A discrete time DP approach on a tree structure for finite horizon optimal control problems

Maurizio Falcone

joint works with A. Alla (PUC, Rio) and L. Saluzzi (GSSI, L’Aquila)

ICODE Workshop "Numerical Solution of HJB Equations"

Paris VII, January 9, 2020
HJB equation for the finite horizon problem

Controlled Dynamics and Cost Functional

\[
\begin{aligned}
\dot{y}(s, u) &= f(y(s), u(s), s) \quad s \in (t, T] \\
y(t) &= x
\end{aligned}
\]

\[u(t) \in U = \{ u : [t, T] \rightarrow U \subset \mathbb{R}^m \text{ compact, measurable} \},\]

\[J_{x,t}(u) = \int_t^T L(y(s, u), u(s), s)e^{-\lambda(s-t)} \, ds + g(y(T))e^{-\lambda(T-t)}\]
HJB equation for the finite horizon problem

Controlled Dynamics and Cost Functional

\[
\begin{cases}
\dot{y}(s, u) = f(y(s), u(s), s) & s \in (t, T] \\
y(t) = x
\end{cases}
\]

\[u(t) \in \mathcal{U} = \{u : [t, T] \to U \subset \mathbb{R}^m \text{ compact, measurable}\},\]

\[J_{x,t}(u) = \int_t^T L(y(s, u), u(s), s)e^{-\lambda(s-t)} \, ds + g(y(T))e^{-\lambda(T-t)}\]

Value Function

\[v(x, t) := \inf_{u(\cdot) \in \mathcal{U}} J_{x,t}(u)\]
HJB equation for the finite horizon problem

\[ v(x, t) = \min_{u \in \mathcal{U}} \left\{ \int_t^\tau e^{-\lambda(s-t)} L(y(s), u(s), s) \, ds + v(y(\tau), \tau) e^{-\lambda(\tau-t)} \right\} \]
HJB equation for the finite horizon problem

\[ \nu(x, t) = \min_{u \in U} \left\{ \int_t^\tau e^{-\lambda(s-t)} L(y(s), u(s), s) \, ds + \nu(y(\tau), \tau) e^{-\lambda(\tau-t)} \right\} \]

Dynamic Programming Principle

HJB equation

\[ \begin{cases} 
- \frac{\partial \nu}{\partial t}(x, t) + \lambda \nu(x, t) = \min_{u \in U} \{ L(x, u, t) + \nabla \nu(x, t) \cdot f(x, u, t) \} \\
\nu(x, T) = g(x), x \in \mathbb{R}^d
\end{cases} \]
HJB equation for the finite horizon problem

**Dynamic Programming Principle**

\[ v(x, t) = \min_{u \in U} \left\{ \int_{t}^{T} e^{-\lambda(s-t)} L(y(s), u(s), s) \, ds + v(y(T), \tau) e^{-\lambda(\tau-t)} \right\} \]

**HJB equation**

\[
\begin{cases}
-\frac{\partial v}{\partial t}(x, t) + \lambda v(x, t) = \min_{u \in U} \{ L(x, u, t) + \nabla v(x, t) \cdot f(x, u, t) \} \\
v(x, T) = g(x), \, x \in \mathbb{R}^d
\end{cases}
\]

**Optimal Feedback Map**

\[ u^*(x, t) = \arg \min_{u \in U} \{ L(x, u, t) + \nabla v(x, t) \cdot f(x, u, t) \} \]
Classical approach

Semi-Lagrangian scheme ($\lambda = 0$)

\[
\begin{align*}
V_{i}^{n-1} &= \min_{u \in U} [\Delta t \, L(x_i, u, t_n) + V^n(x_i + \Delta t \, f(x_i, u, t_n))] , \ n = N, \ldots, 1 \\
V_{i}^{N} &= g(x_i) , \quad x_i \in \Omega^{\Delta x} .
\end{align*}
\]
Classical approach

Semi-Lagrangian scheme ($\lambda = 0$)

\[
\begin{align*}
V_{i}^{n-1} &= \min_{u \in U}[\Delta t \, L(x_i, u, t_n) + V^n(x_i + \Delta t \, f(x_i, u, t_n))] \quad n = N, \ldots, 1 \\
V_i^N &= g(x_i) \quad x_i \in \Omega^{\Delta x}.
\end{align*}
\]

Cons of the approach

- $V^n(x_i + \Delta t \, f(x_i, u, t_n))$ is computed by interpolation operator.
- We need a **numerical domain** (not always given in the problem)
- Selection of **boundary conditions** (not always given in the problem)
- The **curse of dimensionality** makes the problem difficult to solve in high dimension (need e.g. model order reduction).
Other approaches and acceleration techniques

Several methods have been developed to accelerate the computation and/or mitigate the curse of dimensionality


Other approaches and acceleration techniques

- **Model Order Reduction**: Kunisch-Volkwein-Xie (2004), Alla-F-Volkwein (2017)
- **Sparse grids**: Bokanowski-Garke-Griebel-Klompmaker (2013), Garke-Kroner (2016)
- **DNN/DGM**: Pham-Warin (2019)
We start with an initial condition $x \in \mathbb{R}^d$ forming the first level $\mathcal{T}^0$. 

$x$
We start with an initial condition \( x \in \mathbb{R}^d \) forming the first level \( \mathcal{T}^0 \).

Discretization: constant \( \Delta t \) for time and \( N_u \) discrete controls.
We start with an initial condition $x \in \mathbb{R}^d$ forming the first level $T^0$.

**Discretization**: constant $\Delta t$ for time and $N_u$ discrete controls.

Starting with $x$, we follow the dynamics given by the discrete controls

$$T^1 = \{ \zeta^1_i \}_{i} = \{ x + \Delta t f(x, u_i, t_0) \}_{i}, \quad i = 1, \ldots, N_u$$
Given the nodes in the previous level, we construct the following one

\[ \mathcal{T}^n = \{ \zeta_i^{n-1} + \Delta t f(\zeta_i^{n-1}, u_j, t_{n-1}) \}_{j=1}^{N_u} \quad i = 1, \ldots, N_u^N. \]
Approximation of the value function

Computation of the value function on the tree

The tree structure defines $\mathcal{T} = \{\mathcal{T}^r\}_{r=0}^{\bar{N}}$, where we can compute the numerical value function:

$$
\begin{align*}
V^n(\zeta^n_i) &= \min_{u \in U} \left\{ V^{n+1}(\zeta^n_i + \Delta t f(\zeta^n_i, u, t_n)) + \Delta t L(\zeta^n_i, u, t_n) \right\} & \zeta^n_i \in \mathcal{T}^n \\
V^{\bar{N}}(\zeta^{\bar{N}}_i) &= g(\zeta^{\bar{N}}_i) & \zeta^{\bar{N}}_i \in \mathcal{T}^{\bar{N}}
\end{align*}
$$
Approximation of the value function

Computation of the value function on the tree

The tree structure defines $\mathcal{T} = \{\mathcal{T}^r\}_{r=0}^N$, where we can compute the numerical value function:

$$
\begin{cases}
V^n(\zeta^n_i) = \min_{u \in U^{\Delta u}} \{ V^{n+1}(\zeta^n_i + \Delta t f(\zeta^n_i, u, t_n)) + \Delta t L(\zeta^n_i, u, t_n) \} & \zeta^n_i \in \mathcal{T}^n \\
V^N(\zeta^N_i) = g(\zeta^N_i) & \zeta^N_i \in \mathcal{T}^N
\end{cases}
$$

Pros

- No need for interpolation since the nodes $x_i + \Delta t f(x_i, u, t_n)$ belong to the tree by construction.
- Mitigation of the curse of dimensionality (e.g., $d \gg 10$).
Approximation of the value function

Computation of the value function on the tree

The tree structure defines \( \mathcal{T} = \{\mathcal{T}^r\}_{r=0}^{N} \), where we can compute the numerical value function:

\[
\begin{align*}
V^n(\zeta^n_i) &= \min_{u \in U^{\Delta u}} \left\{ V^{n+1}(\zeta^n_i + \Delta t f(\zeta^n_i, u, t_n)) + \Delta t L(\zeta^n_i, u, t_n) \right\} \quad \zeta^n_i \in \mathcal{T}^n \\
V^N(\zeta^N_i) &= g(\zeta^N_i) \quad \zeta^N_i \in \mathcal{T}^N
\end{align*}
\]

Pros

- No need for interpolation since the nodes \( x_i + \Delta t f(x_i, u, t_n) \) belong to the tree by construction.
- Mitigation of the curse of dimensionality (e.g., \( d \gg 10 \)).

Cons

- Dimensionality problem. In fact, given \( N_u \) controls and \( N \) time steps, the cardinality of the tree is \( O(N_u^{N+1}) \).
Solution: Pruning the tree
Solution: Pruning the tree

Pruning rule
Given a threshold $\varepsilon_T$, two nodes $\zeta^n_i$ and $\zeta^n_j$ will be merged if

$$\|\zeta^n_i - \zeta^n_j\| \leq \varepsilon_T$$
The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels.

Pruning rule

Given a threshold $\varepsilon_T$, two nodes $\zeta_n^i$ and $\zeta_m^j$ will be merged if $\|\zeta_n^i - \zeta_m^j\| \leq \varepsilon_T$

Computation of the value function on the tree

\[
V_n(\zeta) = \min_{u \in U} \left\{ V_n + 1(\zeta + \Delta t f(\zeta, u)) + \Delta t L(\zeta, u, t_n) \right\}
\]

$\zeta \in \cup_{n,k=0}^{T} \zeta^k$

$V_N(\zeta) = g(\zeta)$ $\zeta \in T$

Important reduction of the cardinality, we can get more information on $V$ and this can be useful for the feedback reconstruction.
The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels.

**Pruning rule**

Given a threshold $\varepsilon_T$, two nodes $\zeta^n_i$ and $\zeta^m_j$ will be merged if

$$\|\zeta^n_i - \zeta^m_j\| \leq \varepsilon_T$$
The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels

**Pruning rule**

Given a threshold $\varepsilon_T$, two nodes $\zeta^n_i$ and $\zeta^m_j$ will be merged if

$$\|\zeta^n_i - \zeta^m_j\| \leq \varepsilon_T$$

**Computation of the value function on the tree**

$$\begin{cases} 
V^n(\zeta) = \min_{u \in U^{\Delta u}} \left\{ V^{n+1}(\zeta + \Delta t f(\zeta, u)) + \Delta t L(\zeta, u, t_n) \right\} & \zeta \in \bigcup_{k=0}^n T^k \\
V^{\overline{N}}(\zeta) = g(\zeta) & \zeta \in T
\end{cases}$$
The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels.

**Pruning rule**

Given a threshold $\varepsilon_T$, two nodes $\zeta_i^n$ and $\zeta_j^m$ will be merged if

$$\|\zeta_i^n - \zeta_j^m\| \leq \varepsilon_T$$

**Computation of the value function on the tree**

$$\begin{cases} 
V^n(\zeta) = \min_{u \in U^{\Delta u}} \left\{ V^{n+1}(\zeta + \Delta t f(\zeta, u)) + \Delta t L(\zeta, u, t_n) \right\} & \zeta \in \bigcup_{k=0}^n \mathcal{T}^k \\
V^N(\zeta) = g(\zeta) & \zeta \in \mathcal{T} 
\end{cases}$$

Important reduction of the cardinality, we can get more information on $V$ and this can be useful for the feedback reconstruction.
Problem

The computation of the distances among all the nodes would be very expensive, especially for high dimensional problems.
Efficient pruning

**Problem**
The computation of the distances among all the nodes would be very expensive, especially for high dimensional problems.

**One possible solution**
We project the data onto a lower dimensional linear space such that the variance of the projected data is maximized. This can be done e.g. computing the Singular Value Decomposition of the data matrix and taking the first basis.
Efficient pruning

Problem
The computation of the distances among all the nodes would be very expensive, especially for high dimensional problems.

One possible solution
We project the data onto a lower dimensional linear space such that the variance of the projected data is maximized. This can be done e.g. computing the Singular Value Decomposition of the data matrix and taking the first basis.

Reduced dynamics
The control problem can be solved in a reduced space, projecting the dynamics via Proper Orthogonal Decomposition.
Efficient pruning II

1. Construction of a rough full tree
2. Computation of the maximum variance direction and its subdivision in buckets of length equal to the tolerance.
3. Construction of the pruned tree comparing the nodes in the same bucket.
Construction of a rough full tree
Construction of a rough full tree

Computation of the maximum variance direction and its subdivision in buckets of length equal to the tolerance.
Efficient pruning II

1. Construction of a rough full tree
2. Computation of the maximum variance direction and its subdivision in buckets of length equal to the tolerance.
3. Construction of the pruned tree comparing the nodes in the same bucket.
Error estimates for the approximate value $V$

**Theorem (F.-Giorgi, ’99)**

Let $f$, $L$ and $g$ be Lipschitz continuous and bounded, then

$$
\sup_{(x,t) \in \mathbb{R}^d \times [0,T]} |v(t, x) - V(t, x)| \leq C(T) \sqrt{\Delta t}.
$$
## Error estimates for the approximate value $V$

<table>
<thead>
<tr>
<th>Theorem (F.-Giorgi, ’99)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $f$, $L$ and $g$ be Lipschitz continuous and bounded, then</td>
</tr>
<tr>
<td>$\sup_{(x,t) \in \mathbb{R}^d \times [0,T]}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (Error estimate: first part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $f$, $L$ and $g$ be Lipschitz continuous and bounded, then</td>
</tr>
<tr>
<td>$\sup_{(x,t) \in \mathbb{R}^d \times [0,T]} (v(t, x) - V(t, x)) \leq C(T) \Delta t$</td>
</tr>
</tbody>
</table>
Error estimates for the approximate value $V$

Theorem (F.-Giorgi, ’99)

Let $f, L$ and $g$ be Lipschitz continuous and bounded, then

$$\sup_{(x,t) \in \mathbb{R}^d \times [0,T]} |v(t, x) - V(t, x)| \leq C(T)\sqrt{\Delta t}.$$ 

Theorem (Error estimate: first part)

Let $f, L$ and $g$ be Lipschitz continuous and bounded, then

$$\sup_{(x,t) \in \mathbb{R}^d \times [0,T]} (v(t, x) - V(t, x)) \leq C(T)\Delta t.$$ 

The opposite inequality is based on the semiconcavity of the approximation $V$, i.e.

$$V(x + z, t + s) - 2V(x, s) + V(x - z, t - s) \leq C(|z|^2 + s^2).$$
Error estimates for the approximate value $V$

**Proposition**

Let $f$, $L$ and $g$ be Lipschitz continuous, bounded. Moreover let $L$ and $g$ be semiconcave and $f \in C^1$. Then the approximate solution $V$ is semiconcave.

**Lemma (Capuzzo-Ishii, '84)**

Let $\xi$ be semiconcave such that $\xi(0,0) = 0$ and

$$\limsup_{(x,t) \to (0,0)} |\xi(x,t)| \leq 0,$$

then

$$\xi(x,t) \leq C_\xi \left( |x|^2 + |t|^2 \right) \quad \forall x \in \mathbb{R}^n, t \in [0,T].$$

**Theorem (Error estimate: second part)**

Under the above assumptions, the following estimate holds

$$\sup_{(x,t) \in \mathbb{R}^d \times [0,T]} \left( V(t,x) - v(t,x) \right) \leq C(T) \Delta t.$$
Error estimates for the approximate value $V$

**Proposition**

Let $f, L$ and $g$ be Lipschitz continuous, bounded. Moreover let $L$ and $g$ be semiconcave and $f \in C^1$. Then the approximate solution $V$ is semiconcave.

**Lemma (Capuzzo-Ishii, ’84)**

Let $\xi$ be semiconcave such that $\xi(0, 0) = 0$ and

$$\limsup_{(x,t) \to (0,0)} \frac{\xi(x,t)}{|x|+|t|} \leq 0,$$

then

$$\xi(x, t) \leq \frac{C_\xi}{6} (|x|^2 + |t|^2) \quad \forall x \in \mathbb{R}^n, t \in [0, T].$$
Error estimates for the approximate value $V$

**Proposition**

Let $f$, $L$ and $g$ be Lipschitz continuous, bounded. Moreover let $L$ and $g$ be semiconcave and $f \in C^1$. Then the approximate solution $V$ is semiconcave.

**Lemma (Capuzzo-Ishii, ’84)**

Let $\xi$ be semiconcave such that $\xi(0, 0) = 0$ and
\[
\limsup_{(x, t) \to (0,0)} \frac{\xi(x, t)}{|x| + |t|} \leq 0,
\]
then
\[
\xi(x, t) \leq \frac{C_\xi}{6} (|x|^2 + |t|^2) \quad \forall x \in \mathbb{R}^n, \ t \in [0, T].
\]

**Theorem (Error estimate: second part)**

Under the above assumptions, the following estimate holds
\[
\sup_{(x,t) \in \mathbb{R}^d \times [0, T]} (V(t, x) - v(t, x)) \leq C(T)\Delta t.
\]
Error estimates with pruning

Let us define the *pruned trajectory*:

\[
\eta_{j}^{n+1} = \eta^{n} + \Delta t f(\eta^{n}, u_j, t_n) + \mathcal{E}_{\varepsilon_T}(\eta^{n} + \Delta t f(\eta^{n}, u_j, t_n), \{\eta_{i}^{n+1}\}_{i}),
\]

where

\[
\mathcal{E}_{\varepsilon_T}(x, \{x_n\}_n) = \begin{cases} 
    x_k - x & \text{if } \min_n |x - x_n| = |x - x_k| \leq \varepsilon_T, \\
    0 & \text{otherwise.}
\end{cases}
\]
Error estimates with pruning

Let us define the pruned trajectory:

\[ \eta_{j}^{n+1} = \eta^{n} + \Delta t f(\eta^{n}, u_j, t_n) + \mathcal{E}_{\varepsilon_T}(\eta^{n} + \Delta t f(\eta^{n}, u_j, t_n), \{\eta_{i}^{n+1}\}_{i}) , \]

where

\[ \mathcal{E}_{\varepsilon_T}(x, \{x_n\}_n) = \begin{cases} x_k - x & \text{if } \min_n |x - x_n| = |x - x_k| \leq \varepsilon_T, \\ 0 & \text{otherwise}. \end{cases} \]

Proposition

Given the Euler approximation \( \{y^n\}_n \) and its perturbation \( \{\eta^n\}_n \), then

\[ |y^n - \eta^n| \leq n \varepsilon_T e^{L_f(t_n-t)}. \]

To guarantee first order convergence, the tolerance must be chosen such that

\[ \varepsilon_T \leq C_{\varepsilon_T} \Delta t^2. \]
Error estimates with pruning

Then we can define the *pruned* discrete cost functional and value function

\[
J_{x,t_n}^{\Delta t,P}(u) = \Delta t \sum_{k=n}^{N-1} L(\eta^k, u, t_k) e^{-\lambda(t_k-s)} + g(\eta^N) e^{-\lambda(t_N-s)},
\]

\[
V^P(x, t) := \inf_{u \in \mathcal{U}^\Delta} J_{x,t}^{\Delta t,P}(u)
\]
Error estimates with pruning

Then we can define the \textit{pruned} discrete cost functional and value function

\[
J_{\Delta t, P}^{\Delta t, P}(u) = \Delta t \sum_{k=n}^{N-1} L(\eta^k, u, t_k) e^{-\lambda(t_k-s)} + g(\eta^N) e^{-\lambda(t_N-s)},
\]

\[
V^P(x, t) := \inf_{u \in U^\Delta} J_{\Delta t, P}^{\Delta t, P}(u)
\]

**Proposition**

Choosing \( \varepsilon_T \leq C_{\varepsilon_T} \Delta t^2 \), we have

\[
|V(x, t) - V^P(x, t)| \leq C(T)\Delta t,
\]

and then

\[
|v(x, t) - V^P(x, t)| \leq C(T)\Delta t.
\]
We consider the following dynamics

\[ f(x, u) = \begin{pmatrix} u \\ x_2 \end{pmatrix}, \quad u \in U \equiv [-1, 1]. \]

where \( x = (x_1, x_2) \in \mathbb{R}^2 \), and the following cost functional:

\[ J_{x,t}(u) = -x_2(T; u). \]

We compare the approximations according to \( \ell_2 \) relative error

\[ E_2(t_n) = \sqrt{\frac{\sum_{x_i \in T^n} |v(x_i, t_n) - V^*(x_i)|^2}{\sum_{x_i \in T^n} |v(x_i, t_n)|^2}}. \]
Test 1: Comparison with exact solution

Figure: Full Tree ($|\mathcal{T}| = 2097151$) (left) and Pruned Tree with $\mathcal{E}_\mathcal{T} = \Delta t^2$ ($|\mathcal{T}| = 3151$) (right)

Figure: Error $\ell_2$ with different initial conditions
Test 1: Comparison with exact solution

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Nodes</th>
<th>CPU</th>
<th>$Err_{2,2}$</th>
<th>$Err_{\infty,2}$</th>
<th>$Order_{2,2}$</th>
<th>$Order_{\infty,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>63</td>
<td>0.05s</td>
<td>6.7e-02</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>2047</td>
<td>0.35s</td>
<td>2.9e-02</td>
<td>0.09</td>
<td>1.16</td>
<td>0.98</td>
</tr>
<tr>
<td>0.05</td>
<td>2097151</td>
<td>1.1s</td>
<td>1.4e-02</td>
<td>0.05</td>
<td>1.08</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Table: Table for Euler scheme for the Full Tree

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Nodes</th>
<th>CPU</th>
<th>$Err_{2,2}$</th>
<th>$Err_{\infty,2}$</th>
<th>$Order_{2,2}$</th>
<th>$Order_{\infty,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>42</td>
<td>0.05s</td>
<td>9.1e-02</td>
<td>0.122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>324</td>
<td>0.08s</td>
<td>4.4e-02</td>
<td>0.062</td>
<td>1.05</td>
<td>0.98</td>
</tr>
<tr>
<td>0.05</td>
<td>3151</td>
<td>0.6s</td>
<td>2.1e-02</td>
<td>0.031</td>
<td>1.04</td>
<td>0.99</td>
</tr>
<tr>
<td>0.025</td>
<td>29248</td>
<td>2.5s</td>
<td>1.1e-02</td>
<td>0.016</td>
<td>1.005</td>
<td>0.994</td>
</tr>
<tr>
<td>0.0125</td>
<td>252620</td>
<td>150s</td>
<td>5.3e-03</td>
<td>0.008</td>
<td>1.004</td>
<td>0.997</td>
</tr>
</tbody>
</table>

Table: Table for Euler scheme with $\varepsilon_T = \Delta t^2$
Test 1: Comparison with exact solution

Figure: Comparison of the order of convergence for the pruned TSA with different tolerances (left) with Euler method and (right) with Heun’s method.
Test 2: Heat Equation

We deal with the control of the **heat equation** with Dirichlet boundary conditions.

This test is unfeasible via a direct semi-Lagrangian approach.

Dynamics

\[
\begin{cases}
    y_t = \sigma y_{xx} + y_0(x)u(t) & (x, t) \in (0, 1) \times (0, T), \\
    y(0, t) = y(1, t) = 0 & t \in (0, T), \\
    y(x, 0) = y_0(x) & x \in [0, 1],
\end{cases}
\]
Semi-discretization in space

We set $T = 1$, $\sigma = 0.15$ and $y_0(x) = x - x^2$. and we apply a centered finite difference method in space getting dynamics

$$\begin{cases}
    \dot{y}(t) &= Ay(t) + Bu(t), \\
y(0) &= y_0
\end{cases}$$

where $A \in \mathbb{R}^{d \times d}$ is the stiffness matrix and $B \in \mathbb{R}^d$ is given by $B_i = y_0(x_i)$ for $i = 1, \ldots, d$, $x_i$ are the nodes.

We want to minimize the cost functional

$$J_{y_0,t}(u) = \int_t^T \left( \|y(s)\|_2^2 + \frac{1}{100}|u(s)|^2 \right) \, ds + \|y(T)\|_2^2.$$
Comparison with the exact solution (Riccati)

When the control is unconstrained, we can derive an exact solution solving the Riccati differential equation. We compute the errors in $L^2$ and in $L^\infty$

$$Err_2 := \frac{\sum_{n=0}^{N} |V(y^*_n, t_n) - v(y^n_R, t_n)|^2}{\sum_{n=0}^{N} |v(y^n_R, t_n)|^2}$$

$$Err_\infty := \frac{\max_{n=0,\ldots,N} |V(y^*_n, t_n) - v(y^n_R, t_n)|}{\max_{n=0,\ldots,N} |v(y^n_R, t_n)|}$$

where $\{y^*_n\}_n$ is the optimal trajectory computed via TSA and $\{y^n_R\}_n$ is obtained solving the Riccati equation.
TSA approximation

For $\Delta x = 10^{-2}$, we get a system of dimension $d = 100$.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Nodes</th>
<th>P/F ratio</th>
<th>CPU</th>
<th>$Err_2$</th>
<th>$Err_\infty$</th>
<th>$Order_2$</th>
<th>$Order_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>134</td>
<td>4.7e-09</td>
<td>0.14s</td>
<td>0.279</td>
<td>0.241</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>863</td>
<td>1.2e-18</td>
<td>0.65s</td>
<td>0.144</td>
<td>0.118</td>
<td>0.95</td>
<td>1.03</td>
</tr>
<tr>
<td>0.025</td>
<td>15453</td>
<td>3.1e-38</td>
<td>12.88s</td>
<td>5.5e-2</td>
<td>5.3e-2</td>
<td>1.40</td>
<td>1.17</td>
</tr>
<tr>
<td>0.0125</td>
<td>849717</td>
<td>3.8e-78</td>
<td>1.1e+3s</td>
<td>1.6e-2</td>
<td>1.6e-2</td>
<td>1.77</td>
<td>1.42</td>
</tr>
</tbody>
</table>

Table: Test 2: Error analysis and order of convergence for forward Euler scheme of the TSA with $\varepsilon_T = \Delta t^2$ and 11 discrete controls.
TSA with and without pruning

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>P/P ratio</th>
<th>F/F ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>6.44</td>
<td>2.6e10</td>
</tr>
<tr>
<td>0.025</td>
<td>17.9</td>
<td>6.7e20</td>
</tr>
<tr>
<td>0.0125</td>
<td>984</td>
<td>4.5e41</td>
</tr>
</tbody>
</table>

Table: Test 2: Comparison between the ratio of cardinality for the full and the pruned tree for $\varepsilon_T = \Delta t^2$ and 11 discrete controls.
We set $\Delta t = 10^{-4}$ for the Riccati equation to get an accurate solution. For a fair comparison, we first computed the LQR problem and then set the control space in the TSA, $U = [-1, 0]$. 

**Figure:** Test 2: Cost functional (left) and optimal control (right) with 11 discrete controls.
**Figure:** Test 2: Cost functional (left) and optimal control (right) with 100 discrete controls.
Conclusions

- We presented a new algorithm to solve finite horizon optimal control problems using a tree structure with first order convergence.
- The pruning rule will mitigate the "curse of dimension"
- It can be easily extended to high-order methods (Saluzzi’s talk).
- It can be applied to general non linear control problems over a finite horizon.
- We can couple this method with POD to obtain a more efficient algorithm (Saluzzi’s talk)
Conclusions and future works

**Conclusions**

- We presented a new algorithm to solve finite horizon optimal control problems using a **tree structure** with first order convergence.
- The pruning rule will mitigate the "curse of dimension"
- It can be easily extended to high-order methods (Saluzzi’s talk).
- It can be applied to general non linear control problems over a finite horizon.
- We can couple this method with POD to obtain a more efficient algorithm (Saluzzi’s talk)

**Future works**

- Extension to stochastic control problems
- Efficient Feedback reconstruction.
- Algorithm improvements.
Thank you for the attention


2. A. Alla, M. Falcone, L. Saluzzi, *High-order Approximation of the Finite Horizon Control Problem via a Tree Structure Algorithm*, IFAC CPDE, 2019


