APPLICATION OF THE STEEPEST DESCENT FOR THE CONTROL OF A LINEAR FINITE-DIMENSIONAL SYSTEM.

E. Zuazua, 17.07.2021

The system

$$(1)\qquad \begin{cases} x' + Ax = Bu, & 0<t<T \\ x(0) = x_0 \end{cases}$$

is controllable when

(2) (A, B) fulfills the Kalman rank condition.

This means that, given $T>0$, $x_0 \in \mathbb{R}^n$ and $x_T \in \mathbb{R}^n$ there is a control $u \in L^2(0,T;\mathbb{R}^m)$ such that the solution of (1) satisfies

(3) $\quad x(T) = x_T.$

We consider the adjoint system

(4) $$\begin{cases} -\varphi' + A^*\varphi = 0 \\ \varphi(T) = \varphi_T. \end{cases} \quad , \quad 0 < t < T$$

We have that $u \in L^2(0,T;\mathbb{R}^m)$ is a successful control satisfying (3) if

(5) $$\int_0^T u(t) \cdot B^*\varphi \, dt = \langle x_T, \varphi_T \rangle - \langle x_0, \varphi(0) \rangle.$$

This means that the control u can be found as the minimum of the functional

(6) $$J(\varphi_T) = \frac{1}{2}\int_0^T |B^*\varphi(t)|^2 dt - \langle x_T, \varphi_T \rangle + \langle x_0, \varphi(0) \rangle.$$

By the DMCV the functional has a unique minimizer $\hat{\varphi}_T$ and the control is

then

$$u = B^* \varphi \tag{7}$$

where φ is the solution of (4) corresponding to the minimizer φ_T.

Let us implement the Gradient Steepest Descent Algorithm for the computation of this control that, furthermore, we know it is the control of minimal $L^2(0,T;\mathbb{R}^m)$-norm.

The algorithm reads

Step 1. Initialization.

Consider any possible $\tilde{\varphi}_T \in \mathbb{R}^n$. For instance $\tilde{\varphi}_T = 0$ and set

$$\varphi_T^0 = \tilde{\varphi}_T. \tag{8}$$

Step 2. Iteration : $k \geq 0$.

$$\varphi_T^{k+1} = \varphi_T^k - \rho \nabla J(\varphi_T^k). \tag{9}$$

We know that the algorithm converges provided $0 < \rho < 1$ is small enough. The smallness condition depends on the conditioning of the quadratic form involved in the functional J, i.e. on the constants $\alpha > 0$ and $M > 0$ such that

$$(10)\quad \alpha \|\varphi_T\|^2 \leq \int_0^T |B^* \varphi|^2 dt \leq M \|\varphi_T\|^2, \quad \forall \varphi_T \in \mathbb{R}^n$$

that we know holds thanks to the Kalman rank condition.

As we have discussed, in practice, the convergence can be accelerated choosing ρ depending on the step $k \geq 0$.

Our goal here is to explain how the iterative algorithm needs to be implemented in practice.

Observe that, according to the previous

developments the optimal control is characterized by the following Optimality System (OS):

$$(11)\quad \begin{cases} x' + Ax = BB^*\varphi, & 0 < t < T \\ x(0) = x_0 \\ -\varphi' + A^*\varphi = 0, & 0 < t < T \\ \varphi(T) = \varphi_T \\ x(T) = x_T \end{cases}$$

This system does not constitute a Cauchy problem. It is built out of the combination of the couplig of the forward state equation for $x = x(t)$ and the backward adjoint equation for $\varphi = \varphi(t)$.

Solving this OS requires the implementation of an iterative algorithm. The Gradient Descent Algo for the minimization of J does precisely

Hot!

Observe that

$$\langle \nabla J(\psi_T), \psi_T \rangle = \int_0^T B^*\varphi \cdot B^*\psi \, dt - \langle x_T, \psi_T \rangle - \langle x_0, \psi(0) \rangle$$

for any other solution $\psi = \psi(t)$ of the adjoint system.

But

$$\int_0^T B^*\varphi \cdot B^*\psi \, dt = \langle x(T), \psi_T \rangle - \langle x_0, \psi(0) \rangle.$$

Thus

$$\langle \nabla J(\varphi_T), \psi_T \rangle = \langle x(T) - x_T, \psi_T \rangle.$$

This means

$$\nabla J(\varphi_T) = x(T) - x_T. \tag{12}$$

In other words, $\nabla J(\varphi_T)$ coincides with the residual in the controllability target

$$x(T) - x_T = \varepsilon_T.$$

This is consistent with the fact that $\nabla J(\hat{\varphi}_T) = 0$, i.e. the minimizer of J leads to the control such that the residual $\varepsilon_T = 0$ and, accordingly,

$$x(T) = x_T.$$

Accordingly, the iterative algorithm reads

$$\varphi_T^{k+1} = \varphi_T^k - \rho\,(x_k(T) - x_T)$$

where $x_k(t)$ is the solution of

$$\begin{cases} x' + Ax = BB^*\varphi_k & 0 < t < T \\ x(0) = x_0. \end{cases}$$

Overall, in each iteration $k \geq 0$ we proceed in several steps:

Step 1. Solve the adjoint system

$$\begin{cases} -\varphi' + A^*\varphi = 0, & 0<t<T \\ \varphi_k(T) = \varphi_k^T. \end{cases}$$

This leads to $\varphi_k(t)$.

Step 2. Solve the state equation

$$\begin{cases} x' + Ax = BB^*\varphi_k, & 0<t<T. \\ x(0) = x_0 \end{cases}$$

This leads to the state $x_k(t)$.

Step 3. Computation of the residual

$$\varepsilon_k = x_k(T) - x_T.$$

Step 4. Adjust the value of the adjoint at time $t = T$:

$$\varphi_T^{k+1} = \varphi_T^k - \rho(x_k(T) - x_T).$$

Essentially each step of the gradient iteration involves solving first the adjoint equation and then the state equation once.

Many variants can be implemented in practice:

- Use variable step: $\rho \to \rho_k$.
- Conjugate gradient algorithm.
- Multi-step algorithms:

$$\varphi_T^{k+1} = \varphi_T^k - \rho \nabla J(\varphi_T^{k-1}).$$

- Two-grid algorithms: Use a fine grid Δt to solve the state equation for x but a coarse grid to solve the adjoint equation for φ. For instance $p\Delta t$ with some $p \geq 1$.
- Random Bath Method to reduce the computational cost of solving the system for x and φ when dimension is large. This leads to algorithms similar to the Stochastic gradient Algorithm.