# Lecture 15: Collocation & Differential Dynamic Programming

**Last Time:** Numerical Methods for solving $\dot{x}(t) = f(t, x(t))$ $x(0) = x_0$

**Today:**

- Detail use of these methods to solve

$$\min_{u(\cdot)} \int_0^T \ell(t, x(t), u(t))\, dt \quad \Rightarrow \text{ DIRECT COLLOCATION}$$

$$\text{s.t. } \dot{x}(t) = f(t, x(t), u(t))$$

- Introduction to Differential Dynamic Programming
  - Combines Direct Shooting, Dynamic Programming, and indirect ideas
  - Computational efficiency & Rapid Convergence

# Optimal Control Family Tree

$$\min_{u(\cdot)} \int_0^{t_f} \ell(t, x(t), u(t))\,dt + \ell_f(x(t_f))$$

## Optimal Control

### Dynamic Programming
Value iteration
HJB / Bellman Eq
Tabulation in State Space

- Small problems

### Indirect methods
PMP
Solve a boundary value problem

- Difficult to make a good guess for $\lambda(t)$
- "Optimize, then discretize"

### Direct Methods
Transform OCP into a nonlinear programing problem

"discretize then optimize"

#### Direct Collocation
- Discretize $u(t)$
- Discretize $x(t)$
- Add constraints that approximate sim.

Simultaneous

#### Multi Shooting
- Discretize $u(t)$
- Simulate multiple Segments
- Add constraints to make them line up

Simultaneous

#### Single shooting
- Discretize $u(t)$
- Simulate to find $x(t)$

Sequential

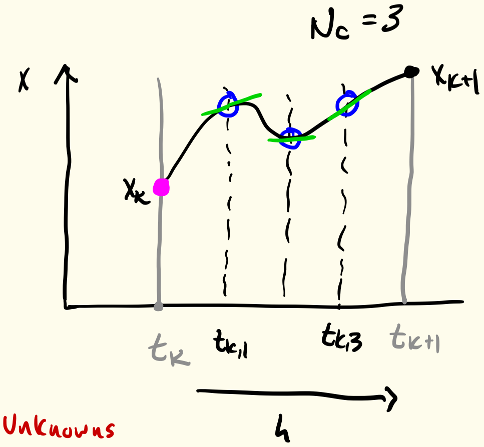
Collocation: Approximating the solution as a polynomial to derive integration schemes

$$X(t) = \sum_{j=0}^{N_c} a_j t^j$$

To BE DETERMINED

$N_c+1$ unknowns

$$\dot{x}(t) = \sum_{j=1}^{N_c} j\, a_j t^{j-1}$$




Consider $N_c$ collocation points

$$\{t_{k,1}, \ldots, t_{k,N_c}\}$$

$\{x_{k,1}, \ldots, x_{k,N_c}\}$ State @ collocation pts $N_c$ Unknowns

Collocation constraints

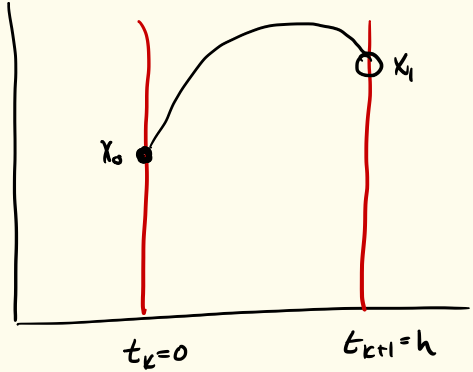
1. $x_k = x(0)$    $N_c$: $X(t_{k,i}) = x_{k,i}$    $N_c$: $\dot{x}(t_{k,i}) = f(t_{k,i}, x_{k,i})$

$2N_c+1$ equations for $2N_c+1$ unknowns

$$x_{k+1} = \sum_{j=0}^{N_c} a_j t_{k+1}^j \Rightarrow O(h^{N_c+1}) \text{ truncation error}$$

Example: $\dot{x} = f(t,x)$ $\quad x(0) = x_0$

Approximate $x(h)$ using collocation w/ collocation points @ $\{0, h\}$




**Constraints:** $x(t) = a_0 + a_1 t + a_2 t^2$ $\quad \dot{x}(t) = a_1 + 2a_2 t$

$x_0 = a_0$

$x_h = x(h) = a_0 + a_1 h + a_2 h^2$

$\dot{x}_0 =: f(0, x_0) = \dot{x}(0) = a_1$

$\dot{x}_h =: f(h, x_h) = a_1 + 2a_2 h$

$$\Rightarrow \frac{\dot{x}_h - \dot{x}_0}{2h} = a_2$$

$$\boxed{x_h = x_0 + \frac{h}{2}(\dot{x}_0 + \dot{x}_h)}$$

$$x_h = x_0 + \dot{x}_0 h + \frac{\dot{x}_h - \dot{x}_0}{2\cancel{h}} h^{\cancel{2}} = x_0 + \frac{h}{2}(\dot{x}_0 + \dot{x}_h)$$

**Trapezoid Rule:** Collocation points $\{0, 1\}$ LTE $O(h^3)$

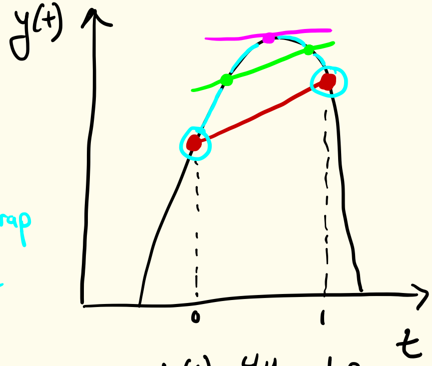
$$\int_0^1 y(t)\,dt = \frac{1}{2}y(0) + \frac{1}{2}y(1)$$

**Midpoint Rule:** Collocation Point $\{1/2\}$ LTE $O(h^3)$

$$\int_0^1 y(t)\,dt = y\left(\tfrac{1}{2}\right)$$

This collocation point is in an optimal location ⇒ Same LTE as trap rule despite 1 less colloc pt!

$$\int_0^1 y(t)\,dt = ?$$




$y(t)$ - 4th order polynomial

**Simpsons Rule:** Collocation points $\{0, \frac{1}{2}, 1\}$ LTE $O(h^5)$

$$\int_0^1 y(t)\,dt = \frac{1}{6}\left(y(0) + 4y\left(\tfrac{1}{2}\right) + y(1)\right)$$

+ One more pt. than trap
+ Point in optimal Spot
⇒ Two orders better LTE than Trap.

**Gauss Quadrature:** 2 collocation points $\{.211, .788\}$

$$\int_0^1 y(t)\,dt = \frac{1}{2}y(.211) + \frac{1}{2}y(.788)$$

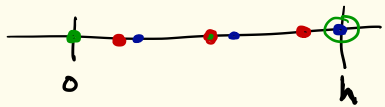
LTE $O(h^5)$

- Same # pts as Trap
+ Both in the optimal location
⇒ Two orders better LTE than trap

# Collocation Schemes: Picking collocation Points: $t_{k,j} = t_k + \tau_j h$

- **Gauss-Legendre** – all collocation points internal

  $O(h^{2N_c+1})$ LTE




  collocation points from roots of Legendre polynomial $P_n(x)$

- **Gauss-Radau** – one point on end $O(h^{2N_c})$

  collocation points from roots of a polynomial $\dfrac{P_{n-1}(x) + P_n(x)}{1+x}$

- **Gauss-Lobatto** – both ends $O(h^{2N_c-1})$ LTE

  roots of $P'_{n-1}(x)$

- Chebychev - Gauss (CG) / CGR / CGL $O(h^{N_c+1})$ LTE

  collocation points by simple trig

# Eliminating Polynomial Coefficients: General Case

$$x(t) = \sum_{j=0}^{N_c} a_j t^j$$

$$\dot{x}(t) = \sum_{j=1}^{N_c} j\, a_j t^{j-1}$$

- Suppose the k-th Fin. element $[t_k, t_{k+1}] = [0, 1]$
- Consider collocation points @ $0 \le \tau_1 < \dots < \tau_{N_c} \le 1$

$$\begin{bmatrix} x_k \\ \dot{x}_{k,1} \\ \vdots \\ \dot{x}_{k,N_c} \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 + 2a_2\tau_1 + 3a_3\tau_1^2 + \dots \\ \vdots \\ a_1 + 2a_2\tau_{N_c} + 3a_3\tau_{N_c}^2 + \dots \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 2\tau_1 & 3\tau_1^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 1 & 2\tau_{N_c} & 3\tau_{N_c}^2 & \dots \end{bmatrix}}_{\text{Invertible!}} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N_c} \end{bmatrix}$$

$\Rightarrow a_0, \dots, a_{N_c}$ are linear in $x_k, \dot{x}_{k,1}, \dots, \dot{x}_{k,N_c}$

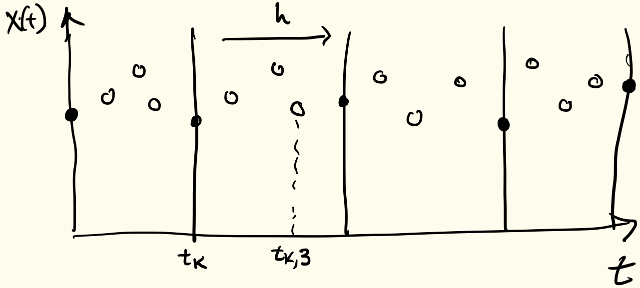
$\Rightarrow x_{k,j} = x(\tau_j) = x_k + \sum_{i=1}^{N_c} B_{ji}\, \dot{x}_{k,i}$ for some $B_{ji}$ fixed

$\Rightarrow x_{k+1} = x_k + \sum_{i=1}^{N_c} w_i\, \dot{x}_{k,i}$ for some $w_i$ fixed

## Formulating Traj. Opt via collocation:

$$\min_{u(\cdot)} \int \ell(t, x(t), u(t))\,dt$$

$$\text{s.t.} \quad \dot{x}(t) = f(t, x, u)$$




**Suppose:** $N_e$ Finite elements, $N_c$ Collocation points/Elem.

$$@ \; t_{K,j} = t_K + \tau_j h$$

4 finite elem

3 colloc pts / elem

**Transcription:**

$$\min_{\{x_K, x_{K,j}, \dot{x}_{K,j}, u_{K,j}\}} \sum_{K=1}^{N_e} h \sum_{j=1}^{N_c} w_j \, \ell(t_{K,j}, x_{K,j}, u_{K,j})$$

$$\text{S.t.} \quad \left.\begin{aligned} \dot{x}_{K,j} &= f(t_{K,j}, x_{K,j}, u_{K,j}) \\ x_{K,j} &= x_K + h \sum_{i=1}^{N_c} B_{ji} \dot{x}_{K,i} \\ x_{K+1} &= x_K + h \sum_{i=1}^{N_c} w_i \dot{x}_{K,i} \end{aligned}\right]$$

Collocation constraints

| | Single Shooting | Mutiple Shooting | Direct Collocation |
|---|---|---|---|
| VARIABLES | Finite dimensional parameterization of control<br><br>Optional: Time Horizon | Finite dimensional parameterization of control<br><br>Initial conditions for each shooting element<br><br>Optional: Time Horizon | State @ start of element<br>State @ each collocation pt<br>Control @ each collocation pt<br><br>Optional: Time Horizon |
| CONSTRAINTS | Control parameter constraints<br><br>Final state constraints | Control parameter constraints<br><br>State constraints at the start of each element<br><br>Continuity constraints form element to element | State/control constraints @ each node or collocation pt<br><br>Collocation constraints |
| PRO / CON | • Easy<br>• Sensitive<br>• Slow<br>• Non-sparse coupling | • Less sensitive<br>⇒ Less pronounced nonlinearities<br>• Precise<br>• More complexity<br>• Formulations have more dense Lin. Alg. vs. Colloc | • Sparse Lin Alg<br>• Less sensitive yet<br>• Scalability<br>• More knobs<br>• Mesh conv. analysis needed<br>• "Easier" to initialize<br>• State constraints easy to enforce |

# Optimal Control Family Tree

$$\min_{u(\cdot)} \int_0^{t_f} \ell(t, x(t), u(t))\,dt + \ell_f(x(t_f))$$

## Optimal Control

### Dynamic Programming
Value iteration
HJB / Bellman Eq
Tabulation in State Space

- Small problems

### Indirect methods
PMP
Solve a boundary value problem

- Difficult to make a good guess for $x(t)$
- "Optimize, then discretize"

### Direct Methods
Transform OCP into a nonlinear programing problem

"discretize then optimize"

#### Direct Collocation
- Discretize $u(t)$
- Discretize $x(t)$
- Add constraints that approximate sim.

Simultaneous

#### Multi Shooting
- Discretize $u(t)$
- Simulate multiple segments
- Add constraints to make them line up

Simultaneous

#### Single shooting
- Discretize $u(t)$
- Simulate to find $x(t)$

Sequential

# Differential Dynamic Programming

Dynamic programming doesn't scale...

Direct optimization is only open loop...

Rather than doing value iteration everywhere...

why not just around a trajectory?

$\Rightarrow$ This is the main Idea of DDP

$\Rightarrow$ Gives an optimal open loop trajectory

and a locally optimal policy

$\Rightarrow$ Exploits structure of the traj. opt. problem

TO BE CONTINUED...