MINI-BEE FLIGHT OPTIMIZATION

- Obstacle avoidance
- Polynomial approximation

BELMANT Cédric – CLERC Ilinka

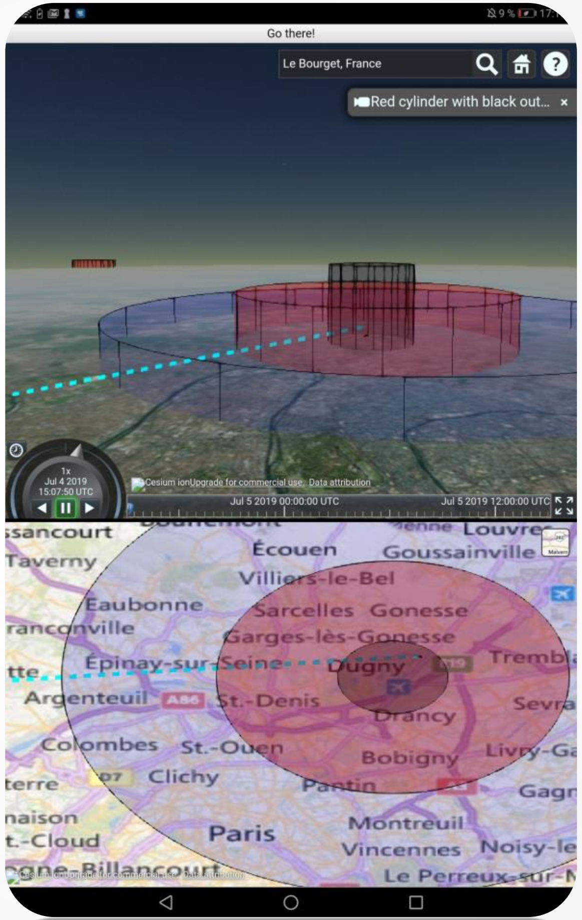
01/23/2020

INTRODUCTION

THE MINI-BEE

The Mini-Bee:

- A Hybrid Vertical Take-Off and Landing aircraft (**VTOL**) - can fly as a **plane**, a **helicopter** or **both**
- Aims to become a flying **ambulance**



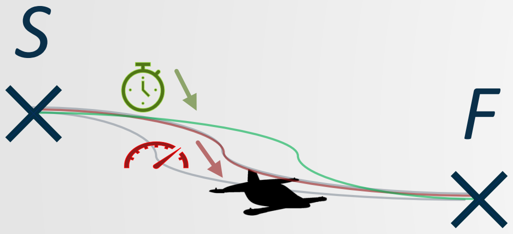
Mini-Bee project:

- A **collaborative** project

→ **A 3D collaborative GPS**

PROBLEM SPECIFICATION

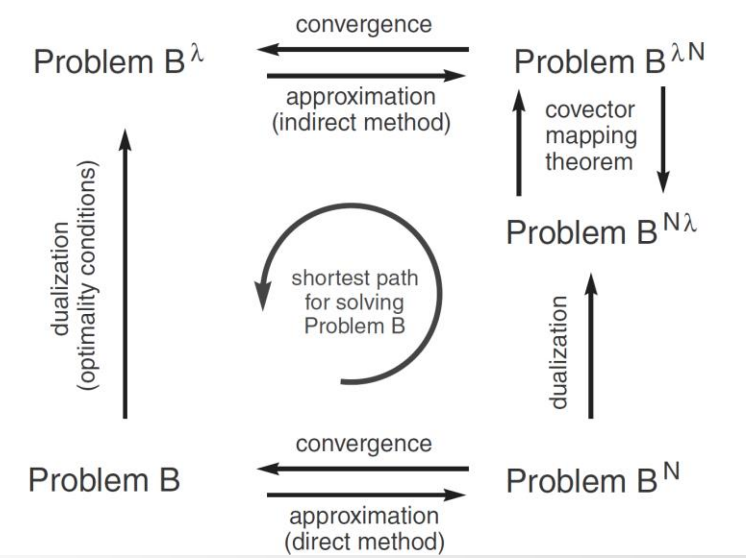
Help in flight management for the pilot



Optimization of some features

- Find an **eligible trajectory** from S to F in a **3D** environment
- Take **static obstacles** into account
- Avoid **moving objects**
- Consider **meteorological phenomena** (as constraints or additional effects on the physical model)
- Favor **air corridors** or **roads**
- Add **intermediary points**
- Implement a **collaborative** aspect
- **Minimize** a given criterion like **flight time** or **fuel consumption**

STATE OF THE ART



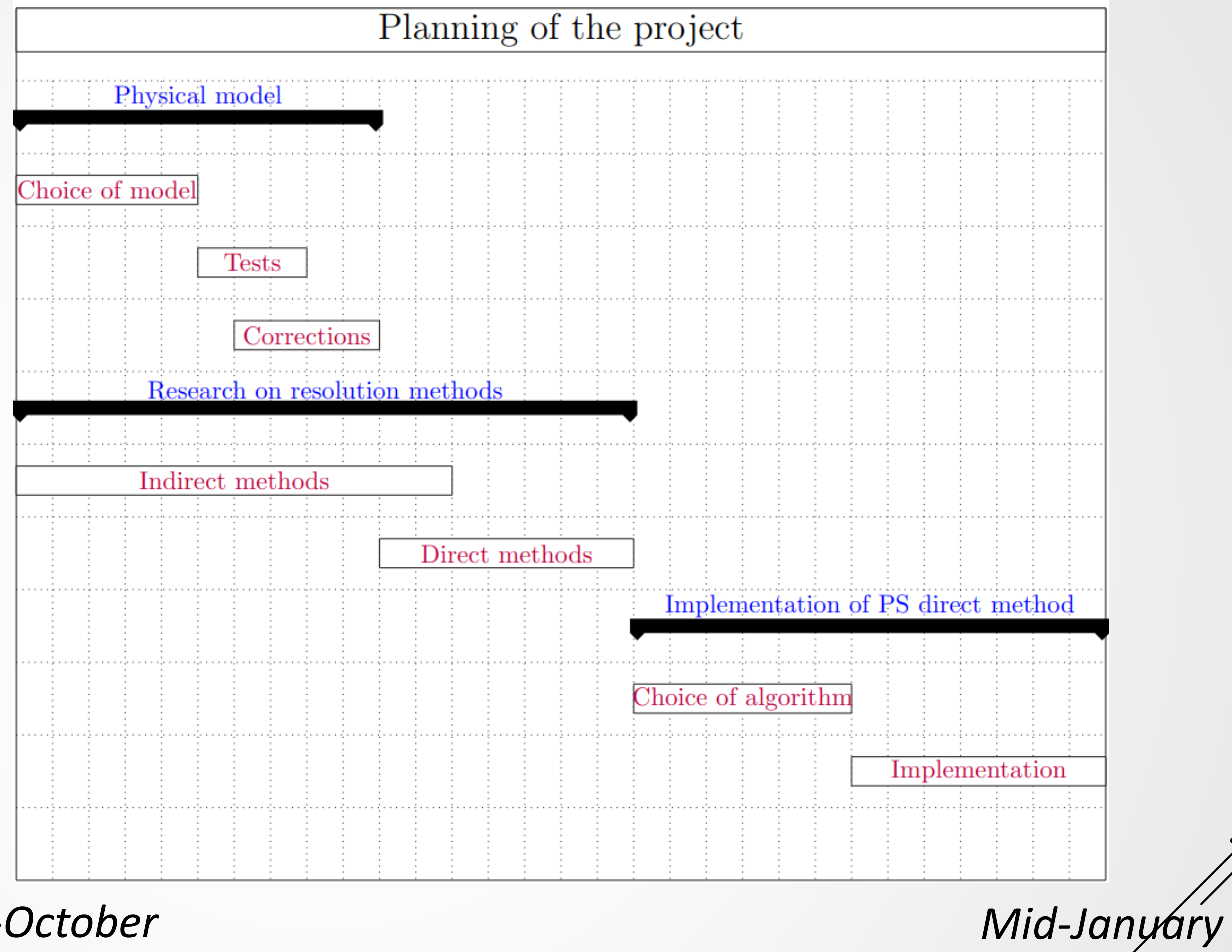
Indirect methods

- Generalized LQR theory
  - Not applicable
- Approximate Dynamic Programming
  - No convergence proofs
  - Very complex

Direct methods

- Finite elements
  - Applicable
- Pseudospectral methods
  - Convergence proofs
  - Already proven

SUMMARY


Planning of the project
Physical model
Choice of model
Tests
Corrections
Research on resolution methods
Indirect methods
Direct methods
Implementation of PS direct method
Choice of algorithm
Implementation
Mid-October
Mid-January

PHYSICAL MODELING

CONTROL

To simplify the problem, we first consider a **plane** configuration

In a aircraft we have commands on:

- The **thrust level** in the forward axis – Control on the fuel consumption
- The **angular velocity** in all axes – No incidence on the fuel consumption

$$\underbrace{U = \begin{bmatrix} U_1 \\ \left.\begin{matrix} U_2 \\ U_3 \\ U_4 \end{matrix}\right] \; U_\omega \end{bmatrix}}_{\text{Control}}$$

MODEL

$$\begin{cases} \dot{q} = \frac{1}{2} M_s(U_\omega) q \\ \dot{x} = v \\ \dot{v} = \frac{F_a}{m} + g + \frac{U_1}{m}(k_t v + i) \\ \dot{m} = -k_t U_1 \end{cases}$$

$$\dot{X}(t) = F(X(t), U(t)) \quad X(t) = \begin{pmatrix} q(t) \\ x(t) \\ v(t) \\ m(t) \end{pmatrix}$$

Hypothesis: the **angular velocity** is directly controlled

Position in the global coordinate system: $x(t) = (x_1(t)\ x_2(t)\ x_3(t))^T$

Speed in the global coordinate system: $v(t) = (v_1(t)\ v_2(t)\ v_3(t))^T$

F_a the **aerodynamic force**

g the **gravitational force**

m the **mass**

$k_t > 0$ **constant of proportionality** depending on the engine type and atmospheric values

Hypothesis: the mass derivative is proportional to the thrust

COST

$$J(X,U,t_f) = \gamma(t_f - t_0) + (1-\gamma)\int_{t_0}^{t_f} \underbrace{(k_t U_1(t))}_{\dot{m}}\, dt$$

Cost on the time flight

$\gamma = 1$
optimization on the flight time

$\gamma \in [0,1]$

Cost on the fuel consumption

$\gamma = 0$
optimization on the fuel consumption

OPTIMIZATION PROBLEM

Find the solution to $\min_{X,U,t_f} J(X,U,t_f)$ — **Minimisation** of the **cost**

Subject to $\dot{X}(t) = F(X(t),U(t))$ — Respect of the **physical equation**

$$\|U_i\|_\infty < U_i^{max}, \forall i \in [\![1,4]\!]$$

$$U_1(t) \geq 0, \forall t \in [t_0, t_f]$$

$$x(t_f) = x_f$$

$$X(t_0) = X_0$$

$$\left\|\frac{x_1 - x_c^{1,i}}{a^i}\right\|^{p_1^i} + \left\|\frac{x_2 - x_c^{2,i}}{b^i}\right\|^{p_2^i} + \left\|\frac{x_3 - x_c^{3,i}}{c^i}\right\|^{p_3^i} \geq 1, \forall i \in [\![1, N_c]\!]$$

Respect of five **constraints** on the **cost** and the **state**

RESOLUTION METHOD

RESOLUTION OF THE OPTIMIZATION PROBLEM

An infinite dimensional optimization problem in continuous time

$$X(t) = \begin{pmatrix} q(t) \\ x(t) \\ \mathrm{v}(t) \\ \mathrm{m}(t) \end{pmatrix} \begin{matrix} 4 \\ 3 \\ 3 \\ 1 \end{matrix} \; 11 \qquad U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} \; 4$$

Complex constraints

Polynomial basis

Finite Dimensional Problem Approximation

Legendre Pseudospectral resolution method

Solution

LEGENDRE PSEUDOSPECTRAL NONLINEAR PROBLEM

$$\dot{\hat{X}}_i(t_k^L) = f(\hat{X}_i(t_k^L), \hat{U}_i(t_k^L))$$

Lagrange polynomials : φ_n

$$\dot{\hat{X}}_i(t_k^L) = \sum_{n=0}^{N} \hat{X}_{i,n}\, \dot{\varphi}_n(t_k^L) = \sum_{n=0}^{N} \hat{X}_{i,n}\, D_{kn}$$

$$\hat{X}_i(t_k^L) = \sum_{n=0}^{N} \hat{X}_{i,n}\, \varphi_n(t_k^L)$$

$$\hat{U}_i(t_k^L) = \sum_{n=0}^{N} \hat{U}_{i,n}\, \varphi_n(t_k^L)$$

$$D_{kn} = \dot{\varphi}_n(t_k^L)$$

$$D_{kn} = \frac{2}{t_f - t_0} \begin{cases} \dfrac{L_N(\tau_k)}{L_N(\tau_n)} \dfrac{1}{\tau_k - \tau_n}, & \text{if } k \neq n \\ -\dfrac{N(N+1)}{4}, & \text{if } k = n = 0 \\ \dfrac{N(N+1)}{4}, & \text{if } k = n = N \\ 0, & \text{otherwise} \end{cases}$$

$$\sum_{n=0}^{N} \hat{X}_{i,n}\, D_{kn} = f(\hat{X}_i(t_k^L), \hat{U}_i(t_k^L))$$

REWRITED OPTIMIZATION PROBLEM

Find the solution to $\min_{\hat{X},\hat{U},t_f} J(\hat{X},\hat{U},t_f)$ — **Minimisation** of the **cost**

Subject to $\left\| \sum_{n=0}^{N} \hat{X}_{i,n} D_{kn} - f\left(\hat{X}(t_k^L), \hat{U}(t_k^L)\right)_i \right\|_\infty - \delta \leq 0, \forall i \in [\![1, 11]\!]$ — Respect of the **physical equation** : $\dot{X} = F(X, U)$

$$\left\|\hat{U}_i\right\|_\infty - \hat{U}_i^{max} < 0, \forall i \in [\![1, 4]\!]$$

$$-\hat{U}_1(t_n^L) \leq 0, \forall n \in [0, N]$$

$$\left\|\hat{x}_i(t_N^L) - \hat{x}_f^i\right\|_\infty - \delta \leq 0, \forall i \in [\![1, 11]\!]$$

$$\left\|\hat{X}_i(t_0^L) - \hat{X}_0^i\right\|_\infty - \delta \leq 0, \forall i \in [\![1, 11]\!]$$

$$1 - \left\|\frac{\hat{x}_1 - x_c^{1,i}}{a^i}\right\|^{p_1^i} - \left\|\frac{\hat{x}_2 - x_c^{2,i}}{b^i}\right\|^{p_2^i} - \left\|\frac{\hat{x}_3 - x_c^{3,i}}{c^i}\right\|^{p_3^i} \leq 0, \forall i \in [\![1, N_c]\!]$$

Respect of five **constraints** on the **cost** and the **state**

PROGRESS

INVESTIGATED ALGORITHMS

Global optimization (NOMAD)

- Blackbox optimization
- Very slow
- Poor parametrization

Discovery of the PS method

Legendre PS method

- Proprietary versions (MATLAB, optimal control toolboxes…)
- Only one open-source C++ project

→ Complex code, hard to customize and debug

→ Implementation of a PS algorithm in Python (in progress)

SOFTWARE DEVELOPMENT

Requirements

- **Fast** optimization (equivalent to C++)
- **Easy** to use, develop and interface

→ **ADOL-C** (automatic differentiation)
IPOPT (nonlinear optimizer)
Just-in-Time compilation (*numba*)

All the building blocks are implemented…

…but convergence is not achieved

REMAINING WORK

Solve convergence **problems**

Integrate all the specificities of **the initial goal**

- Elaborate the **physical model**
- Consider **real time aspects**
- Add **other functionalities**

CONCLUSION

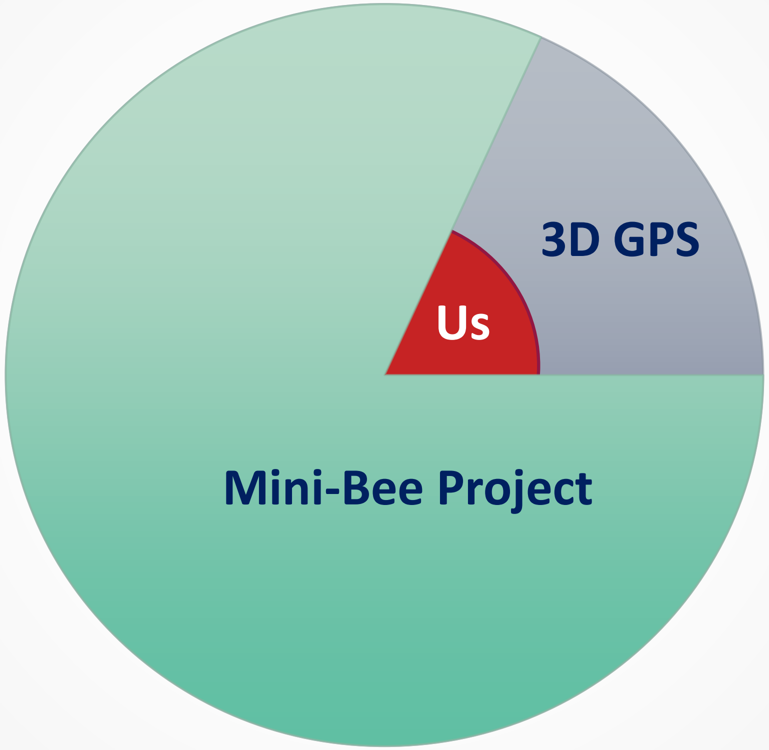
PROJECT PROGRESS

Construction of a 3D GPS for the Mini-Bee project

Simple case:
- Plane configuration
- Few initial specifications
- ➢ Still a **complex** optimization problem to solve

1. Writing of a physical model
2. Finding of an appropriate method
3. Re…lts

PROJECT CONTINUITY



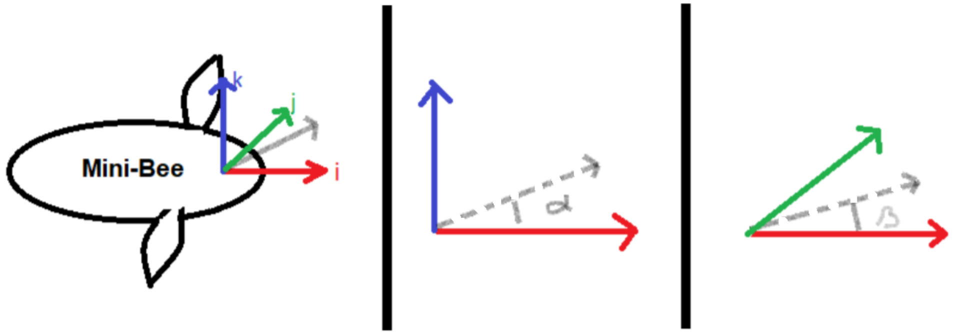
THANKS

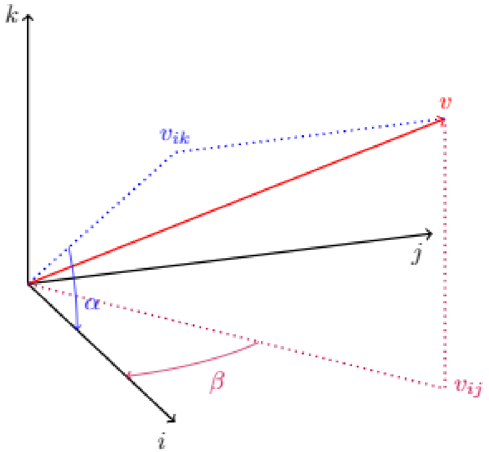
COORDINATE SYSTEM

Body coordinate system

Attack angle

Sideslip angle





$$i = \begin{pmatrix} 2(q_0^2 + q_1^2) - 1 \\ 2(q_1 q_2 + q_0 q_3) \\ 2(q_1 q_3 - q_2 q_0) \end{pmatrix}$$

Quaternions :

- Dimension 4 elements representing the **aircraft attitude**
- **Numerically more stable** than Euler angles but more **complex** to represent

$$q(t) = \begin{pmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \end{pmatrix}$$

AERODYNAMIC FORCES

Wind frame

$$F_a = F_D + F_L + F_C$$

$$F_D = \eta_a \|v_a\|^2 \boxed{C_D}\, x_a$$ **Drag coefficient**

$$F_L = \eta_a \|v_a\|^2 \boxed{C_L}\, y_a$$ **Lift coefficient**

$$F_C = \eta_a \|v_a\|^2 \boxed{C_C}\, z_a$$ **Cross-wind coefficient**

$$\begin{pmatrix} C_X \\ C_Y \\ C_Z \end{pmatrix} = P_{B \to W} \begin{pmatrix} -C_D \\ -C_C \\ C_L \end{pmatrix}$$

Body frame

$$F_a = F_X + F_Y + F_Z$$

$$F_X = \eta_a \|v\|^2 C_X i$$

$$F_Y = \eta_a \|v\|^2 C_Y j$$

$$F_Z = \eta_a \|v\|^2 C_Z k$$

$$P_{B \to W} = \begin{pmatrix} cos\alpha\, cos\beta & -cos\alpha\, sin\beta & sin\alpha \\ sin\beta & cos\beta & 0 \\ -sin\alpha\, cos\beta & sin\alpha\, sin\beta & cos\alpha \end{pmatrix}$$

CONSTRAINTS

$$\|U_i\|_\infty < U_i^{max}, \forall i \in [\![1,4]\!]$$

Boundary condition on the **control**

$$U_1(t) \geq 0, \forall t \in [t_0, t_f]$$

Positivity of the **thrust**

$$x(t_f) = x_f$$

Constraint on the **final position**

$$X(t_0) = X_0$$

Constraint on the **initial state**

$$\left\|\frac{x_1 - x_c^{1,i}}{a^i}\right\|^{p_1^i} + \left\|\frac{x_2 - x_c^{2,i}}{b^i}\right\|^{p_2^i} + \left\|\frac{x_3 - x_c^{3,i}}{c^i}\right\|^{p_3^i} \geq 1, \forall i \in [\![1, N_c]\!]$$

Obstacle avoidance constraints

JACOBI AND LEGENDRE POLYNOMIALS

Jaco Polynomials

$L_\omega^2([-1,1]) = \{f:[-1,1] \rightarrow \mathbb{R}, \int_{-1}^{1} f(\tau)^2 \omega(\tau)\,d\tau < \infty\}$

$(P_i) \in \mathbb{P} \subset L_\omega^2([-1,1])$ a family of **orthogonal** polynomials of **degree i**

$\omega(\tau) = 1$

Legendre Polynomials

$(L_i) \in \mathbb{P} \subset L_1^2([-1,1])$

$L_0(\tau) = 1, L_1(\tau) = \tau, (n+1)L_{n+1}(\tau) = (2n+1)\tau L_n(t) - nL_{n-1}(\tau) \quad n \geq 2$

DISCRETE COEFFICIENTS

$$f(t) = \sum_{n=0}^{+\infty} f_n P_n(t)$$

$$\tilde{f}(t) = \sum_{n=0}^{N} \tilde{f}_n P_n(t) \quad t \in [t_0, tf]$$

Finite Dimensional Problem Approximation

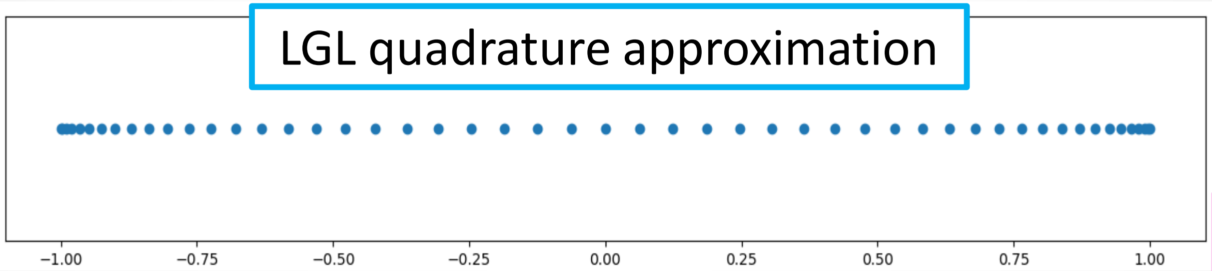
$$\tau = 2\frac{t - t_0}{t_f - t_0} - 1$$

$$\tilde{f}(\tau) = \sum_{n=0}^{N} \tilde{f}_n P_n(\tau) \quad \tau \in [-1,1]$$

Cannot be computed exactly

$$\tilde{f}_n = \frac{1}{\|P_n\|^2} \int_{-1}^{1} f(\tau) P_n(\tau)\, d\tau$$

LGL quadrature approximation



$$\tau_0^L = -1 \qquad \dot{L}(\tau_i^L) = 0 \qquad 1 = \tau_N^L$$

$$\hat{f}_n = \frac{1}{\|P_n\|^2} \sum_{i=0}^{N} f(\tau_i^L) P_n(\tau_i^L) \omega_i$$

$$\hat{f}(\tau) = \sum_{n=0}^{N} \hat{f}_n P_n(\tau) \quad \tau \in [-1,1]$$

$$t = (\tau + 1)\frac{t_f - t_0}{2} + t_0$$

$$\hat{f}(t) = \sum_{n=0}^{N} \hat{f}_n P_n(t) \quad t \in [t_0, tf]$$

$$\hat{f}_n = \frac{1}{\|P_n\|^2} \sum_{i=0}^{N} f(t_i^L) P_n(t_i^L) \omega_i$$