REU: Numerical optimal control

Hoang Nguyen
Note 9

June 28, 2018

1 Task 1: System of ODE

Problem: Given the dynamics

$$\begin{cases} X' = u_1 \\ Y' = u_2 - X \end{cases}$$

Minimize $\int_0^1 (u_1^2 + u_2^2) dt$

Solution: Using of calculus of variations, we have:

$$\delta \int_0^1 (u_1^2 + u_2^2) dt$$

$$= \delta \int_0^1 \left[(X')^2 + (Y' + X)^2 \right]$$

$$= \int_0^1 2X'^T \delta X' + 2(Y' + X)^T \delta (Y' + X) dt$$

$$= \int_0^1 2X'^T \delta X' + 2(Y' + X)^T \delta Y' + 2(Y' + X)^T \delta X dt$$

$$= \int_0^1 -2X''^T \delta X - 2(Y'' + X')^T \delta Y + 2(Y' + X)^T \delta X dt$$

$$= 2 \int_0^1 (-X'' + Y' + X)^T \delta X - 2 \int_0^1 (Y'' + X')^T \delta Y dt = 0$$

This gives us:

$$\begin{cases} -X'' + Y' + X = 0 \\ -Y'' - X' = 0 \end{cases}$$

Hence, we can model the dynamic as following (implicit method):

$$\begin{cases} d_\tau X = d_t^2 X - d_t Y - X \\ d_\tau Y = d_t^2 Y + d_t X \end{cases}$$

$$\Leftrightarrow \begin{cases} \frac{X_{i,j+1} - X_{i,j}}{\Delta \tau} = \frac{X_{i+1,j+1} - 2X_{i,j+1} + X_{i-1,j+1}}{(\Delta t)^2} - \frac{Y_{i+1,j+1} - Y_{i,j+1}}{(\Delta t)} - X_{i,j+1} \\ \frac{Y_{i,j+1} - Y_{i,j}}{\Delta \tau} = \frac{Y_{i+1,j+1} - 2Y_{i,j+1} + Y_{i-1,j+1}}{(\Delta t)^2} + \frac{X_{i+1,j+1} - X_{i,j+1}}{(\Delta t)} \end{cases}$$

$$\Leftrightarrow \begin{cases} \frac{X_{j+1} - X_j}{\Delta \tau} = L_1 X_{j+1} - L_2 Y_{j+1} - X_{j+1} \\ \frac{Y_{j+1} - Y_j}{\Delta \tau} = L_1 Y_{j+1} + L_2 X_{j+1} \end{cases}$$

$$\Leftrightarrow \begin{cases} \left(\frac{I_{n-2}}{h} + I_{n-2} - L_1\right) X_{j+1} = \frac{X_j}{h} - L_2 Y_j \\ \left(\frac{I_{n-2}}{h} - L_1\right) Y_{j+1} = \frac{Y_j}{h} + L_2 X_j \end{cases}$$

$$\Leftrightarrow \begin{cases} X_{j+1} = \left(\frac{I_{n-2}}{h} + I_{n-2} - L_1\right)^{-1} \left(\frac{X_j}{h} - L_2 Y_j\right) \\ Y_{j+1} = \left(\frac{I_{n-2}}{h} - L_1\right)^{-1} \left(\frac{Y_j}{h} + L_2 X_j\right) \end{cases}$$

2 Task 2: Newtonian dynamic with arbitrary cost

Problem: Solve the Euler-Lagrange equation of $\int_0^T (q'' - f(q))^2 dt$ where f is arbitrary.

Solution:

$$\begin{aligned} &\delta \int_0^T (q'' - f(q))^2 dt \\ &= 2\int_0^T (q'' - f(q))^T \delta(q'' - f(q)) dt \\ &= 2\int_0^T (q'' - f(q))^T \delta q'' - (q'' - f(q))^T \delta f(q)) dt \\ &= 2\int_0^T -(q''' - f'(q)q')^T \delta q' - (q'' - f(q))^T f'(q) \delta q dt \\ &= 2\int_0^T \left[(q^{(4)} - f'(q)q'^2 - f'(q)q'')^T - (q'' - f(q))^T f'(q)\right] \delta q = 0 \end{aligned}$$

From the fundamental lemma and taking transpose both sides, we have:

$$\begin{aligned} &q^{(4)} - f'(q)q'^2 - f'(q)q'' - f'(q)^T(q'' - f(q)) = 0 \\ \Leftrightarrow\ &q^{(4)} - \left[f'(q) + f'(q)^T\right] q'' - f''(q)q'^2 + f'(q)^T f(q) = 0 \end{aligned}$$

Note that we have $f''(q)$ is a 3-tensor, which does not have clear definition on how to perform matrix multiplications. Professor Tao recommended to expand everything by elements to avoid confusion.

3 Gradient descent with momentum

Algorithm: The gradient descent with momentum is a stochastic iterative method that is defined by the following sequences:

$$\begin{cases} \delta_\tau q = p \\ \delta_\tau p = -\nabla V(q) - \gamma p \end{cases}$$

$$\Leftrightarrow \begin{cases} \frac{q_{i,j+1} - q_{i,j}}{\Delta\tau} = p_{i,j} \\ \frac{p_{j+1} - p_j}{\Delta\tau} = -\nabla V(q) - \gamma p_j \end{cases}$$

Intuition

It originates from Langevin dynamics (stochastic version) or the Newtonian dynamics with friction. The momentum quantity helps us converge to the local optimum faster albeit there would be some fluctuations.

Application

Let us apply the problem to the heat equation. We have:

$$\begin{cases} \delta_\tau q = p \\ \delta_\tau p = -\delta(q')^2 - \gamma p \end{cases}$$

$$\Leftrightarrow \begin{cases} \delta_\tau q = p \\ \delta_\tau p = 2q'' - \gamma p \end{cases}$$

According to calculus of variations. Discretize the problem, we have:

$$\begin{cases} \frac{q_{i,j+1}-q_{i,j}}{\Delta\tau} = p_{i,j+1} \\ \frac{p_{i,j+1}-p_{i,j}}{\Delta\tau} = \frac{2(q_{i+1,j+1}-2q_{i,j+1}+q_{i-1,j+1})}{(\Delta t)^2} - \gamma p_{i,j+1} \end{cases}$$

$$\Leftrightarrow \begin{cases} q_{j+1} - \Delta\tau I_n p_{j+1} = q_j + b \\ (1+\gamma\Delta\tau)\, I_n p_{j+1} - \frac{2\Delta\tau}{(\Delta t)^2} L q_{j+1} = p_j \end{cases}$$

$$\Leftrightarrow \begin{cases} \left[(1+\gamma\Delta\tau)I_n - 2\left(\frac{\Delta\tau}{\Delta t}\right)^2 L\right] p_{j+1} = p_j + \frac{2\Delta\tau}{(\Delta t)^2} L(q_j + b) \\ \left[(1+\gamma\Delta\tau)\, I_n - 2\left(\frac{\Delta\tau}{\Delta t}\right)^2 L\right] q_{j+1} = \Delta\tau p_j + (1+\gamma\Delta\tau)\,(q_j + b) \end{cases}$$

$$\Leftrightarrow \begin{cases} p_{j+1} = \left[(1+\gamma\Delta\tau)\, I_n - 2\left(\frac{\Delta\tau}{\Delta t}\right)^2 L\right]^{-1} \left[p_j + \frac{2\Delta\tau}{(\Delta t)^2} L(q_j + b)\right] \\ q_{j+1} = \left[(1+\gamma\Delta\tau)\, I_n - 2\left(\frac{\Delta\tau}{\Delta t}\right)^2 L\right]^{-1} \left[\Delta\tau p_j + (1+\gamma\Delta\tau)\,(q_j + b)\right] \end{cases}$$

Where $b(n) = \frac{h}{\gamma k^2}$

Result: For $\gamma = 6, p = 0$, we have the algorithm reaches optimal solution at around 1.75 second in simulation time. After tuning p, we reach the optimal solution in 1.2 second at $\gamma = 10, p = 5$. Surprisingly, this is not better than the simple implicit method that can reach the optimal solution in just 0.5 second.

4 Appendix

Changelog

- 25/06/2018: Completed task 1 and task 2
- 28/06/2018: Add gradient descent with momentum part

Questions

- What should we put as p in the gradient descent method with momentum
- How do we apply stability analysis for two dynamics
- How to choose proper γ
- How to prove that gradient descent is GDwM with infinite friction
- How to address the 4-th order derivative FD method without a 4-th condition.

Answers

- It's an art
- Necessary conditions are eigenvalues < 1. Keywords: Algebraic multiplicity, geometric multiplicity.
- It's also an art
- Langevin dynamics is sth like Newtonian + friction.
- Just pick a random 4th condition and let it run.