1 Exponentially asymptotically autonomous systems

Let $\sigma \in \mathbb{N}$. Consider the exponentially asymptotically autonomous equation $\dot{x} = f(t,x)$, $x \in \mathbb{R}^n$, where $f, g \in C^\sigma$ and $f(t,x) \to g(x)$ in the following sense: for each compact set $K \subset \mathbb{R}^n$ and each $\epsilon > 0$ there is a T with

$$\|\partial^\gamma f(t,x) - \partial^\gamma g(x)\| \, e^{\alpha t} \quad < \quad \epsilon \text{ for all } \gamma \text{ with } |\gamma| \leq \sigma,$$

for all $x \in K$ and all $t > T$. Let $x(t) = 0$ be an exponentially stable solution.

Time transformation

Choose $0 < \beta \leq \frac{\alpha}{\sigma}$ and define $h(\tau) = -\frac{1}{\beta} \ln |\tau|$. The transformed system is

$$\begin{aligned} \dot{\tau} &= -\beta\tau \\ \dot{x} &= \begin{cases} f(-\frac{1}{\beta}\ln|\tau|, x) & \text{if } \tau \neq 0 \\ g(x) & \text{if } \tau = 0 \end{cases} \end{aligned}$$

Local Lyapunov function

Solve the matrix equation

$$\begin{pmatrix} -\beta & 0 \\ 0 & Dg(0) \end{pmatrix} Q^T + Q \begin{pmatrix} -\beta & 0 \\ 0 & Dg(0) \end{pmatrix} = -I$$

for $Q \in \mathbb{R}^{(n+1)\times(n+1)}$. Set

$$v_{loc}(\tau, x) = (\tau, x)\, Q \begin{pmatrix} \tau \\ x \end{pmatrix}.$$

Find R^* such that all points in

$$E := \{(\tau, x) \in (-\infty, 0] \times \mathbb{R}^n \mid v_{loc}(\tau, x) \leq R^*\}$$

except $(\tau, x) = (0, 0)$ satisfy $v'_{loc}(\tau, x) < 0$.

Radial Basis Functions

Choose a Radial Basis Function ϕ. For $n = 1$ or $n = 2$ we choose $c > 0$ and
$$\phi(r) = \psi_{4,2}(c \cdot r) = \begin{cases} (1 - cr)^6[35c^2r^2 + 18cr + 3] & \text{for } r \leq \frac{1}{c} \\ 0 & \text{otherwise.} \end{cases}$$

Choose grid points $X \subset (-\infty, 0] \times \mathbb{R}^n$ approximately in the expected basin of attraction – do not include $(0,0)$.

Find R such that all points in

$$K := \{(\tau, x) \in (-\infty, 0] \times \mathbb{R}^n \mid v(\tau, x) \leq R\}$$

satisfy $v'(\tau, x) < 0$ or $(\tau, x) \in \overset{\circ}{E}$.

Then K is a subset of the basin of attraction.

Transform K back:

$$\tilde{K} := \left\{(t, x) \in (-\infty, \infty) \times \mathbb{R}^n \mid v(-e^{-\beta t}, x) \leq R\right\}$$

Matlab Files

The plots for the Matlab files are designed for dimension $n = 1$.

- **f_fun.m** is the right-hand side $f(t, x)$ of the original system.
- **g_fun.m** is the right-hand side $g(x)$ of the limiting system.
- **F.m** is the right-hand side $F(\tau, x)$ of the transformed system, depending on the transformation and in particular β.
- **hinv.m** is the inverse of the transformation h, i.e. $h^{-1}(t) = -e^{-\beta t}$.
- **locv.m** is the local Lyapunov function $l_{loc}(\tau, x)$.
- **gradlocv.m** is the gradient of the local Lyapunov function $l_{loc}(\tau, x)$ with respect to τ and x.
- **plotlocv.m**

  plotlocv(T,MAX,tnumber,xnumber,valloc),

  plots the level set $v_{loc}(\tau, x) =$valloc in Figure 6 and 3 and the function $v_{loc}(\tau, x)$ depending on (τ, x) in Figure 1 on a grid in $[-T, 0] \times [-MAX, MAX]$ where the time interval is divided into tnumber equal steps, and the space interval is divided into 2 xnumber equal steps.
- **plotlocvs.m**

  plotlocvs(T,MAX,tnumber,xnumber,beta),

  plots the level set $v'_{loc}(\tau, x) = 0$ in Figure 3 and the function $v'_{loc}(\tau, x)$ depending on (τ, x) in Figure 2 on a grid in $[-T, 0] \times [-MAX, MAX]$ where the time interval is divided into tnumber equal steps, and the space interval is divided into 2 xnumber equal steps.

- **phi.m, phi1.m, phi2.m**

  phi(r) is the Radial Basis Function $\psi_{4,2}(c \cdot r)$, phi1 is defined by $\frac{d\,phi}{dr}/r$, and phi2 is defined by $\frac{d\,phi1}{dr}/r$

- **coefficients.m**

  [alpha,points,points0,pointsneg,N,N0]=coefficients(T,tN,xN,xdist,c,n,beta)

  calculates the coefficients alpha for the approximation v

  - $T > 0$ and $[-T, T]$ is the time interval
  - tN denotes the number of steps in time direction from $-T$ to 0
  - xdist is the distance of a step in x-direction
  - xN denotes the number of steps in positive x-direction
  - c denotes the constant in the RBF
  - n is the dimension
  - points are the points with positive τ-value (τ_k, x_k) in $(0, T] \times \mathbb{R}^n$, N points
  - pointsneg are the points with negative value, also N points
  - points0 are the points in $\{0\} \times \mathbb{R}^n$, N0 points
  - points, pointsneg and points0 together form the grid X

- **v.m** [v]=v(c,alpha,points,points0,pointsneg,beta,t,x)

  calculates the value of the approximation v at point (t, x)

- **vs.m** [vs]=vs(c,alpha,points,points0,pointsneg,beta,t,x)

  calculates the orbital derivative v' of approximation v at (t, x)

- **plotv.m**

  plotv(T,MAX,c,n,beta,alpha,points,points0,pointsneg,tnumber,xnumber,val),

  plots the level set $v(\tau, x)$ =val in Figure 6 and the function $v(\tau, x)$ depending on (τ, x) in Figure 4 on a grid in $[-T, 0] \times [-MAX, MAX]$ where the time interval is divided into tnumber equal steps, and the space interval is divided into 2 xnumber equal steps.

- **plotvs.m**

  plotv(T,MAX,c,n,beta,alpha,points,points0,pointsneg,tnumber,xnumber),

  plots the level set $v'(\tau, x) = 0$ in Figure 6 and the function $v'(\tau, x)$ depending on (τ, x) in Figure 5 on a grid in $[-T, 0] \times [-MAX, MAX]$

where the time interval is divided into tnumber equal steps, and the space interval is divided into 2 xnumber equal steps.

- **plotvorg.m**

  plotvorg(T1,T2,MAX,c,n,beta,alpha,points,points0,pointsneg,tnumber,xnumber,val),

  plots the level set $v(t,x)$ =val in Figure 7 on a grid in $[T1,T2] \times [-MAX, MAX]$ where the time interval is divided into tnumber, and the space interval is divided into 2 xnumber equal steps

- **plotvsorg.m**

  plotvsorg(T1,T2,MAX,c,n,beta,alpha,points,points0,pointsneg,tnumber,xnumber),

  plots the level set $v'(t,x) = 0$ in Figure 7 on a grid in $[T1,T2] \times [-MAX, MAX]$ where the time interval is divided into tnumber, and the space interval is divided into 2 xnumber equal steps

Figures

1. Plot of $v_{loc}(\tau, x)$
2. Plot of $v'_{loc}(\tau, x)$
3. Level set $v'_{loc}(\tau, x) = 0$ (red), $v_{loc}(\tau, x) = valloc$ (green)
4. Plot of $v(\tau, x)$
5. Plot of $v'(\tau, x)$
6. Level set $v'(\tau, x) = 0$ (red), $v(\tau, x) = val$ (black) and $v_{loc}(\tau, x) = valloc$ (green)
7. Level set $v'(t, x) = 0$ (red), $v(t, x) = val$ (black) and $v_{loc}(t, x) = valloc$ (green) in the original time t

What to do

- Choose the constant β and modify f_fun.m, g_fun.m, locv.m and grad-locv.m.
  - **f_fun**: define $f(t, x)$
  - **g_fun**: define $g(x)$

- **locv**: Solve $\begin{pmatrix} -\beta & 0 \\ 0 & Dg(0) \end{pmatrix} Q^T + Q \begin{pmatrix} -\beta & 0 \\ 0 & Dg(0) \end{pmatrix} = -I$. Output is $(\tau, x)Qx$
- **gradlocv** is the gradient of $locv(\tau, x)$ with respect to τ and x

- Use plotlocv.m and plotlocvs.m. Choose the value **valloc** in plotlocv so small that the sublevel set of v_{loc} is in the area where v'_{loc} is negative (Figure 3)
- Choose constants T, tN, xN, xdist, c and n
- Calculate alpha using coefficients.m
- Use plotv.m and plotvs.m. Choose the value **val** in plotv so small that the sublevel set of v is either in the area where v' is negative or which is covered by the sublevel set of v_{loc}
- If not possible, put more points into the grid (tN, xN larger and T,xdist smaller) and/or make area covered smaller (T,xdist smaller) – (Figure 6)
- Use plotvorg.m, plotvsorg.m and plotlocvor.m to obtain a plot with respect to the original time t (Figure 7)

1.1 Example

Example from [1].

```
beta=0.5;valloc=0.5;
plotlocv(1.5,2,54,54,valloc);
plotlocvs(1.5,2,54,54,beta);


T=2;tN=40;xN=11;xdist=0.06;c=2;n=1;%Definition of the constants
[alpha,points,points0,pointsneg,N,N0]=coefficients(T,tN,xN,xdist,c,n,beta);
%Calculation of the vector alpha


val=-0.45;
plotv(1.5,1.5,c,n,beta,alpha,points,points0,pointsneg,54,54,val);
plotvs(1.5,1.5,c,n,beta,alpha,points,points0,pointsneg,104,104);
```

```
plotvorg(-1,5,1.5,c,n,beta,alpha,points,points0,pointsneg,100,70,val);
plotvsorg(-1,5,1.5,c,n,beta,alpha,points,points0,pointsneg,100,70);
plotlocvorg(-1,5,2,54,54,beta,valloc);
```

Acknowledgement This work was supported by the Engineering and Physical Research Council [EP/H051627/1, EP/I000860/1].

References

[1] P. Giesl and H. Wendland, *Numerical determination of the basin of attraction for exponentially asymptotically autonomous dynamical systems*, Nonlinear Anal., **74** (2011), 3191–3203.