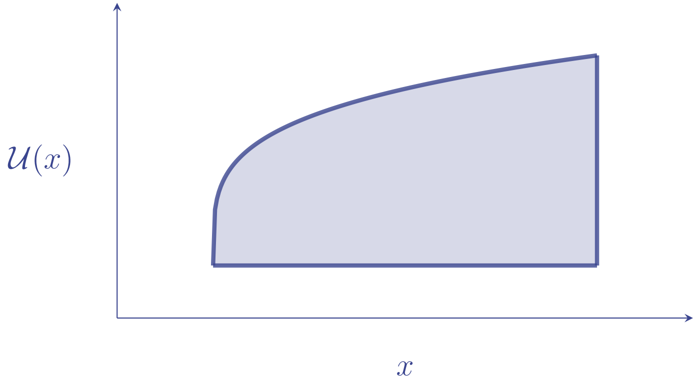
It is reasonable to assume that if all the components of the Bellman operator are Lipschitz in some sense, that V ought to be Lipschitz also.

Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$

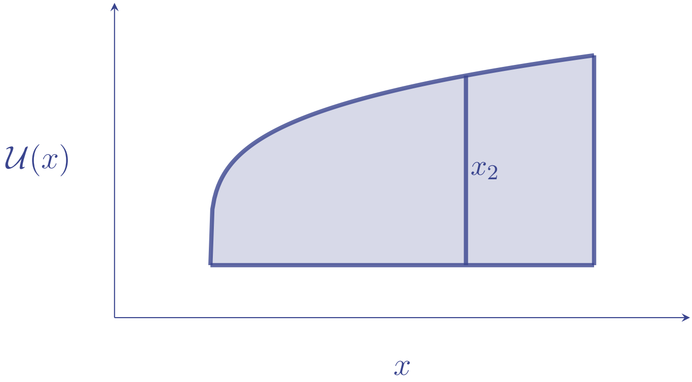
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



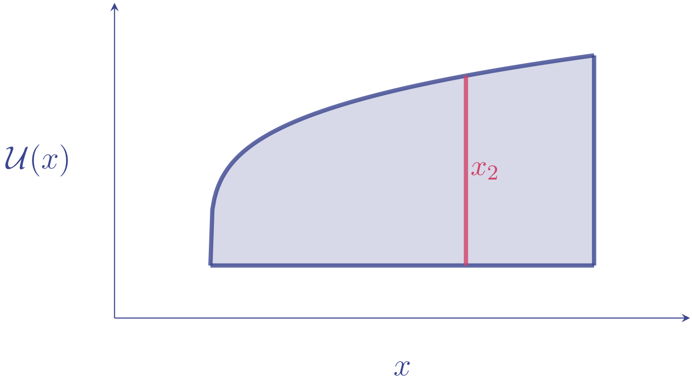
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa ||x_1 - x_2|| \mathsf{Ball}_{||\cdot||}, \ \forall x_1, x_2 \in \mathcal{X}$$



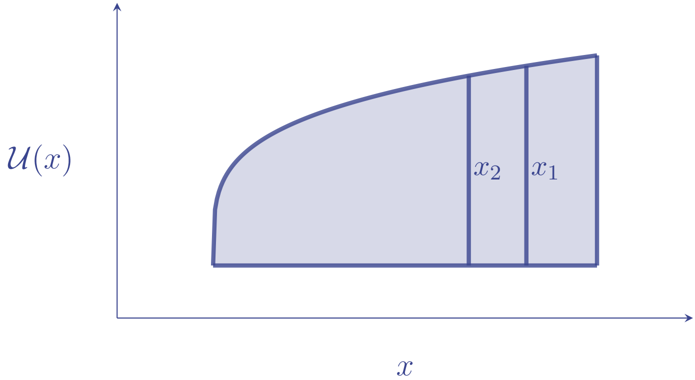
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



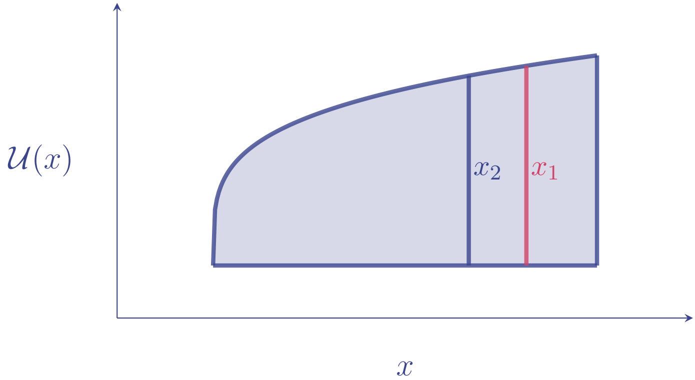
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



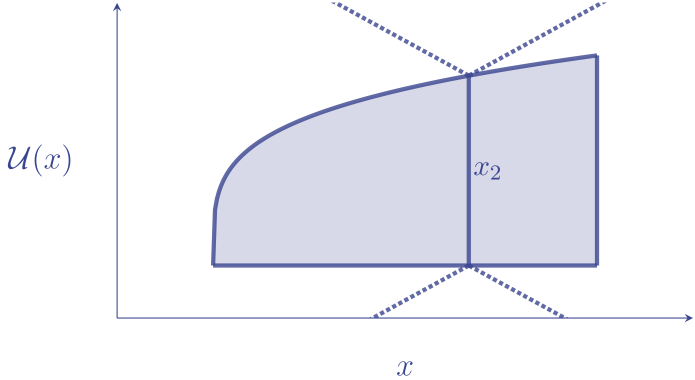
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



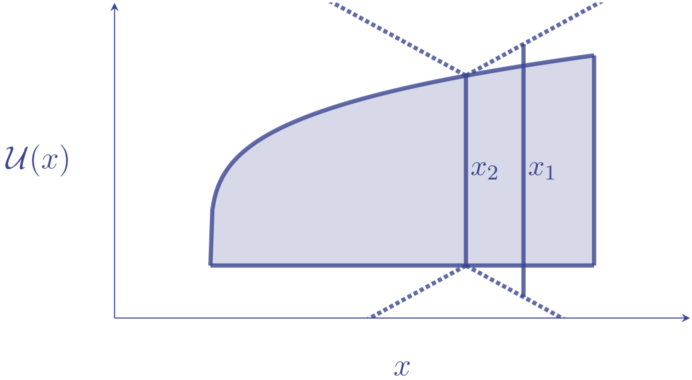
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



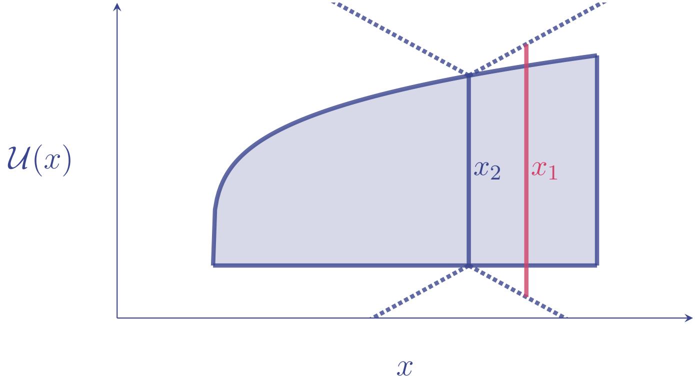
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



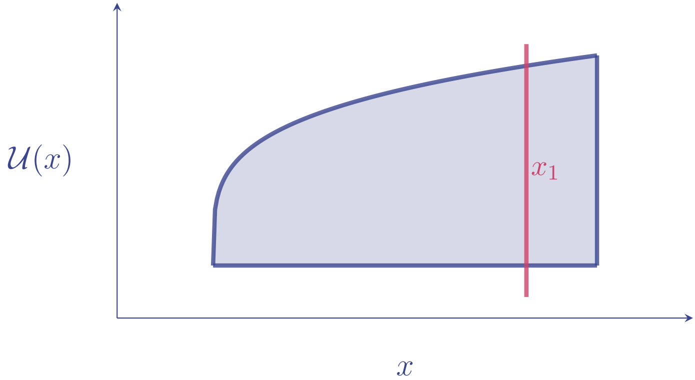
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||.||},\ \forall x_1, x_2 \in \mathcal{X}$$



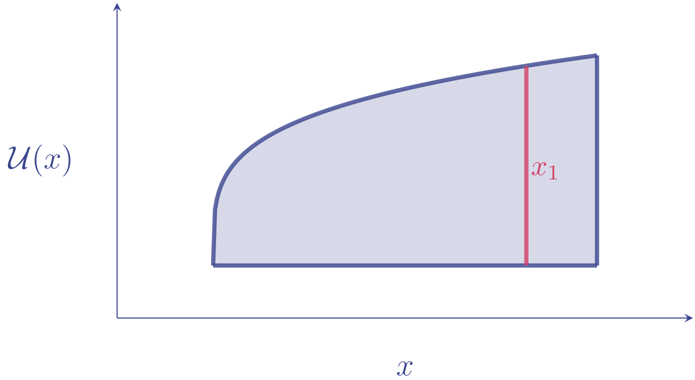
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\text{Ball}_{||.||},\ \forall x_1, x_2 \in \mathcal{X}$$



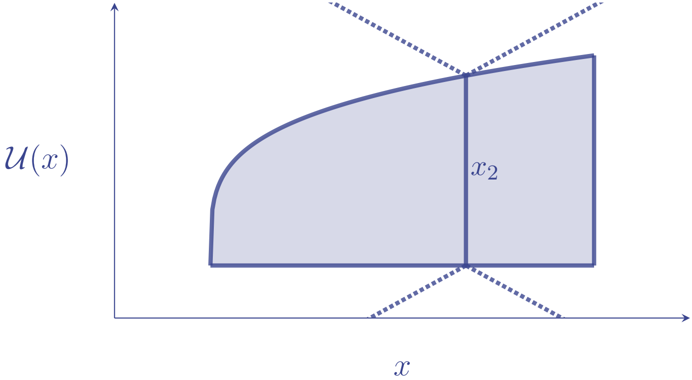
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



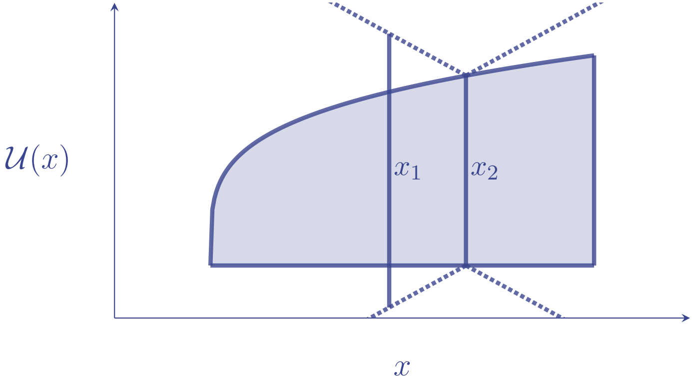
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa ||x_1 - x_2|| \mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



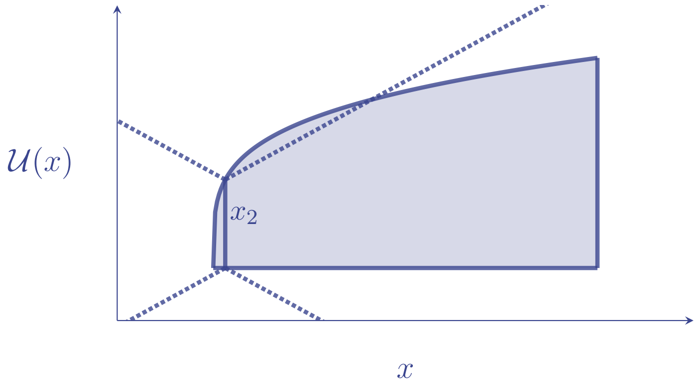
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



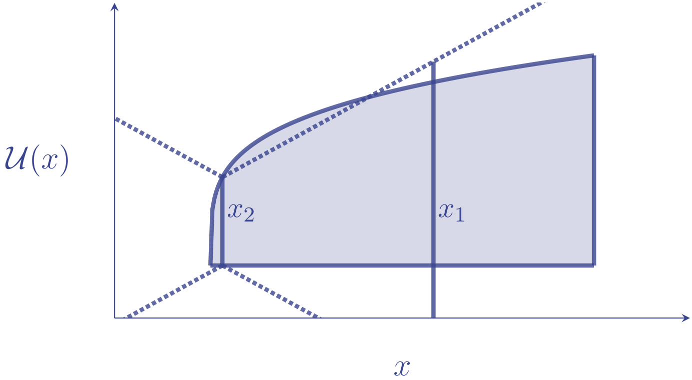
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



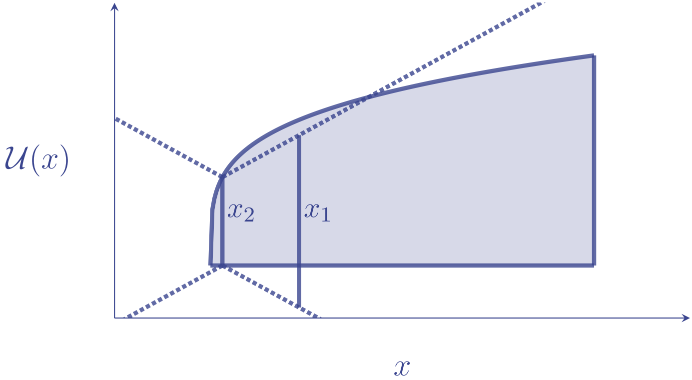
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||}, \ \forall x_1, x_2 \in \mathcal{X}$$



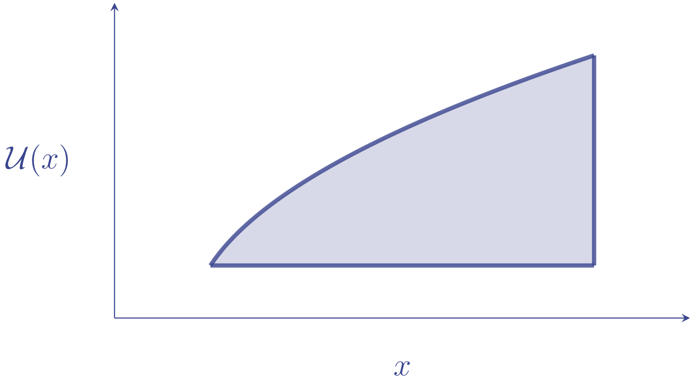
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



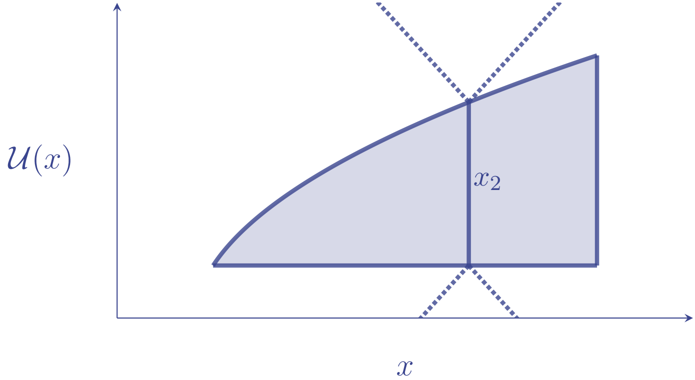
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



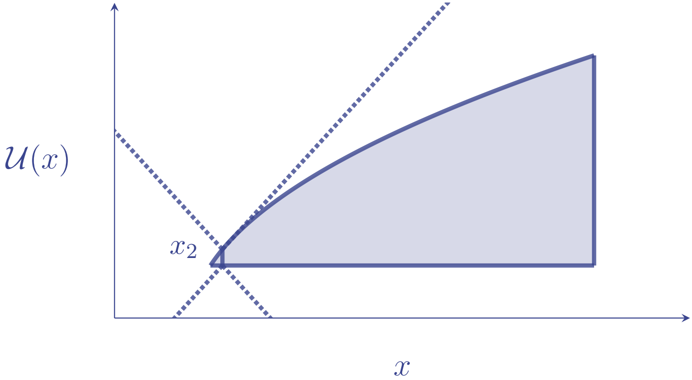
Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa ||x_1 - x_2|| \mathsf{Ball}_{||\cdot||}, \ \forall x_1, x_2 \in \mathcal{X}$$



Lipschitz continuity of a multifunction $\mathcal{U} : \mathbb{R}^x \rightrightarrows \mathbb{R}^u$, R. Wets 2002.

$$\mathcal{U}(x_1) \subseteq \mathcal{U}(x_2) + \kappa||x_1 - x_2||\mathsf{Ball}_{||\cdot||},\ \forall x_1, x_2 \in \mathcal{X}$$



$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

Lemma (K. Hinderer 2005)

Suppose $\mathcal{U}, C$ *and* f *have Lipschitz constants* $L^{\mathcal{U}}, L^{C}, L^{f}$*, respectively. Then*

$$\textit{Lip}\big(\mathbb{B}(H)\big) \leq L^{C}(1 + L^{\mathcal{U}}) + \gamma L^{f}(1 + L^{\mathcal{U}})\textit{Lip}(H)$$

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

Corollary

Suppose $\mathcal{U}, C$ *and* f *have Lipschitz constants* $L^{\mathcal{U}}, L^{C}, L^{f}$*, respectively. If* $\gamma L^{f}(1 + L^{\mathcal{U}}) < 1$ *then*

$$\lim_{n \to \infty} \mathit{Lip}\big(\mathbb{B}^n(0)\big) \leq \frac{L^{C}(1 + L^{\mathcal{U}})}{1 - \gamma L^{f}(1 + L^{\mathcal{U}})}$$

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x),\\ f(x,u) \in \mathcal{X}}} C(x,u) + \gamma \times V(f(x,u)),$$

Lemma (K. Hinderer 2005)

Suppose $\mathcal{U}, C$ *and* f *have Lipschitz constants* $L^{\mathcal{U}}, L^C, L^f$*, respectively. If* $\gamma L^f(1 + L^{\mathcal{U}}) < 1$*, then* V *is Lipschitz-continuous on* $\mathcal{X}$ *with Lipschitz constant bounded above by*

$$L^V \leq \frac{L^C(1 + L^{\mathcal{U}})}{1 - \gamma L^f(1 + L^{\mathcal{U}})}$$

Similar to DDP, we are going to generate a sequence of functions $\underline{V}^k \leq V \leq \bar{V}^k$.

Question: Where should we compute cuts?

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The ‘Sticky’ algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

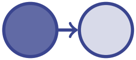
For a given iteration k:

1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$

The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

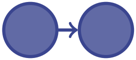
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k + 1$



The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

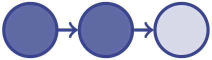
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

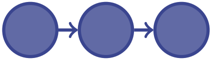
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

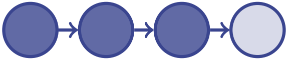
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

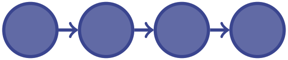
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



The 'Sticky' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

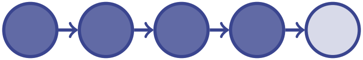
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



The ‘Sticky’ algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

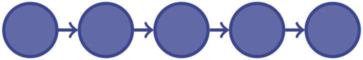
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k + 1$



The ‘Sticky’ algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

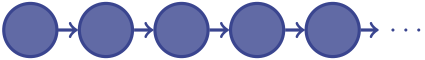
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



The ‘Sticky’ algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

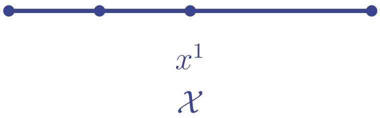
1. generate new point using $x^k \in \mathbb{F}[\underline{V}^{k-1}](x^{k-1})$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^{k-1}](x^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^{k-1}]}{\partial x}\right|_{x^k}$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^{k-1}](x^k)$,
4. set $k \leftarrow k+1$



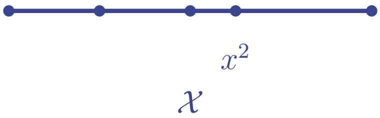
Problem: if x^k converges, then our algorithm will get stuck.

Problem: if x^k converges, then our algorithm will get stuck.

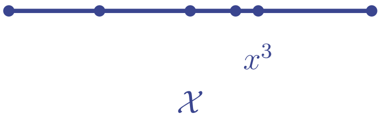
Problem: if x^k converges, then our algorithm will get stuck.



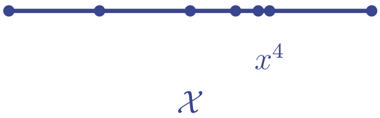
Problem: if x^k converges, then our algorithm will get stuck.



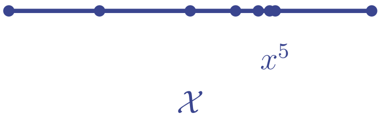
Problem: if x^k converges, then our algorithm will get stuck.



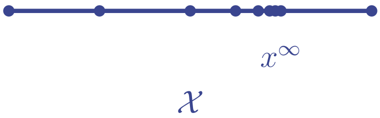
Problem: if x^k converges, then our algorithm will get stuck.



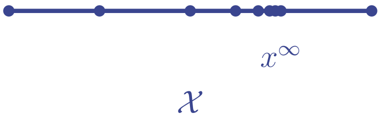
Problem: if x^k converges, then our algorithm will get stuck.



Problem: if x^k converges, then our algorithm will get stuck.



Problem: if x^k converges, then our algorithm will get stuck.



This begs the question, what type of convergence are we looking for?

It would be nice to have convergence on points generated by the policy:

$$\lim_{k\to\infty} \bar{V}^k(x_t^k) - \underline{V}^k(x_t^k) = 0, \forall t \in \mathbb{N}.$$

The key to this type of convergence is to start from x_0 at each iteration and generate longer and longer state trajectories as $k \to \infty$. Let T_k be how 'far' we look out at iteration k.

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

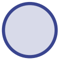
$$T_1 = 1$$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_1 = 1$$

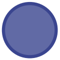


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_1 = 1$$



The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

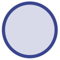
$$T_2 = 2$$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_2 = 2$

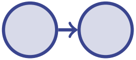


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_2 = 2$$

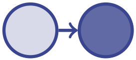


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_2 = 2$

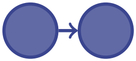


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_2 = 2$



The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

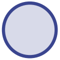
$$T_3 = 3$$

The ‘Yo-yo’ algorithm, refining bounding functions $\underline{V}^k \le V \le \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_3 = 3$$

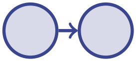


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_3 = 3$$

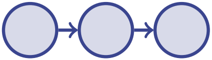


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_3 = 3$$

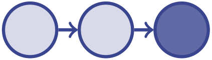


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_3 = 3$$

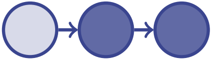


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_3 = 3$

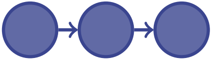


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_3 = 3$



The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

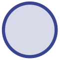
$$T_4 = 4$$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_4 = 4$$

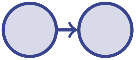


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_4 = 4$$

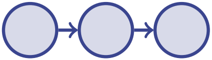


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_4 = 4$$

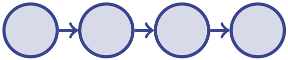


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_4 = 4$

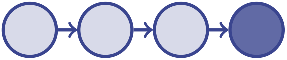


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_4 = 4$

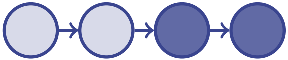


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_4 = 4$

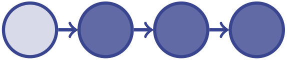


The ‘Yo-yo’ algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_4 = 4$

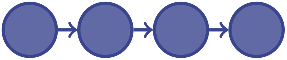


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_4 = 4$$



The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

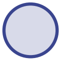
$$T_5 = 5$$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_5 = 5$$

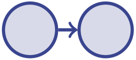


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_5 = 5$

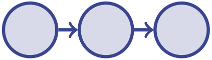


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_5 = 5$$

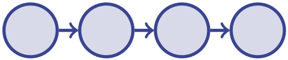


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_5 = 5$$

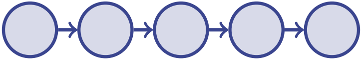


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_5 = 5$$

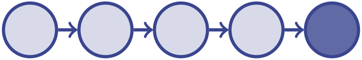


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_5 = 5$

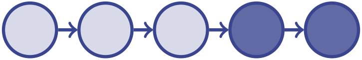


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_5 = 5$



The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

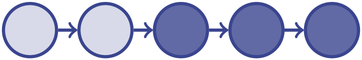
1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$

2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$

3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$

4. set $k \leftarrow k + 1$

$$T_5 = 5$$

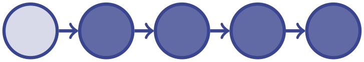


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

$$T_5 = 5$$

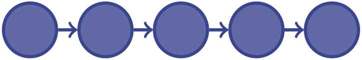


The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

$T_5 = 5$



Theorem

$$\limsup_{k\to\infty} T_k = \infty \implies \lim_{k\to\infty} \bar{V}^k(x_t^k) - \underline{V}^k(x_t^k),\ \forall t \in \mathbb{N}.$$

Proof.

1. We are always looking further and further ahead and $\gamma \in [0, 1)$.
2. Value functions are Lipschitz-continuous
3. Bounding functions have nice properties

□

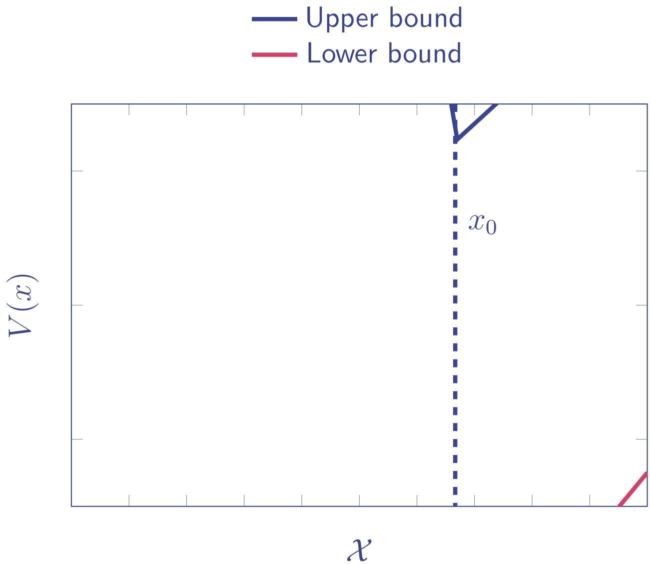
Theorem

$$\limsup_{k\to\infty} T_k = \infty \implies \lim_{k\to\infty} \bar{V}^k(x_t^k) - \underline{V}^k(x_t^k),\ \forall t \in \mathbb{N}.$$

Proof.

1. We are always looking further and further ahead and $\gamma \in [0, 1)$.
2. Value functions are Lipschitz-continuous
3. Bounding functions have nice properties

$\square$


Upper bound
Lower bound
x_0
$V(x)$
$\mathcal{X}$

Now we have a discrete, independent noise in the dynamics, $\omega \in \Omega$
We define a new Bellman operator

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f_\omega(x,u) \in \mathcal{X}}} C_n(x,u) + \gamma \sum_{\omega \in \Omega} P(\omega) V(f_\omega(x,u)),$$

With a new 'future-state' operator

$$\mathbb{F}[V](x,\omega) := f_\omega(x, u^\#(x))$$

We seek a function

$$V = \mathbb{B}[V]$$

and to compute it at $V(x^0)$.

Now we have a discrete, independent noise in the dynamics, $\omega \in \Omega$
We define a new Bellman operator

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f_\omega(x,u) \in \mathcal{X}}} C_n(x,u) + \gamma \sum_{\omega \in \Omega} P(\omega) V(f_\omega(x,u)),$$

With a new 'future-state' operator

$$\mathbb{F}[V](x,\omega) := f_\omega(x, u^{\#}(x))$$

We seek a function

$$V = \mathbb{B}[V]$$

and to compute it at $V(x^0)$.

Now we have a discrete, independent noise in the dynamics, $\omega \in \Omega$
We define a new Bellman operator

$$\mathbb{B}[V](x) := \min_{\substack{u \in \mathcal{U}(x), \\ f_\omega(x,u) \in \mathcal{X}}} C_n(x,u) + \gamma \sum_{\omega \in \Omega} P(\omega) V(f_\omega(x,u)),$$

With a new 'future-state' operator

$$\mathbb{F}[V](x,\omega) := f_\omega(x, u^{\#}(x))$$

We seek a function

$$V = \mathbb{B}[V]$$

and to compute it at $V(x^0)$.

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

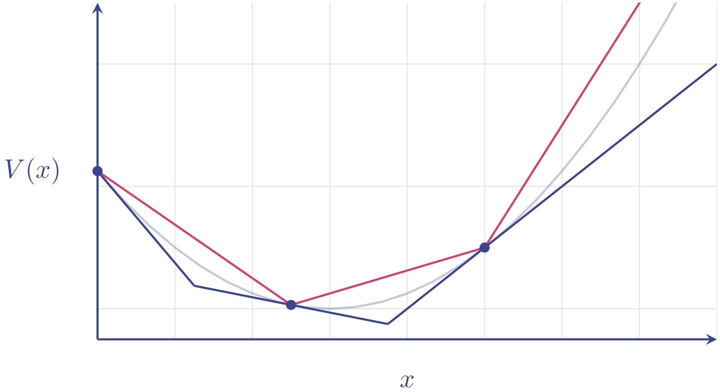
For a given iteration k:

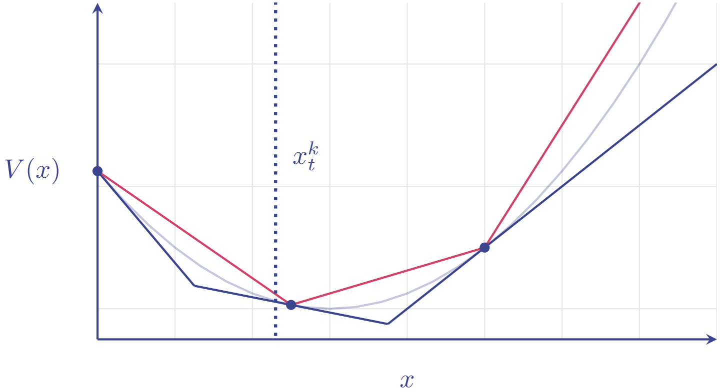
1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
4. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

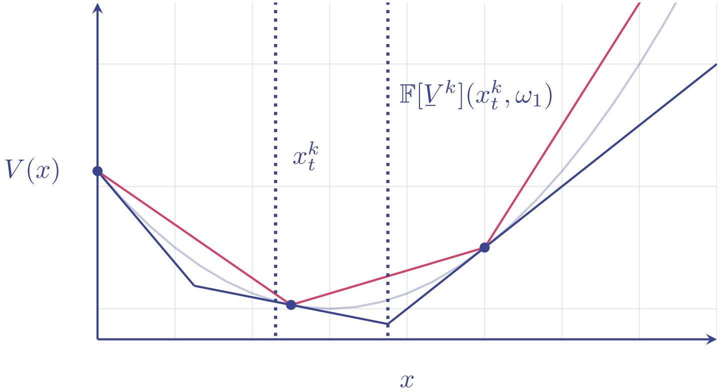
For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega)$
2. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
3. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
4. set $k \leftarrow k + 1$

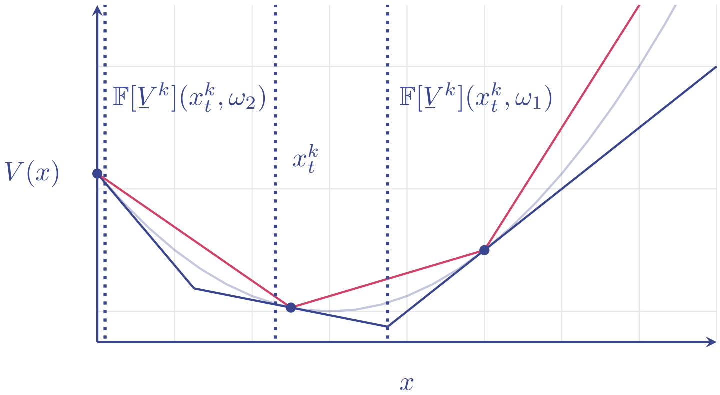

$V(x)$
x



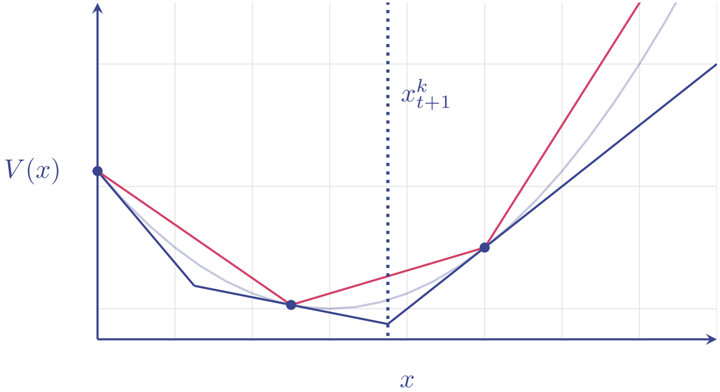
$V(x)$
x_t^k
x



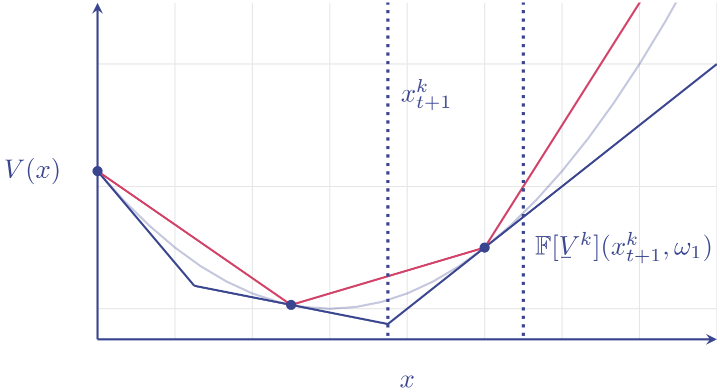
$\mathbb{F}[\underline{V}^k](x_t^k, \omega_1)$
x_t^k
$V(x)$
x

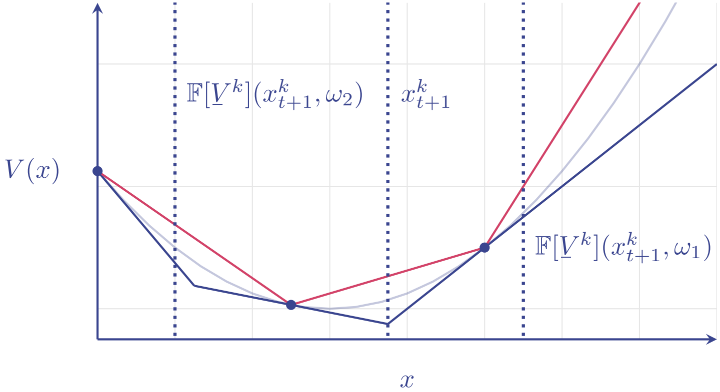


$\mathbb{F}[\underline{V}^k](x_t^k, \omega_2)$
$\mathbb{F}[\underline{V}^k](x_t^k, \omega_1)$
x_t^k
$V(x)$
x



x_{t+1}^k
$V(x)$
x


x_{t+1}^k
$V(x)$
$\mathbb{F}[\underline{V}^k](x_{t+1}^k, \omega_1)$
x



$\mathbb{F}[\underline{V}^k](x^k_{t+1}, \omega_2)$
x^k_{t+1}
$V(x)$
$\mathbb{F}[\underline{V}^k](x^k_{t+1}, \omega_1)$
x

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega^*)$
2. where $\omega^* = \arg\max_{\omega} \bar{V}^k(\mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega)) - \underline{V}^k(\mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega))$
3. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
4. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
5. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega^*)$
2. where $\omega^* = \arg\max_{\omega} \bar{V}^k(\mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega)) - \underline{V}^k(\mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega))$
3. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
4. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
5. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega^*)$
2. where $\omega^* = \arg\max_{\omega} \bar{V}^k(\mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega)) - \underline{V}^k(\mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega))$
3. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
4. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
5. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x_{t+1}^k \in \mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega^*)$
2. where $\omega^* = \arg\max_{\omega} \bar{V}^k(\mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega)) - \underline{V}^k(\mathbb{F}[\underline{V}^{k-1}](x_t^k, \omega))$
3. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x_t^k)$ and $\left.\frac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x_t^k}$, $\forall t$
4. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x_t^k)$, $\forall t$
5. set $k \leftarrow k + 1$

The 'Yo-yo' algorithm, refining bounding functions $\underline{V}^k \leq V \leq \bar{V}^k$

For a given iteration k:

1. generate state trajectory from $t = 0$ to $t = T_k$ using $x^k_{t+1} \in \mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega^*)$
2. where $\omega^* = \arg\max_\omega \bar{V}^k(\mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega)) - \underline{V}^k(\mathbb{F}[\underline{V}^{k-1}](x^k_t, \omega))$
3. update $\underline{V}^k$ using value of $\mathbb{B}[\underline{V}^k](x^k_t)$ and $\left.\dfrac{\partial \mathbb{B}_t[\underline{V}^k]}{\partial x}\right|_{x^k_t}$, $\forall t$
4. update $\bar{V}^k$ using value of $\mathbb{B}[\bar{V}^k](x^k_t)$, $\forall t$
5. set $k \leftarrow k + 1$

Theorem

$$\limsup_{k\to\infty} T_k = \infty \implies$$
$$\lim_{k\to\infty} \bar{V}^k(x_t^k(\omega)) - \underline{V}^k(x_t^k(\omega)), \ \forall t \in \mathbb{N}, \forall \omega \in \{\Omega\}^t.$$

Proof.

1. Previous theorem
2. Always focusing on the 'scenario' with the largest gap.

□

Theorem

$$\limsup_{k\to\infty} T_k = \infty \implies$$
$$\lim_{k\to\infty} \bar{V}^k(x_t^k(\omega)) - \underline{V}^k(x_t^k(\omega)),\ \forall t \in \mathbb{N}, \forall \omega \in \{\Omega\}^t.$$

Proof.

1. Previous theorem
2. Always focusing on the 'scenario' with the largest gap.

$\square$

Conclusions

- Verify the conditions of Lipschitz continuity of V
- Adapting SDDP to the infinite horizon case
- Proof of convergence

R. Baucke, An algorithm for solving infinite horizon Markov dynamic programmes. *www.optimization-online.org* , 2018.

R. Wets. Lipschitz Continuity of inf-Projections. *Computational Optimization and Applications*, 2002.

K. Hinderer. Lipschitz continuity of value functions in Markovian decision processes. *Mathematical Methods of Operations Research*, 2005.

Conclusions

- Verify the conditions of Lipschitz continuity of V
- Adapting SDDP to the infinite horizon case
- Proof of convergence

R. Baucke, An algorithm for solving infinite horizon Markov dynamic programmes. *www.optimization-online.org* , 2018.

R. Wets. Lipschitz Continuity of inf-Projections. *Computational Optimization and Applications*, 2002.

K. Hinderer. Lipschitz continuity of value functions in Markovian decision processes. *Mathematical Methods of Operations Research*, 2005.

Conclusions

- Verify the conditions of Lipschitz continuity of V
- Adapting SDDP to the infinite horizon case
- Proof of convergence

R. Baucke, An algorithm for solving infinite horizon Markov dynamic programmes. *www.optimization-online.org* , 2018.

R. Wets. Lipschitz Continuity of inf-Projections. *Computational Optimization and Applications*, 2002.

K. Hinderer. Lipschitz continuity of value functions in Markovian decision processes. *Mathematical Methods of Operations Research*, 2005.

Conclusions

- Verify the conditions of Lipschitz continuity of V
- Adapting SDDP to the infinite horizon case
- Proof of convergence

R. Baucke, An algorithm for solving infinite horizon Markov dynamic programmes. *www.optimization-online.org* , 2018.

R. Wets. Lipschitz Continuity of inf-Projections. *Computational Optimization and Applications*, 2002.

K. Hinderer. Lipschitz continuity of value functions in Markovian decision processes. *Mathematical Methods of Operations Research*, 2005.