Hybrid systems

Sébastien Gros
NTNU, Eng. Cybernetics

Slides for TTK4130
2021

Outline

Hybrid systems

- Continuous and discrete states & dynamics co-exist
- Continuous states $\mathbf{x}(t)$ can jump
- Discontinuities can e.g. occur due to the state meeting some conditions

Hybrid systems

- Continuous and discrete states & dynamics co-exist
- Continuous states $\mathbf{x}(t)$ can jump
- Discontinuities can e.g. occur due to the state meeting some conditions

Simple examples:

- Thermostat
- Bouncing ball
- Gear shifting

Hybrid systems

- Continuous and discrete states & dynamics co-exist
- Continuous states $\mathbf{x}(t)$ can jump
- Discontinuities can e.g. occur due to the state meeting some conditions

Simple examples:

- Thermostat
- Bouncing ball
- Gear shifting

More difficult examples:

- Walk and running dynamics
- Take-off and landing
- Gripping (robots)

Example bouncing ball

Continuous dynamics (unique model):

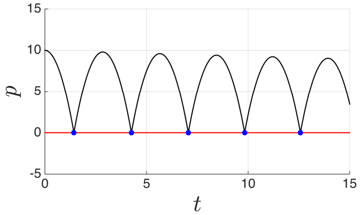
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}, \qquad \dot{\mathbf{x}} = \begin{bmatrix} v \\ -g \end{bmatrix}$$

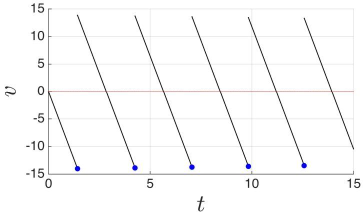
where $p,\ v \in \mathbb{R}$

Discontinuity (state condition):

$$v = -\gamma v \quad \text{if} \quad (p = 0 \quad \& \quad v < 0)$$

produces a "jump" in the continuous state $\mathbf{x}(t)$ when the ground is met





A bit of formalism on hybrid systems

Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

A bit of formalism on hybrid systems

Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)

A bit of formalism on hybrid systems

Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)

A bit of formalism on hybrid systems

Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)

A bit of formalism on hybrid systems

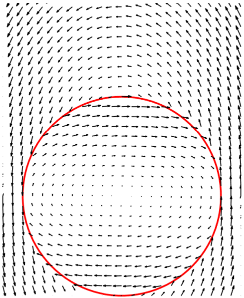
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} \,:\, Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

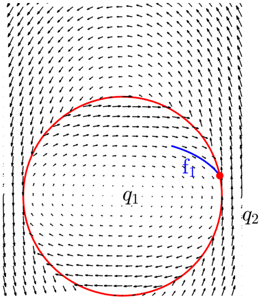
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \dots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

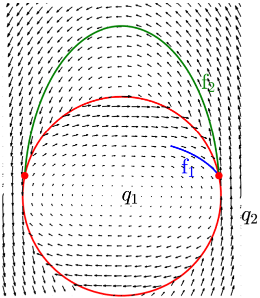
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

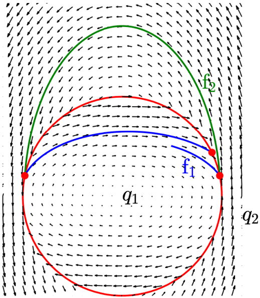
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \quad \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

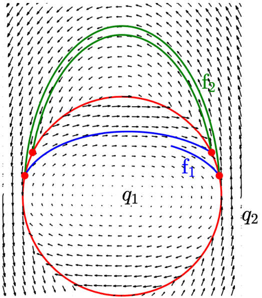
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} \,:\, Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

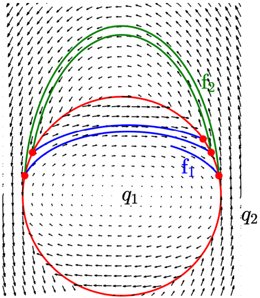
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \dots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



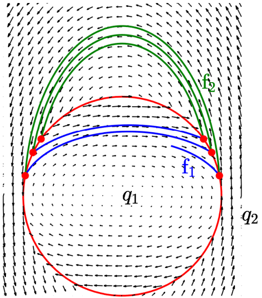
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} \,:\, Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

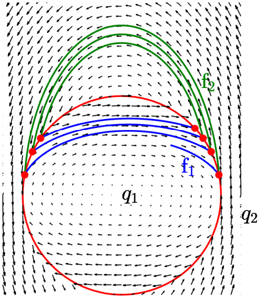
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} \,:\, Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

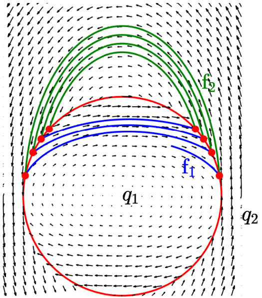
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \dots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

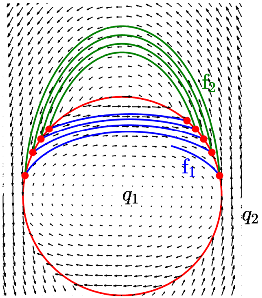
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

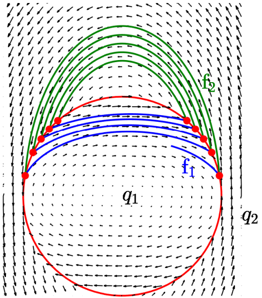
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \dots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

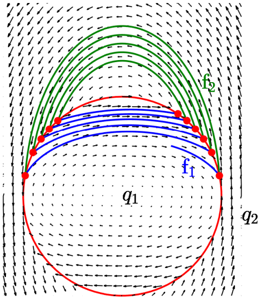
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} : Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

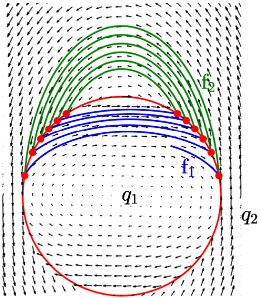
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} \, : \, Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \quad \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems

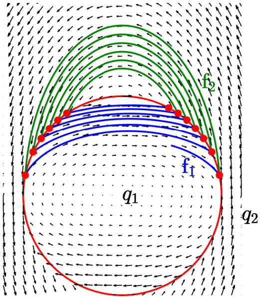
Simple examples of hybrid systems can be easy to describe, but in general they can be very complex and require a clean formalism

- **Discrete states**: $Q = \{q_1, q_2, \ldots\}$ (e.g. gears)
- **Continuous states**: $\mathbf{x} \in X \subseteq \mathbb{R}^n$ (as usual)
- **Inputs**: $\mathbf{u} \in U \subseteq \mathbb{R}^m$ (as usual)
- **Continuous dynamics**: $\mathbf{f} \,:\, Q \times X \times U \mapsto \mathbb{R}^n$, e.g.

$$\mathbf{f}(q, \mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_1 \\ \vdots & \\ \mathbf{f}_n(\mathbf{x}, \mathbf{u}) & \text{if} \quad q = q_n \end{cases}$$

yield

$$\dot{\mathbf{x}} = \mathbf{f}(q, \mathbf{x}, \mathbf{u})$$



A bit of formalism on hybrid systems - Guard Conditions

Guard conditions: $G(.,.)\,:\,Q \times Q \mapsto X_G \subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered

A bit of formalism on hybrid systems - Guard Conditions

Guard conditions: $G(.,.) : Q \times Q \mapsto X_G \subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered

E.g.:

$$G\left(1^{\mathrm{st}}\text{ gear},\ 2^{\mathrm{nd}}\text{ gear}\right) = \{\mathbf{x}\,|\,\mathrm{rpm} > 2000\}$$

means: "we shift from 1^{st} to 2^{nd} when the engine rpm is higher than 2000".

A bit of formalism on hybrid systems - Guard Conditions

Guard conditions: $G(.,.) : Q \times Q \mapsto X_G \subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered

E.g.:

$$G\left(1^{\mathrm{st}} \text{ gear}, \; 2^{\mathrm{nd}} \text{ gear}\right) = \{\mathbf{x} \,|\, \mathrm{rpm} > 2000\}$$

means: "we shift from 1^{st} to 2^{nd} when the engine rpm is higher than 2000".

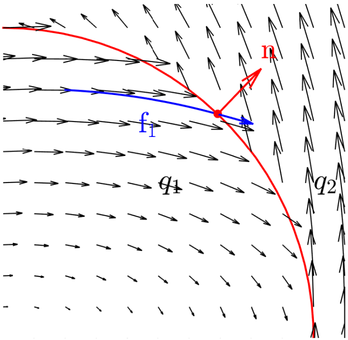
A bit of formalism on hybrid systems - Guard Conditions

Guard conditions: $G(.,.)\,:\,Q\times Q\mapsto X_G\subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered

E.g.:

$$G\left(1^{\text{st}}\text{ gear},\ 2^{\text{nd}}\text{ gear}\right)=\{\mathbf{x}\,|\,\text{rpm}>2000\}$$

means: "we shift from 1^{st} to 2^{nd} when the engine rpm is higher than 2000".



Guard condition:

$$G\left(q_1,q_2\right)=\left\{\mathbf{x}\quad|\quad|\mathbf{x}|=1\quad\text{and}\quad\mathbf{f}_1\left(\mathbf{x}\right)^\top\mathbf{n}>0\right\}$$

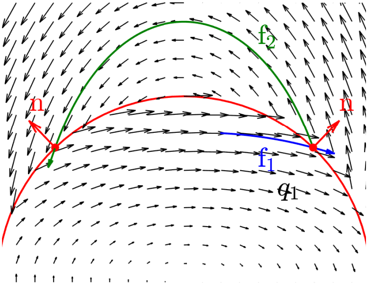
A bit of formalism on hybrid systems - Guard Conditions

Guard conditions: $G(.,.) : Q \times Q \mapsto X_G \subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered

E.g.:

$$G\left(1^{\text{st}} \text{ gear}, \; 2^{\text{nd}} \text{ gear}\right) = \{\mathbf{x} \,|\, \text{rpm} > 2000\}$$

means: "we shift from 1^{st} to 2^{nd} when the engine rpm is higher than 2000".



Guard condition:

$$G(q_1, q_2) = \left\{\mathbf{x} \quad | \quad |\mathbf{x}| = 1 \quad \text{and} \quad \mathbf{f}_1(\mathbf{x})^\top \mathbf{n} > 0\right\}$$
$$G(q_2, q_1) = \left\{\mathbf{x} \quad | \quad |\mathbf{x}| = 1 \quad \text{and} \quad \mathbf{f}_2(\mathbf{x})^\top \mathbf{n} < 0\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.) : Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j

A bit of formalism on hybrid systems (cont')

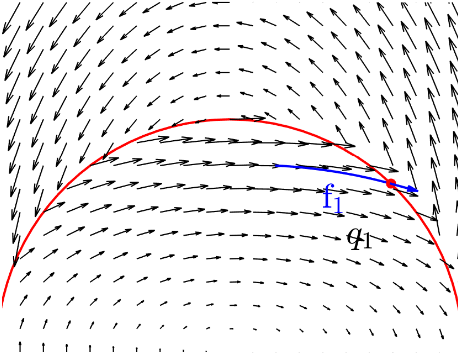
Reset map: $R(.,.) \; : \; Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}} \text{ gear}, \; 2^{\mathrm{nd}} \text{ gear}, \; \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.)\,:\,Q\times Q\times X\mapsto X_R\subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}}\ \text{gear},\ 2^{\mathrm{nd}}\ \text{gear},\ \mathrm{rpm}\right)=\left\{\frac{2}{3}\mathrm{rpm}\right\}$$



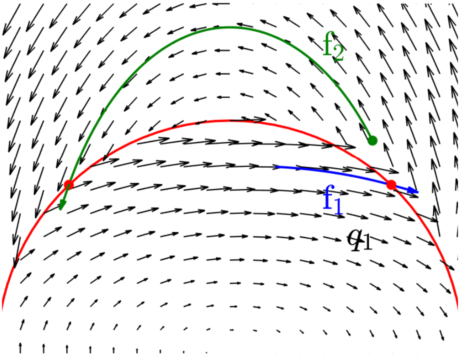
Reset condition:

$$R\left(q_1,q_2,\mathbf{x}\right)=\left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2,q_1,\mathbf{x}\right)=\left\{M_2\mathbf{x}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.) : Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}}\text{ gear},\ 2^{\mathrm{nd}}\text{ gear},\ \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$



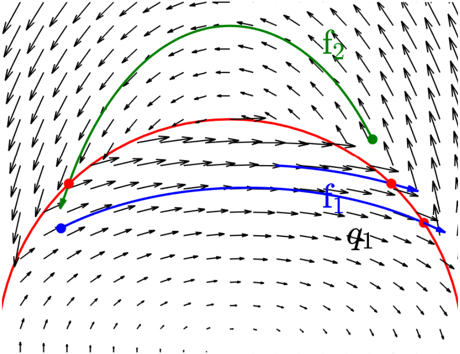
Reset condition:

$$R\left(q_1, q_2, \mathbf{x}\right) = \left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2, q_1, \mathbf{x}\right) = \left\{M_2\mathbf{x}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.) : Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}}\ \text{gear},\ 2^{\mathrm{nd}}\ \text{gear},\ \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$



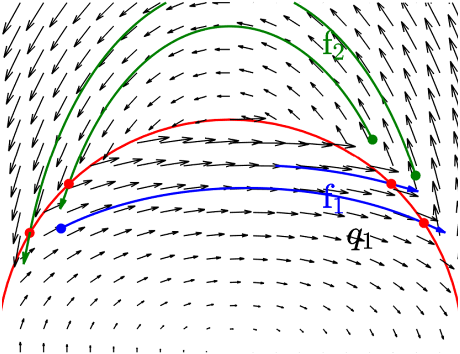
Reset condition:

$$R\left(q_1, q_2, \mathbf{x}\right) = \left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2, q_1, \mathbf{x}\right) = \left\{M_2\mathbf{x}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.)\,:\,Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}}\ \text{gear},\ 2^{\mathrm{nd}}\ \text{gear},\ \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$



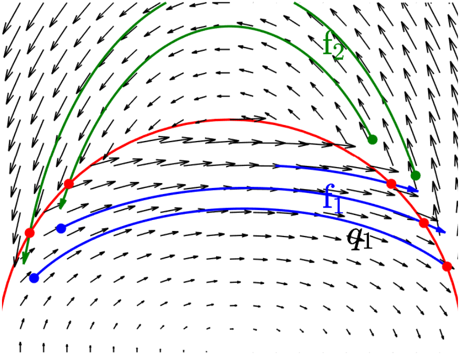
Reset condition:

$$R\left(q_1, q_2, \mathbf{x}\right) = \left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2, q_1, \mathbf{x}\right) = \left\{M_2\mathbf{x}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.) : Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}}\ \text{gear},\ 2^{\mathrm{nd}}\ \text{gear},\ \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$



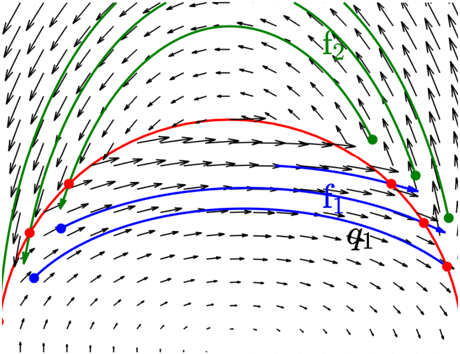
Reset condition:

$$R\left(q_1, q_2, \mathbf{x}\right) = \left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2, q_1, \mathbf{x}\right) = \left\{M_2\mathbf{x}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.) : Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}} \text{ gear},\ 2^{\mathrm{nd}} \text{ gear},\ \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$



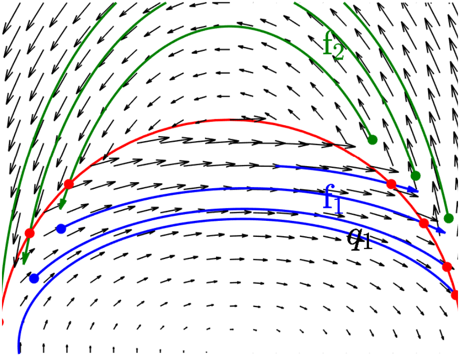
Reset condition:

$$R\left(q_1, q_2, \mathbf{x}\right) = \left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2, q_1, \mathbf{x}\right) = \left\{M_2\mathbf{x}\right\}$$

A bit of formalism on hybrid systems (cont')

Reset map: $R(.,.) : Q \times Q \times X \mapsto X_R \subseteq X$ defines how the continuous states are affected via a jump from a discrete state q_i to a discrete state q_j, e.g. the gear shifting has the reset map:

$$R\left(1^{\mathrm{st}} \text{ gear},\ 2^{\mathrm{nd}} \text{ gear},\ \mathrm{rpm}\right) = \left\{\frac{2}{3}\mathrm{rpm}\right\}$$



Reset condition:

$$R\left(q_1, q_2, \mathbf{x}\right) = \left\{M_1\mathbf{x}\right\}$$
$$R\left(q_2, q_1, \mathbf{x}\right) = \left\{M_2\mathbf{x}\right\}$$

A few remarks

Execution: is the collection of discrete and continuous trajectories resulting from the hybrid system, together with the time intervals

$$I_i = \left[\tau_i, \tau_i'\right]$$

where they apply (i.e. τ_i' denotes a change of discrete state, and $\tau_{i+1} = \tau_i'$)

A few remarks

Execution: is the collection of discrete and continuous trajectories resulting from the hybrid system, together with the time intervals

$$I_i = \left[\tau_i, \tau_i'\right]$$

where they apply (i.e. τ_i' denotes a change of discrete state, and $\tau_{i+1} = \tau_i'$)

Execution can be

- **Finite**: if $\tau = \{I_0, \ldots, I_N\}$ is finite and I_N is closed.

A few remarks

Execution: is the collection of discrete and continuous trajectories resulting from the hybrid system, together with the time intervals

$$I_i = \left[\tau_i, \tau_i'\right]$$

where they apply (i.e. τ_i' denotes a change of discrete state, and $\tau_{i+1} = \tau_i'$)

Execution can be

- **Finite**: if $\tau = \{I_0, \ldots, I_N\}$ is finite and I_N is closed.
- **Infinite**: if $\tau = \{I_0, \ldots\}$ is infinite or if

  $$\sum_{i=0}^{N} \tau_i' - \tau_i = \infty$$

  i.e. the sum of the duration of the intervals is infinite.

A few remarks

Execution: is the collection of discrete and continuous trajectories resulting from the hybrid system, together with the time intervals

$$I_i = \left[\tau_i, \tau_i'\right]$$

where they apply (i.e. τ_i' denotes a change of discrete state, and $\tau_{i+1} = \tau_i'$)

Execution can be

- **Finite**: if $\tau = \{I_0, \ldots, I_N\}$ is finite and I_N is closed.
- **Infinite**: if $\tau = \{I_0, \ldots\}$ is infinite or if

$$\sum_{i=0}^{N} \tau_i' - \tau_i = \infty$$

  i.e. the sum of the duration of the intervals is infinite.
- **Zeno**: if $\tau = \{I_0, \ldots\}$ is infinite but

$$\sum_{i=0}^{N} \tau_i' - \tau_i < \infty$$

Example of Zeno execution

The bouncing ball has a Zeno execution:

$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$

Example of Zeno execution

The bouncing ball has a Zeno execution:

$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$

Example of Zeno execution

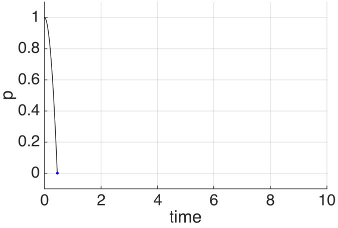
The bouncing ball has a Zeno execution:

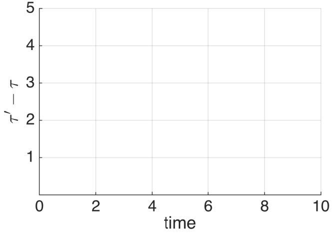
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

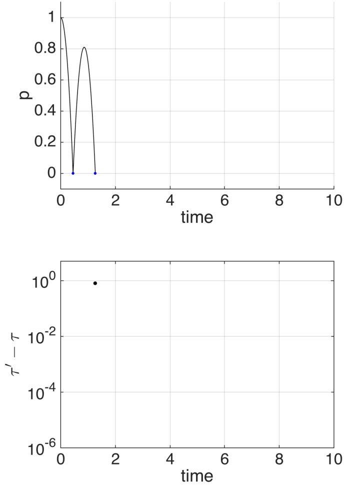
The bouncing ball has a Zeno execution:

$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$



Example of Zeno execution

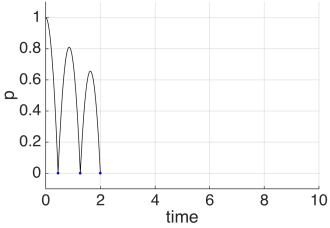
The bouncing ball has a Zeno execution:

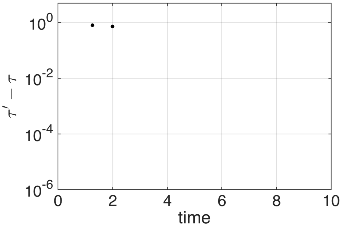
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

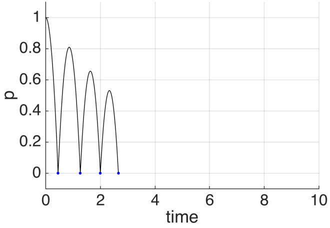
The bouncing ball has a Zeno execution:

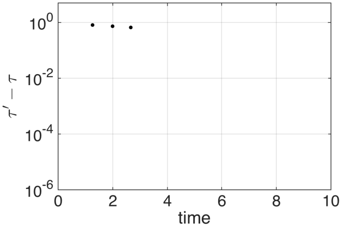
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

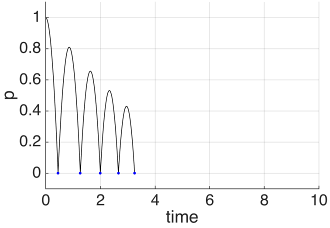
The bouncing ball has a Zeno execution:

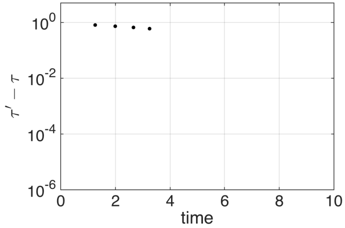
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

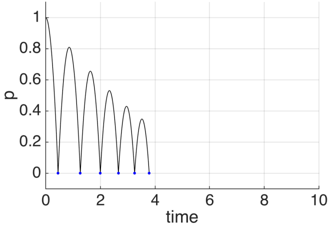
The bouncing ball has a Zeno execution:

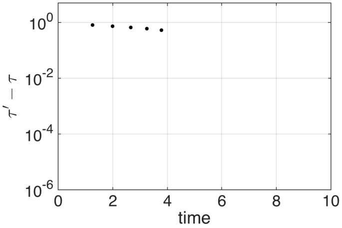
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

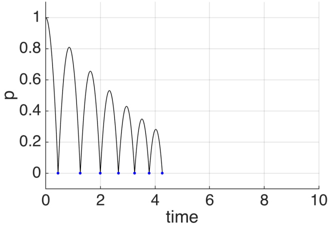
The bouncing ball has a Zeno execution:

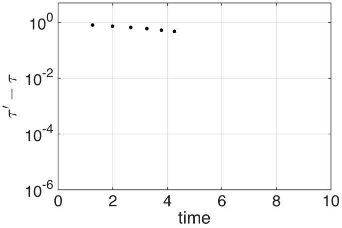
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

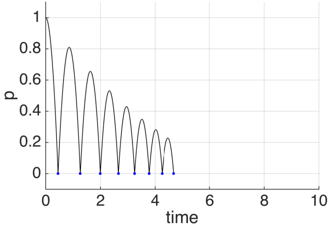
The bouncing ball has a Zeno execution:

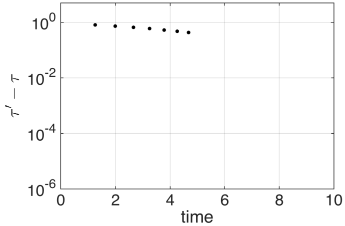
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

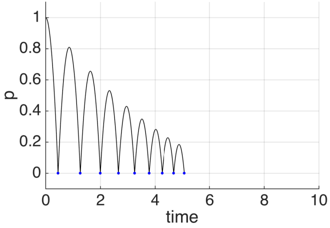
The bouncing ball has a Zeno execution:

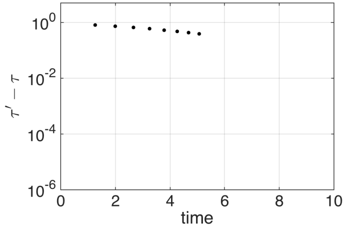
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

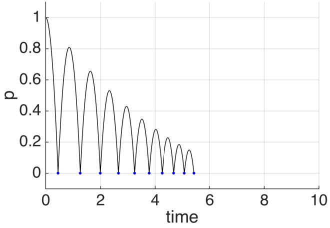
The bouncing ball has a Zeno execution:

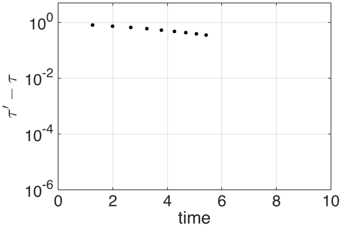
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g}\frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

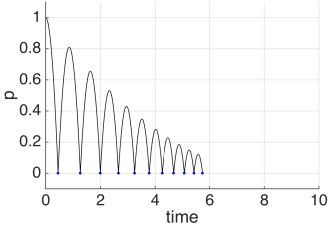
The bouncing ball has a Zeno execution:

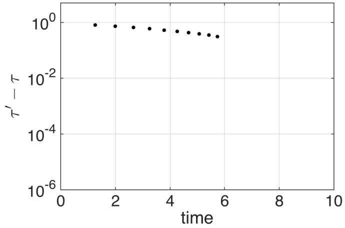
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g}\frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

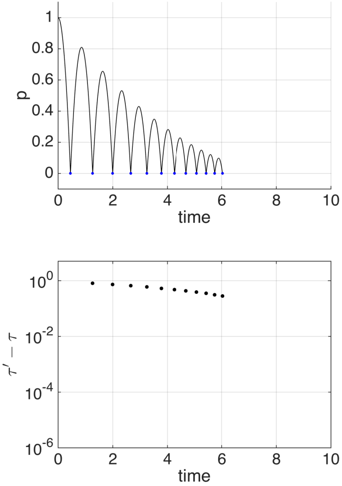
The bouncing ball has a Zeno execution:

$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$



Example of Zeno execution

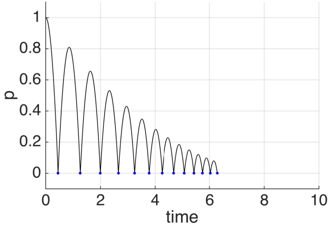
The bouncing ball has a Zeno execution:

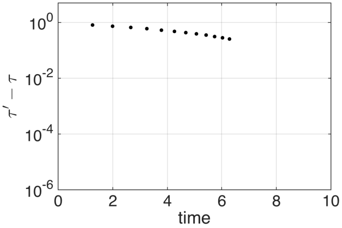
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g}\frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

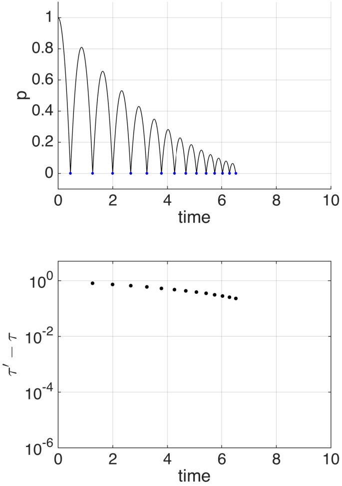
The bouncing ball has a Zeno execution:

$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$



Example of Zeno execution

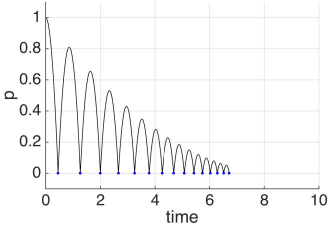
The bouncing ball has a Zeno execution:

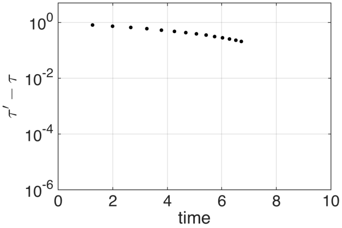
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g}\frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

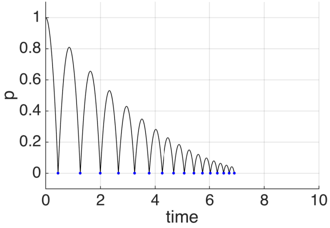
The bouncing ball has a Zeno execution:

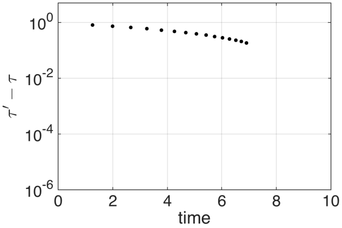
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

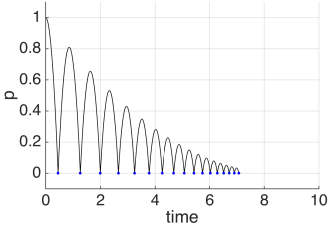
The bouncing ball has a Zeno execution:

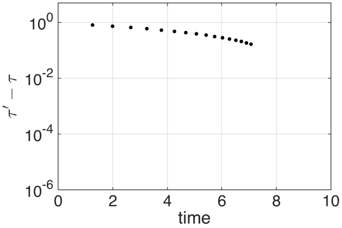
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

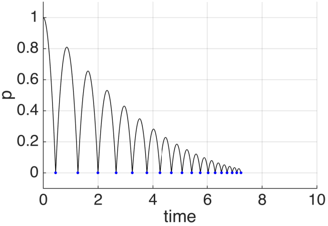
The bouncing ball has a Zeno execution:

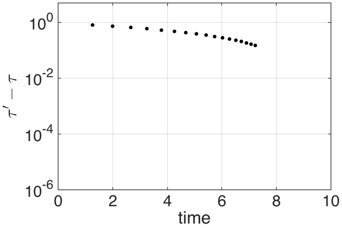
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

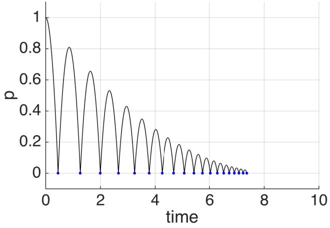
The bouncing ball has a Zeno execution:

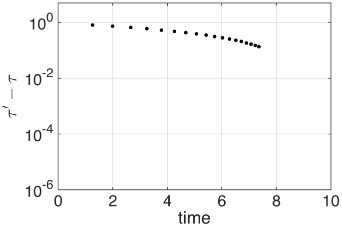
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

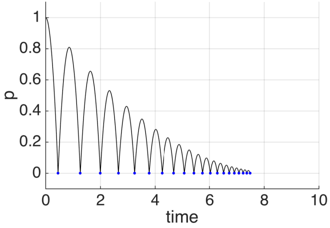
The bouncing ball has a Zeno execution:

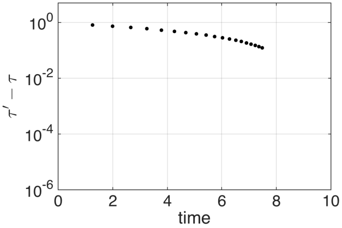
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

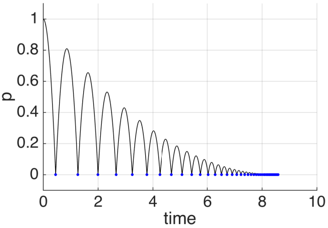
The bouncing ball has a Zeno execution:

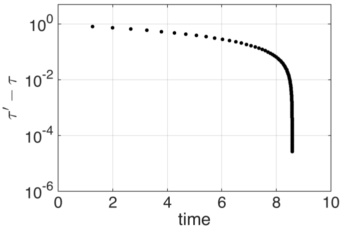
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$





Example of Zeno execution

The bouncing ball has a Zeno execution:

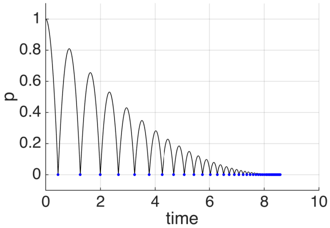
$$\tau_i' - \tau_i = \frac{2|v_i|}{g} \quad \text{where } v_i \text{ is the velocity at bounce } i$$

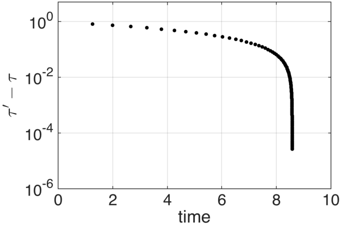
$$v_{i+1} = -\gamma v_i \quad \text{hence} \quad |v_i| = \gamma^i |v_0|$$

It follows that

$$\sum_{i=0}^{\infty} \tau_i' - \tau_i = \frac{2|v_0|}{g} \frac{1}{1-\gamma} < \infty$$

- Rebound velocity & time decay exponentially
- Assymptotically, ball rebounds ∞ fast





Example of a “degenerate” hybrid system

The mad thermostat:

- Temperature dynamics:

$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

  also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$

Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

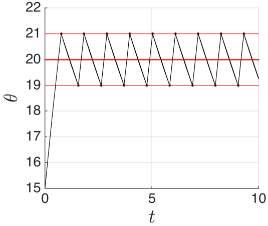
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

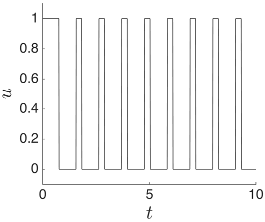
- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$





Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

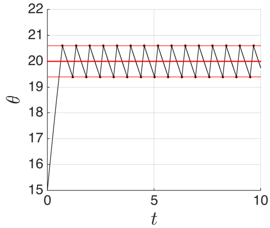
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

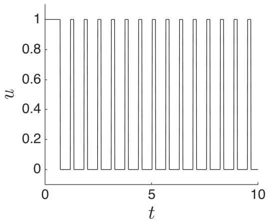
- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \{\theta \,|\, \theta > \theta_{\text{up}}\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \{\theta \,|\, \theta < \theta_{\text{down}}\}$$





Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

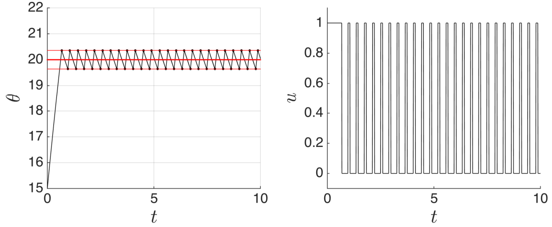
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$



Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

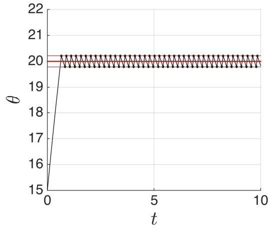
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

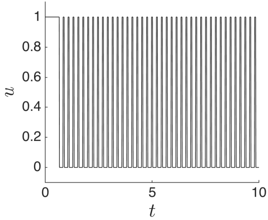
- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

  also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$





Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

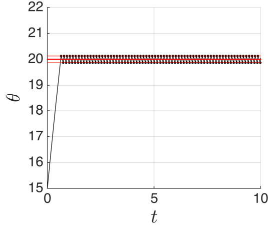
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

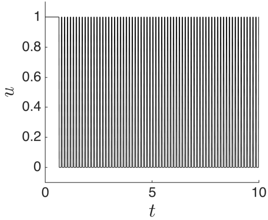
- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

  also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$





Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

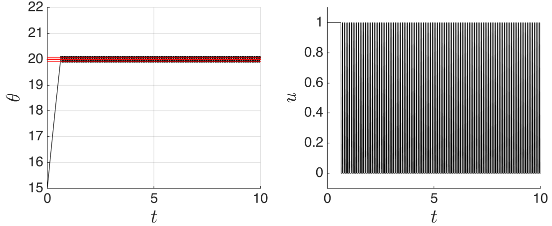
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$



Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

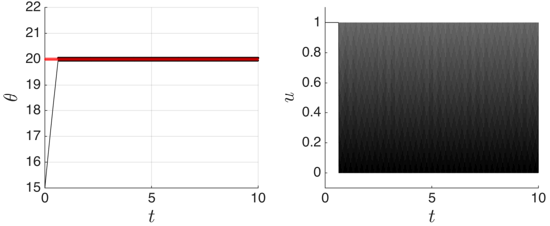
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$



Example of a "degenerate" hybrid system

The mad thermostat:

- Temperature dynamics:

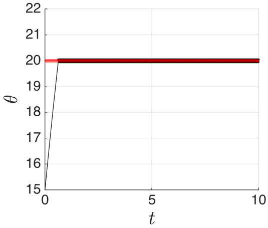
$$\dot{\theta} = \underbrace{\alpha\left(\theta_{\text{out}} - \theta\right)}_{\text{heat flux}} + \underbrace{u}_{\text{heating}}$$

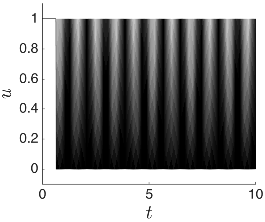
- Thermostat behavior q_{on} and q_{off}:

$$q_{\text{on}} \rightarrow q_{\text{off}} \quad \text{if} \quad \theta \geq \theta_{\text{up}}, \qquad q_{\text{off}} \rightarrow q_{\text{on}} \quad \text{if} \quad \theta \leq \theta_{\text{down}}$$

also expressed as:

$$G\left(q_{\text{on}}, q_{\text{off}}\right) = \left\{\theta \,|\, \theta > \theta_{\text{up}}\right\}, \qquad G\left(q_{\text{off}}, q_{\text{on}}\right) = \left\{\theta \,|\, \theta < \theta_{\text{down}}\right\}$$



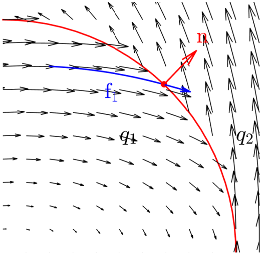


The temperature dynamics become ill-defined for $\theta_{\text{up}} - \theta_{\text{down}} \rightarrow 0$

Outline

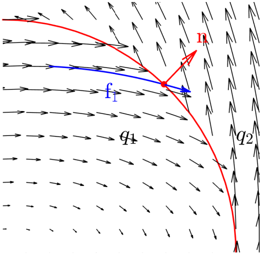
What is an event?

Guard conditions: $G(.,.) : Q \times Q \mapsto X_G \subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered



What is an event?

Guard conditions: $G(.,.) : Q \times Q \mapsto X_G \subseteq X$ defines the set of $\mathbf{x}$ for which a jump from a discrete state q_i to a discrete state q_j is triggered



Formally an event is when the system is in state $(q_i, \mathbf{x}_i)$ meets a guard condition, i.e.

$$\mathbf{x}_i(t) \in G(q_i, q_j)$$

E.g.

- Ball touching the ground
- Thermostat hitting the "switching temperature"
- Engine reaches the RPM for a gear shift

How to simulate dynamics with events?

Can we just "ride through" the events without care?

How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

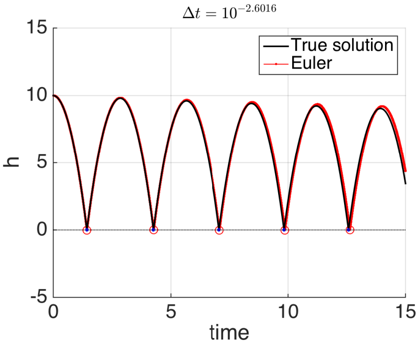
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N-1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

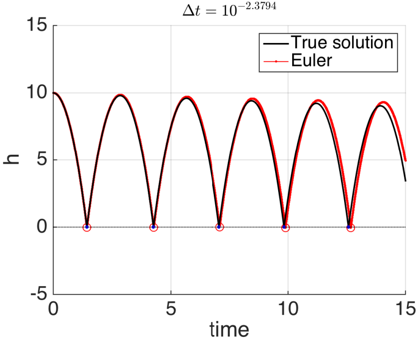
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N-1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

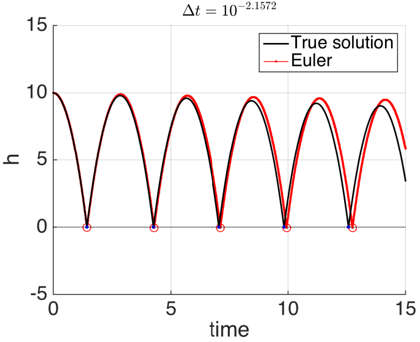
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N-1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

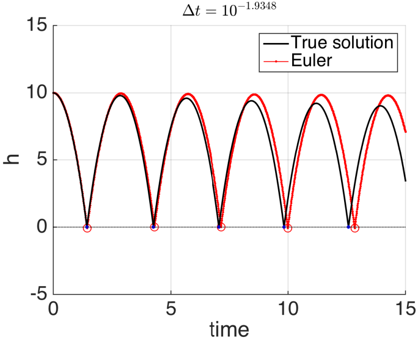
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N - 1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

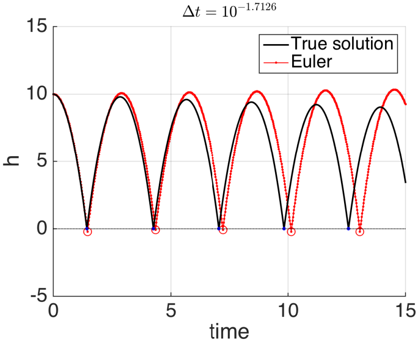
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N-1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

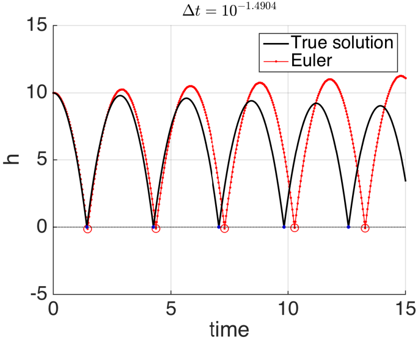
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N-1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$

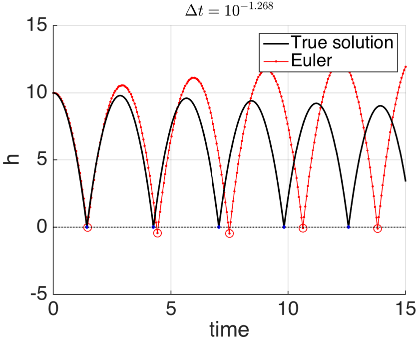
for $k = 0 : N - 1$ **do**

  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

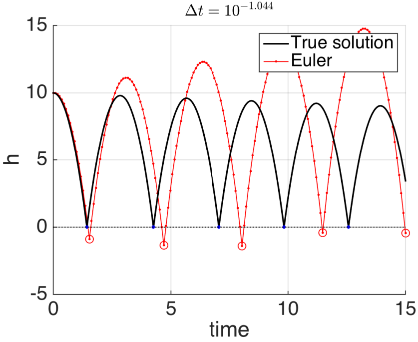
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N - 1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

Can we just "ride through" the events without care?

Explicit Euler on the Bouncing ball, with

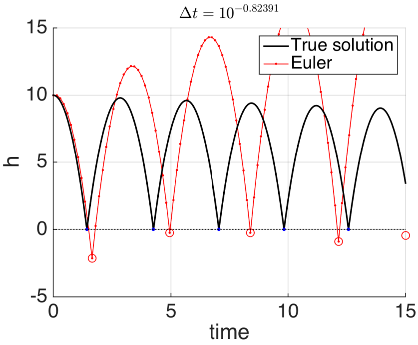
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

Input: Initial conditions $\mathbf{x}_0$
for $k = 0 : N - 1$ **do**
  if $p_k \leq 0$ *and* $v_k < 0$ **then**

$$v_k = -\gamma v_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



How to simulate dynamics with events?

- Simulations ignoring the events can be “risky”. Small step size are typically needed, expensive
- Problem was already observed with friction (non-smooth models), where $v = 0$ was the “event” triggering a change of dynamics (because the friction force changes sign)
- Clean simulations of dynamics with events (hybrid dynamics) require a proper treatment

How to simulate dynamics with events?

Event-based integrators

- Provide dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
- Provide "event condition" (i.e. relevant guard conditions), typically in the form

$$e(\mathbf{x}) = 0$$

How to simulate dynamics with events?

Event-based integrators

- Provide dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
- Provide "event condition" (i.e. relevant guard conditions), typically in the form

$$e(\mathbf{x}) = 0$$

- Integrator uses a regular simulation code, but monitors $e(\mathbf{x})$

How to simulate dynamics with events?

Event-based integrators

- Provide dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
- Provide "event condition" (i.e. relevant guard conditions), typically in the form

$$e(\mathbf{x}) = 0$$

- Integrator uses a regular simulation code, but monitors $e(\mathbf{x})$
- If $e(\mathbf{x})$ changes sign through a step, i.e. $e(\mathbf{x}_k) \cdot e(\mathbf{x}_{k+1}) < 0$ then integrator seeks step length Δt such that

$$e(\mathbf{x}_{k+1}) = 0$$

How to simulate dynamics with events?

Event-based integrators

- Provide dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
- Provide "event condition" (i.e. relevant guard conditions), typically in the form

$$e(\mathbf{x}) = 0$$

- Integrator uses a regular simulation code, but monitors $e(\mathbf{x})$
- If $e(\mathbf{x})$ changes sign through a step, i.e. $e(\mathbf{x}_k) \cdot e(\mathbf{x}_{k+1}) < 0$ then integrator seeks step length Δt such that

$$e(\mathbf{x}_{k+1}) = 0$$

- Typically integrator stops here and let the main code decide what to do next, e.g. apply discrete state change $q_i \rightarrow q_j$, apply Reset conditions (if there are) and resume simulation with new discrete and continuous states

How to simulate dynamics with events?

Event-based integrators

- Provide dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
- Provide "event condition" (i.e. relevant guard conditions), typically in the form

$$e(\mathbf{x}) = 0$$

- Integrator uses a regular simulation code, but monitors $e(\mathbf{x})$
- If $e(\mathbf{x})$ changes sign through a step, i.e. $e(\mathbf{x}_k) \cdot e(\mathbf{x}_{k+1}) < 0$ then integrator seeks step length Δt such that

$$e(\mathbf{x}_{k+1}) = 0$$

- Typically integrator stops here and let the main code decide what to do next, e.g. apply discrete state change $q_i \rightarrow q_j$, apply Reset conditions (if there are) and resume simulation with new discrete and continuous states
- Event-based integrators can take the condition:

$$e(\mathbf{x}) = 0, \qquad \dot{e}(\mathbf{x}) > 0$$

i.e. an event requires "crossing" the zero in a specific direction (up or down)

Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N - 1$ **do**

  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

return $\mathbf{x}_{0,\ldots,N}$

Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

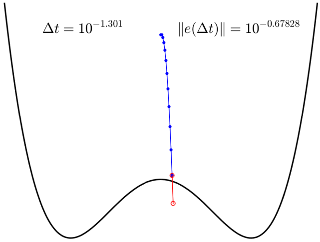
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\ldots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N - 1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

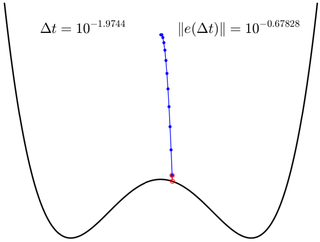
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



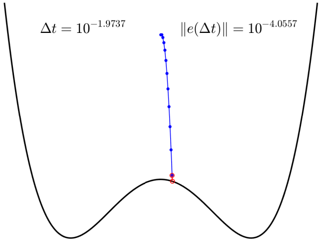
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
    if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
        $\Delta t_{\mathrm{e}} = \Delta t$
        while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\ldots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

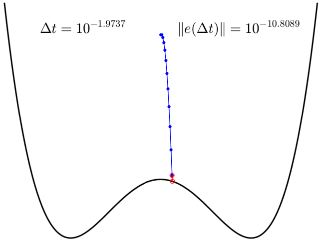
$\quad\quad \Delta t_\mathrm{e} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



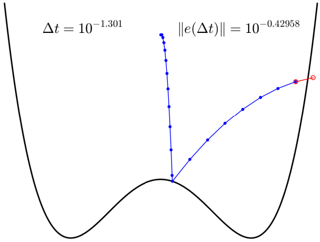
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
    $\Delta t_{\mathrm{e}} = \Delta t$
    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



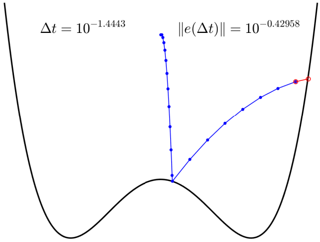
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
    $\Delta t_{\mathrm{e}} = \Delta t$
    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

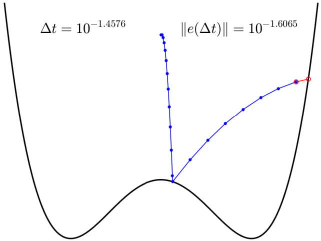
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

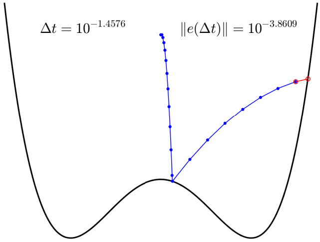
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

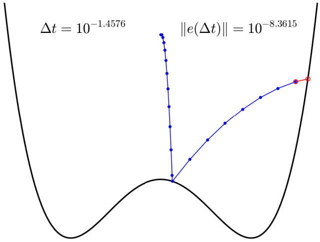
$\quad\quad \Delta t_\mathrm{e} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

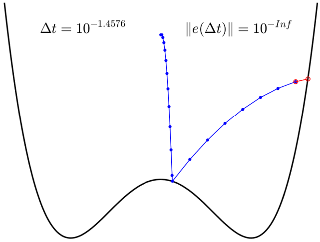
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\ldots,N}$



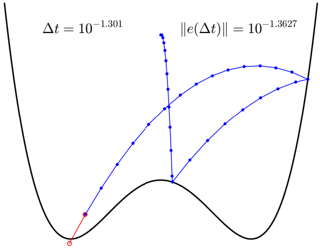
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
    if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
        $\Delta t_\mathrm{e} = \Delta t$
        while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

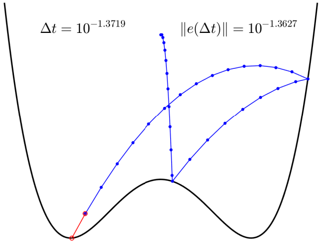
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\ldots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

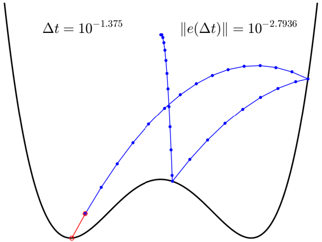
    $\Delta t_{\mathrm{e}} = \Delta t$

    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



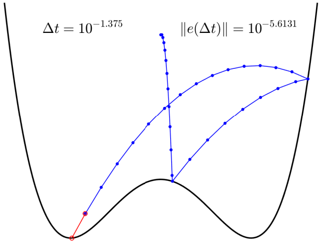
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
    $\Delta t_{\mathrm{e}} = \Delta t$
    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N - 1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

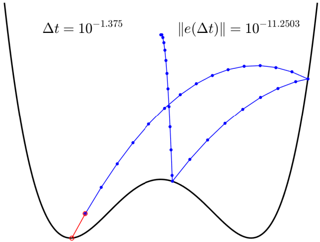
$\qquad \Delta t_\mathrm{e} = \Delta t$

$\qquad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



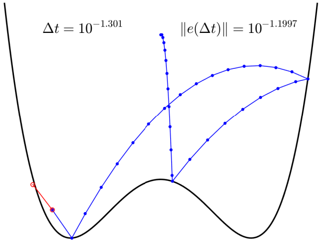
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
    $\Delta t_{\mathrm{e}} = \Delta t$
    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



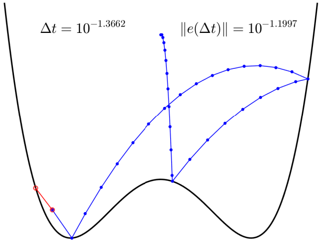
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
    $\Delta t_\mathrm{e} = \Delta t$
    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

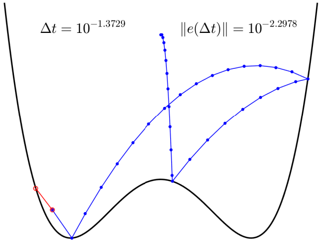
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



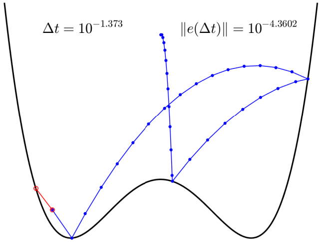
Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$
  if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
    $\Delta t_\mathrm{e} = \Delta t$
    while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Example - Euler with event detection

Input: Initial conditions $\mathbf{x}_0$, Δt

for $k = 0 : N-1$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k)$

$\quad$ **if** $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**

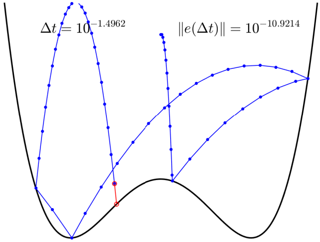
$\quad\quad \Delta t_{\mathrm{e}} = \Delta t$

$\quad\quad$ **while** $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_{\mathrm{e}} \leftarrow \Delta t_{\mathrm{e}} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_{\mathrm{e}} \cdot \mathbf{f}(\mathbf{x}_k)$$

return $\mathbf{x}_{0,\dots,N}$



Note:

- Derivative in the case of Euler

$$\frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_{\mathrm{e}}} = \left.\frac{\partial e}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k+1}} \frac{\partial \mathbf{x}_{k+1}}{\partial \Delta t_{\mathrm{e}}} = \left.\frac{\partial e}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k+1}} \mathbf{f}(\mathbf{x}_k)$$

- If $e(\mathbf{x})$ is linear, then Newton iteration converges in one full step (linear equation to solve)

Event detection with ERK integrators

Input: Initial conditions $\mathbf{x}_0$, Δt
for $k = 0 : N - 1$ **do**

$$\mathbf{K}_i = \mathbf{f}\left(\mathbf{x}_k + \Delta t \cdot \sum_{i=1}^{s} a_{ij}\mathbf{K}_j, \mathbf{u}(t_k + \Delta t c_i)\right)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \sum_{i=1}^{s} b_i \mathbf{K}_i$$

if $e(\mathbf{x}_k)\, e(\mathbf{x}_{k+1}) < 0$ **then**
$\Delta t_\mathrm{e} = \Delta t$
while $|e(\mathbf{x}_{k+1})| > \mathrm{Tol}$ **do**

$$\Delta t_\mathrm{e} \leftarrow \Delta t_\mathrm{e} - \frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}}^{-1} e(\mathbf{x}_{k+1})$$

$$\mathbf{K}_i = \mathbf{f}\left(\mathbf{x}_k + \Delta t_\mathrm{e} \cdot \sum_{i=1}^{s} a_{ij}\mathbf{K}_j, \mathbf{u}(t_k + \Delta t_\mathrm{e} c_i)\right)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_\mathrm{e} \cdot \sum_{i=1}^{s} b_i \mathbf{K}_i$$

return $\mathbf{x}_{0,\dots,N}$

$$\frac{\partial e(\mathbf{x}_{k+1})}{\partial \Delta t_\mathrm{e}} = \left.\frac{\partial e}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k+1}} \frac{\partial \mathbf{x}_{k+1}}{\partial \Delta t_\mathrm{e}}$$

$$\frac{\partial \mathbf{x}_{k+1}}{\partial \Delta t_\mathrm{e}} = \sum_{i=1}^{s} b_i \mathbf{K}_i + \Delta t_\mathrm{e} \cdot \sum_{i=1}^{s} b_i \frac{\partial \mathbf{K}_i}{\partial \Delta t_\mathrm{e}}$$

$$\frac{\partial \mathbf{K}_i}{\partial \Delta t_\mathrm{e}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \sum_{i=1}^{s} a_{ij}\mathbf{K}_j + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial t} c_i$$