Greedy Algorithm for Parameter Dependent Operator Lyapunov Equations. Application to Control Problems

Martin Lazar, Jerome Weston
University of Dubrovnik

ConDyS Workshop
Dubrovnik, June 2021

Table of Contents

We consider a family of parameter-dependent operator Lyapunov equations

$$A_\nu P_\nu + P_\nu A_\nu^* = -Q_\nu \qquad (OLE_\nu)$$

- ν – a parameter ranging over compact set $\mathcal{N} \subseteq \mathbf{R}^d$
- A_ν – an unbounded operator on a Hilbert space X
- Q_ν – a bounded operator on X, $Q_\nu \geq 0$
- P_ν – the solution

The goal:

- to find an efficient algorithm for solving (OLE_ν) for a wide range of parameters.

Assumptions

For each ν

- $D(A_\nu)$ is dense in X
- the operator A_ν is closed and stable

Then there exists a unique nonnegative solution $P_\nu \in \mathcal{L}(X)$

$$P_\nu = \int_0^\infty e^{tA_\nu} Q_\nu e^{tA_\nu^*} dt$$

Different methods for computing the solution.

- Bartels, Stewart *Comm. ACM*, 1972. - the Schur decomposition
- Saad (1990) - Krylov suvspace methods
- Simoncini *SIAM Rev.*, 2016. - iterative methods

Computational expensive.

Can we construct the **solution manifold**

$$\mathcal{P} = \{P_\nu : \nu \in K\}$$

without applying the above methods for each new value of ν?

Greedy algorithms

In order to achieve the goal we rely on greedy algorithms:

- one of the most popular reduced bases methods,
- introduced and analysed two decades ago with the aim of solving parametric PDEs[1,2],
- later applied to optimal control problems[3], and to controllability of linear systems[4].

[1] K. Veroy, C. Prud'Homme, D. Rovas, A. Patera, A Posteriori Error Bounds for Reduced-Basis Approximation ..., 16th AIAA Computational Fluid Dynamics Conference, 2003, Orlando, United States.

[2] A. Cohen, R. DeVore, Approximation of high-dimensional parametric PDEs, *Acta Numer.*, 2015.

[3] M. Kärcher, M. A. Grepl, A certified reduced basis method for parametrized elliptic optimal control problems, *ESAIM COCV*, 2014.

[4] M. Lazar, E. Zuazua, Greedy controllability of nite dimensional linear systems, *Automatica*, 2016.

The pure greedy method

X – a Banach space $K \subset X$ – a compact subset.

- The method approximates K by a a series of finite dimensional linear spaces V_n (a linear method).
- Offline procedure generates approximation subspace within given precision error; Online routine calculates approximations for any element in K.

The algorithm

The first step Choose $x_1 \in K$ such that

$$\|x_1\|_X = \max_{x \in K} \|x\|_X .$$

The general step Having found $x_1..x_n$, denote $V_n = \operatorname{span}\{x_1, \dots, x_n\}$. Choose the next element

$$x_{n+1} := \arg\max_{x \in K} \operatorname{dist}(x, V_n) . \tag{1}$$

The algorithm stops when the greedy approximation error

$$\sigma_n(K) := \max_{x \in K} \operatorname{dist}(x, V_n)$$

becomes less than the given tolerance ε.

Efficiency

In order to estimate **the efficiency of the (weak) greedy algorithm** we compare its approximation rates $\sigma_n(K)$ with the best possible one.

The Kolmogorov n width, $d_n(K)$

– measures how well K can be approximated by a subspace in X of a fixed dimension n.

$$d_n(K) := \inf_{\dim Y = n} \sup_{x \in K} \inf_{y \in Y} \|x - y\|_X .$$

Thus $d_n(K)$ represents optimal approximation performance that can be obtained by a n-dimensional linear space.
The greedy approximation rates have same decay as the Kolmogorov widths.

Theorem[1]

For any $\alpha > 0, C_0 > 0$

$$d_n(K) \leq C_0 n^{-\alpha} \quad \Longrightarrow \quad \sigma_n(K) \leq C_1 n^{-\alpha}, \quad k \in \mathbf{N},$$

where $C_1 := C_1(\alpha, C_0, \gamma)$.

[1] A. Cohen, R. DeVore, *Acta Numerica*, 2015.

Performance obstacles

- The set K in general consists of infinitely many vectors.
- In practical implementations the set K is often unknown (e.g. it represents the family of solutions to parameter dependent problems).

Performance obstacles

- The set K in general consists of infinitely many vectors.
  Finite discretisation of K.
- In practical implementations the set K is often unknown (e.g. it represents the family of solutions to parameter dependent problems).

Performance obstacles

- The set K in general consists of infinitely many vectors.
  Finite discretisation of K.
- In practical implementations the set K is often unknown (e.g. it represents the family of solutions to parameter dependent problems).
  One uses some **surrogate** value replacing the exact distance by some uniformly equivalent term.

Performance obstacles

- The set K in general consists of infinitely many vectors.
  Finite discretisation of K.
- In practical implementations the set K is often unknown (e.g. it represents the family of solutions to parameter dependent problems).
  One uses some **surrogate** value replacing the exact distance by some uniformly equivalent term.

Practical realisations depends crucially on construction of an appropriate surrogate!

Implementation: Residual Analysis

Knowing P_1 how to measure

$$\mathrm{dist}(P_1 - P_\nu)$$

without knowing P_ν?

Check residual

$$R_\nu(P_1) := A_\nu P_1 - P_1 A_\nu + B_\nu B_\nu^*$$

Implementation: Residual Analysis

Knowing P_1 how to measure

$$\mathrm{dist}(P_1 - P_\nu)$$

without knowing P_ν?

Check residual

$$R_\nu(P_1) := A_\nu P_1 - P_1 A_\nu + B_\nu B_\nu^*$$

Theorem

Suppose that

1) A_ν *is a generator of an exponentially stable strongly continuous semigroup with a* ν *independent growth bound* $\omega < 0$*;*
2) *Each* A_ν *is diagonalizable*
3) $D(A_{\nu_1}) = D(A_{\nu_2})$ *and* $D(A^*_{\nu_1}) = D(A^*_{\nu_2})$ *for* $\nu_1,\ \nu_2 \in \mathcal{N}$*.*

Then

$$||R_\nu(P_1)|| \sim ||P_1 - P_\nu||$$

Implementation: Residual Analysis

Knowing P_1 how to measure

$$\text{dist}(P_1 - P_\nu)$$

without knowing P_ν?

Check residual

$$R_\nu(P_1) := A_\nu P_1 - P_1 A_\nu + B_\nu B_\nu^*$$

Theorem

Suppose that

1) A_ν *is a generator of an exponentially stable strongly continuous semigroup with a* ν *independent growth bound* $\omega < 0$*;*
2) *Each* A_ν *is diagonalizable*
3) $D(A_{\nu_1}) = D(A_{\nu_2})$ *and* $D(A^*_{\nu_1}) = D(A^*_{\nu_2})$ *for* $\nu_1, \ \nu_2 \in \mathcal{N}$.

Then

$$||R_\nu(P_1)|| \sim ||P_1 - P_\nu||$$

Tricky part - functional setting (norms in which spaces?)

Result in the finite dimensional setting:

N.T. Son, T. Stykel *Siam J. Matrix Anal. Appl.*, 2017,

Implementation: Residual Analysis

Knowing P_1 how to measure

$$\mathrm{dist}(P_1 - P_\nu)$$

without knowing P_ν?
Check residual

$$R_\nu(P_1) := A_\nu P_1 - P_1 A_\nu + B_\nu B_\nu^*$$

Theorem

Suppose that

1) A_ν *is a generator of an exponentially stable strongly continuous semigroup with a* ν *independent growth bound* $\omega < 0$*;*
2) *Each* A_ν *is diagonalizable*
3) $D(A_{\nu_1}) = D(A_{\nu_2})$ *and* $D(A_{\nu_1}^*) = D(A_{\nu_2}^*)$ *for* $\nu_1,\ \nu_2 \in \mathcal{N}$.

Then

$$||R_\nu(P_1)||_{\mathcal{L}(X_1^d, X_{-1})} \sim ||P_1 - P_\nu||_{\mathcal{L}(X_1^d, X)}$$

X_1^d - the set $D(A_\nu^*)$ equipped with the norm

$$||x||_{1,d} \quad = \quad ||(\beta I - A_\nu^*)x||_X, \qquad x \in D(A_\nu^*),\ \beta \in \rho(A_\nu^*)$$

Implementation: Residual Analysis

Knowing P_1 how to measure

$$\mathrm{dist}(P_1 - P_\nu)$$

without knowing P_ν?
Check residual

$$R_\nu(P_1) := A_\nu P_1 - P_1 A_\nu + B_\nu B_\nu^*$$

Theorem

Suppose that

1) A_ν *is a generator of an exponentially stable strongly continuous semigroup with a* ν *independent growth bound* $\omega < 0$*;*
2) *Each* A_ν *is diagonalizable*
3) $D(A_{\nu_1}) = D(A_{\nu_2})$ *and* $D(A_{\nu_1}^*) = D(A_{\nu_2}^*)$ *for* $\nu_1, \ \nu_2 \in \mathcal{N}$.

Then

$$||R_\nu(P_1)||_{\mathcal{L}(X_1^d, X_{-1})} \sim ||P_1 - P_\nu||_{\mathcal{L}(X_1^d, X)}$$

Collateral result:

Theorem

Lyapunov operator $L_A(P) = AP + PA^*$ *is a bounded and coercive operator from* $\mathcal{L}(X_1^d, X)$ *to* $\mathcal{L}(X_1^d, X_{-1})$.

Control problem

Consider the control system

$$\begin{cases} \frac{d}{dt}x(t) & = & Ax(t) + Bu(t), \qquad 0 \leq t \leq T \\ x(0) & = & x_0 \end{cases}$$

where B is an admissible control operator.
Suppose that x_T is a reachable state.
Then the optimal norm control $\hat{u}$ is of the type

$$\hat{u} = B^* e^{(T-t)A^*} \phi_T$$

for some vector ϕ_T which corresponds to initial datum of the adjoint equation.
In addition, the following equation holds

$$x_T - e^{tA} x_0 = \Lambda_T \phi_T,$$

where Λ_T is the Gramian operator

$$\Lambda_T = \int_0^T e^{tA} B B^* e^{tA} dt$$

The minimal control energy is given by

$$\|\hat{u}\|^2 = \Lambda_T \phi_T \cdot \phi_T.$$

For dissipative systems Λ_T can be well approximated by the infinite time Gramian operator:

$$\Lambda_\infty = \int_0^\infty e^{tA} BB^* e^{tA} dt,$$

For dissipative systems Λ_T can be well approximated by the infinite time Gramian operator:

$$\Lambda_\infty = \int_0^\infty e^{tA} BB^* e^{tA} dt,$$

which is the solution to (OLE) with $Q = BB^*$
Solving for Λ_∞ is much easier than constructing Λ_T (which satisfies differential Lyapunov equation).

But we even want to avoid solving for Λ_∞ !

For dissipative systems Λ_T can be well approximated by the infinite time Gramian operator:

$$\Lambda_\infty = \int_0^\infty e^{tA} BB^* e^{tA} dt,$$

which is the solution to (OLE) with $Q = BB^*$
Solving for Λ_∞ is much easier than constructing Λ_T (which satisfies differential Lyapunov equation).

But we even want to avoid solving for Λ_∞ !

We introduce parameter dependence

$$\begin{cases} \frac{d}{dt} x_\nu(t) & = & A_\nu x_\nu(t) + B_\nu u_\nu(t), \qquad 0 \leq t \leq T \\ x_\nu(0) & = & x_{0,\nu} \end{cases}$$

We apply the greedy algorithm for solving (approximately) $\Lambda_{\infty,\nu}$

The algorithm is independent of $x_{0,\nu}, x_{T,\nu}$ and T!

Example 1: 1D Heat Equation

$$\begin{cases} \frac{\partial}{\partial t} z - \nu \Delta z = 0 & \text{in} \quad (0,1) \times (0,T), \\ z(0,t) = 0, & z(1,t) = u_\nu(t), \\ z(x,0) = z_0. & \end{cases}$$

The parameter ν ranges within $\mathcal{N} = [0.7, 1300]$

The greedy algorithm has been applied with

- semi-discretized system of dimension $N = 40$,
- $\epsilon = 0.01$,
- uniform discretization of $\mathcal{N}$ in $l = 100$.

The offline algorithm stops **after only one** iteration in approximately 0.06 seconds!

Example 1: 1D Heat Equation

$$\begin{cases} \frac{\partial}{\partial t} z - \nu \Delta z = 0 & \text{in} \quad (0,1) \times (0,T), \\ z(0,t) = 0, & z(1,t) = u_\nu(t), \\ z(x,0) = z_0. \end{cases}$$

The parameter ν ranges within $\mathcal{N} = [0.7, 1300]$

The greedy algorithm has been applied with

- semi-discretized system of dimension $N = 40$,
- $\epsilon = 0.01$,
- uniform discretization of $\mathcal{N}$ in $l = 100$.

The offline algorithm stops **after only one** iteration in approximately 0.06 seconds!

By change of variables:

$$A_\nu = \nu A \implies \Lambda_{\infty,\nu} = \nu \Lambda_\infty$$

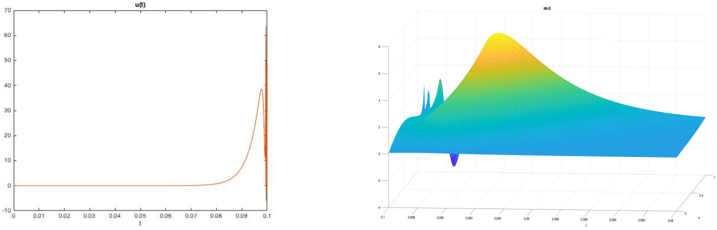
(Holds just for $T = \infty$!)

Example 1: 1D Heat Equation - Online part

We aim to steer the system

- from $z_0 = 0$ to $z_1 = \sin(\pi x)$
- in time $T = 0.1$
- for $\nu = 23$

Calculation of the approximate Gramian is rather straightforward.
It is applied for construction of the optimal control.
It drives the system to final state z^1 within the error $|z^1 - z(T)| = 3.77 \times 10^{-5}$.



a) b)

Figure: Evolution of a) the approximate control and b) the solution of semi-discretized example problem.

Example 2: Anisotropic 2D Heat Equation

$$\begin{array}{lll} \frac{\partial}{\partial t} z - \Delta_\nu z = 0 & \text{in} & (0,1)^2 \times (0,T), \\ z(x,t) = v_0(x,t), & \text{for} & x \in \partial([0,1]^2) \\ z(x,0) = 0 & & \end{array}$$

- $\Delta_\nu = \frac{\partial^2}{\partial x_1^2} + (1+\nu)\frac{\partial^2}{\partial x_2^2}, \qquad \nu \in \mathcal{N} = [0,1]$
- $$v_0(x,t) = \begin{cases} u_\nu(t), & x_1 = 1 \\ 0, & \text{otherwise} \end{cases}$$

The greedy algorithm has been applied with $\epsilon = 0.05$ and the uniform discretization of $\mathcal{N}$ in $l = 100$.

N	offline run time (s)	number of iterations	online run time (s)	$\|\|\Lambda_\nu - \Lambda_\nu^\star\|\|_2$
100	110	15	0.03	1.2e-7
900	7820	16	2.58	2.1e-7
10 000	40018	16	6.21	4.0e-7

Table: Basic numerical indicators corresponding to different system dimensions N. $\nu = 0.1$.

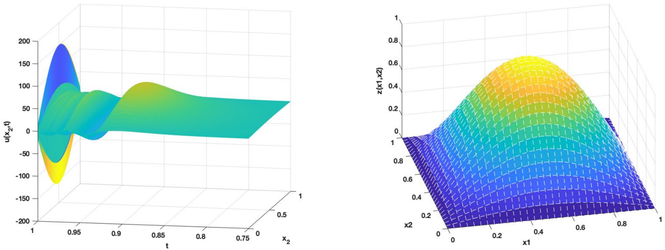
Example 2: Anisotropic 2D Heat Equation - Online part

We aim to steer the system

- from $z_0 = 0$ to $z_1 = \sin(\pi x_1) * \sin(\pi x_2)$
- in time $T = 1$
- for $\nu = 0.1$
- and $N = 10\,000$.

$\Lambda_{\infty,\nu}$ is approximated by a suitable linear combination of $\Lambda_{\infty,i}, i = 1..16$.

Elapsed time is 6.21 s and the error is $|z^1 - z(T)| = 9.7$e-3.



a) b)

Figure: a) Evolution of the approximate control and b) the states $z(T)$ (dashed) and z^1 for $\nu = 0.1$ and $N = 10\,000$ at $T = 1$.

Conclusion

Done:

- Greedy algo for solving parameter dependent OLE
- Provides approximation of infinite time control Gramians (independent of initial and final data, and final time!)
- Enables construction of optimal controls for dissipative systems

Further work:

- Differential Lyapunov equation
- It would provides approximation of finite time control Gramians
- Enables construction of optimal controls for non-dissipative systems

Conclusion

Done:

- Greedy algo for solving parameter dependent OLE
- Provides approximation of infinite time control Gramians (independent of initial and final data, and final time!)
- Enables construction of optimal controls for dissipative systems

Further work:

- Differential Lyapunov equation
- It would provides approximation of finite time control Gramians
- Enables construction of optimal controls for non-dissipative systems

Thanks for your attention!