AN ITERATIVE METHOD FOR PROVING SAFENESS OF NONLINEAR SYSTEMS

RENATO MANCUSO
ADEL AHMADYAN

Embedded System Verification (ECE-584) Project Presentation
December 2012

OUTLINE

- **Problem statement & system definition**
- **Lapses**
  - Verification
    - System approximation
    - Error bounds
  - Falsification
    - Modeling systems
- **Discussions**
- **Future works**

PROBLEM DEFINITION

Safety:

“Nothing bad is going to happen to the system.”

Formally:

every execution remains within the safe region.

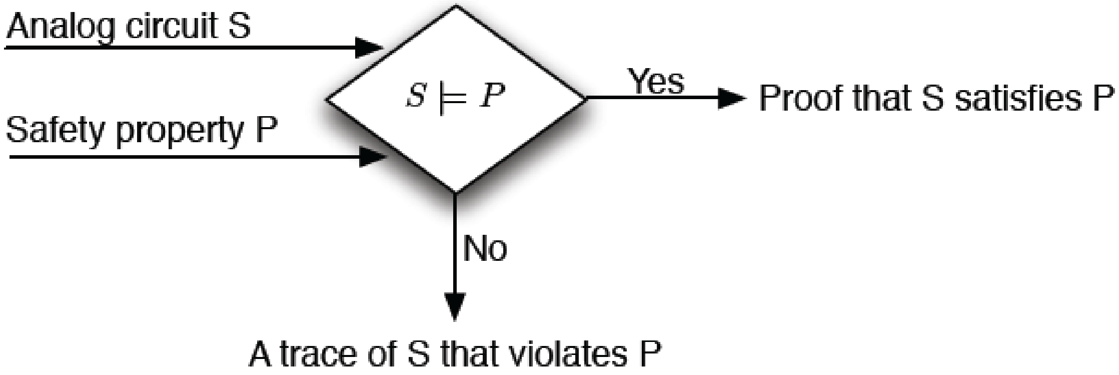
STRATEGIES FOR SAFETY VERIFICATION

Proving Safety :Determining every execution is safe

Falsification: Finding a violating bug

Outputs:

- a counter example: A trajectory from a state in initial set to a state in unsafe set. (Falsification)
- Proof that such counter example does not exists. (Verification)



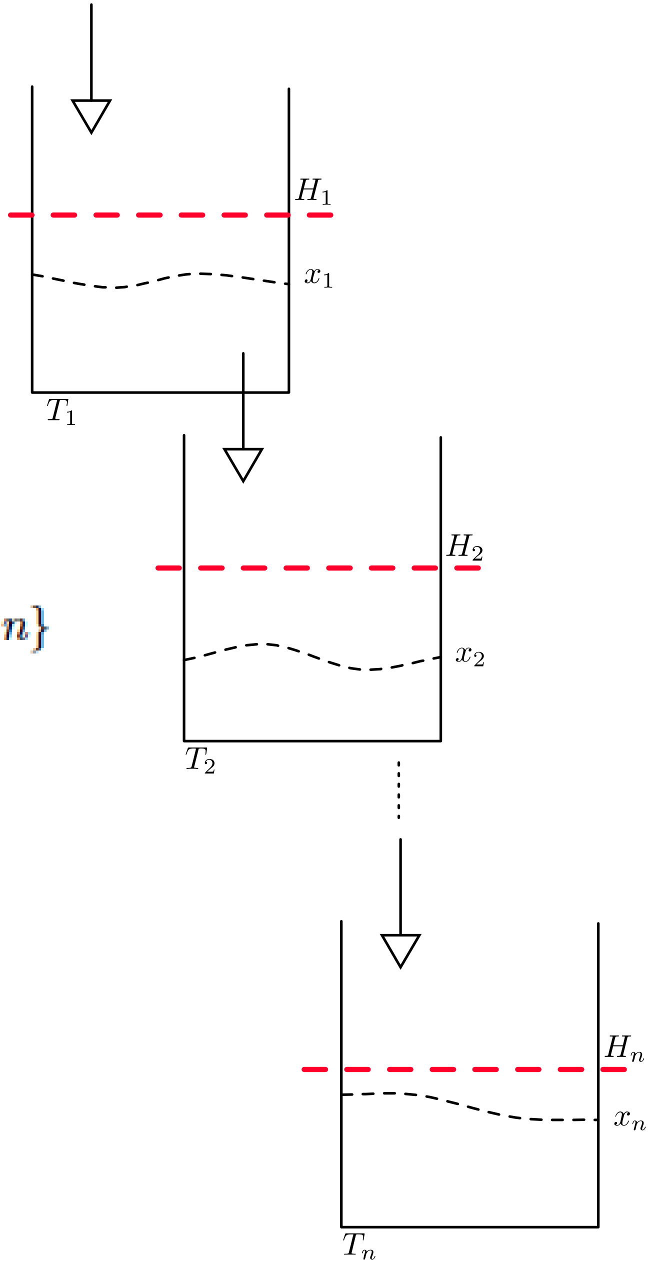
PARAMETERIZED CONTINUOUS SYSTEMS

- **Model for the system** $f(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t)) = 0$
- **State space** $\mathbb{S} \subseteq \mathbb{R}^n$
- **Time is bounded [0, T)**
- ***f* is smooth (*f* belongs to** C^∞**)**
- **There is an upper bound M on all derivatives of F on [0, T).**
  - There should be a limit on **how fast** the system can **evolve**. $1 \leq n \leq \infty : |f^{(n)}| \leq M$
  - This assumptions are not limiting, since we are considering **cyber-physical** systems.

CASE-STUDY

Tank Systems

$$
\begin{aligned}
\dot{x_1} &= -k_1 x_1, \\
\dot{x_2} &= k_1 x_1 - k2x_2, \\
&\dots \\
\dot{x_n} &= -k_{n-1}x_{n-1} - k_n x_n \quad \text{where} \quad k_i = \frac{r}{V_i}, i \in \{1, \dots, n\}
\end{aligned}
$$



Safety properties:

$$\forall i : 1 \leq i \leq N : x_i \leq H_i$$

SOLUTION FOR TANK SYSTEM

( )

Where: and are **constants** and:

So that the **solution** ( ) for **each tank** looks like:

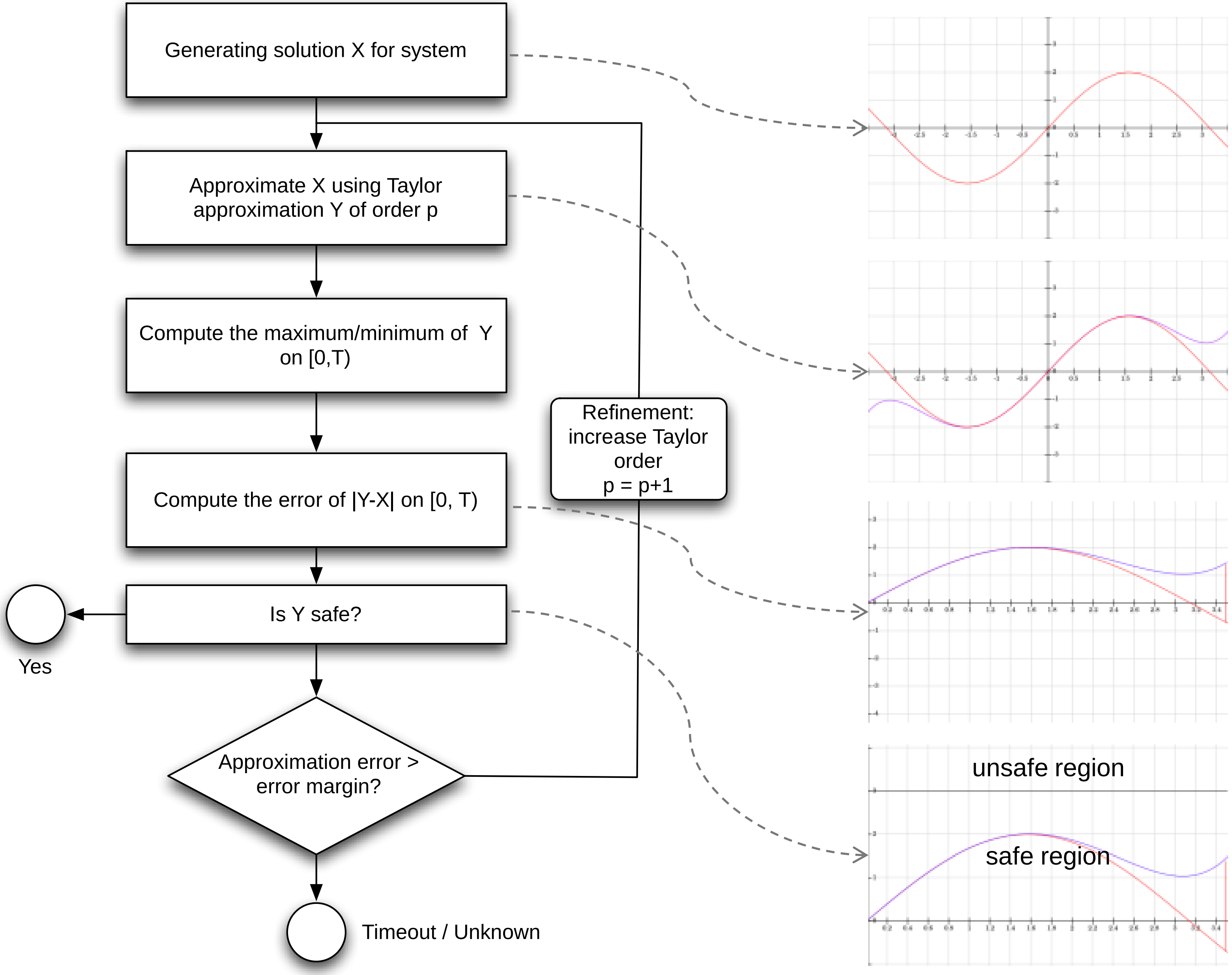
( )

( )

...

PART I:
LAPSES VERIFICATION

LAPSES VERIFICATION FLOW



Generating solution X for system

Approximate X using Taylor approximation Y of order p

Compute the maximum/minimum of Y on [0,T)

Compute the error of |Y-X| on [0, T)

Is Y safe?

Yes

Approximation error > error margin?

Refinement: increase Taylor order p = p+1

Timeout / Unknown

unsafe region

safe region

OBJECTIVES

- Finding **maximum** & **minimum** of the solution over **bounded time** [0, T)

- Generally, there is **no sound technique** for computing max/min of unrestricted **non-linear** functions**.**

- **Sound technique** for maximum/minimum/root finding of **polynomial** equations.

TAYLOR APPROXIMATION

Approximate solution of the system using

single/multivariate Taylor approximation ( )

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k,$$

Where to **expand**? We know that in the expansion point :

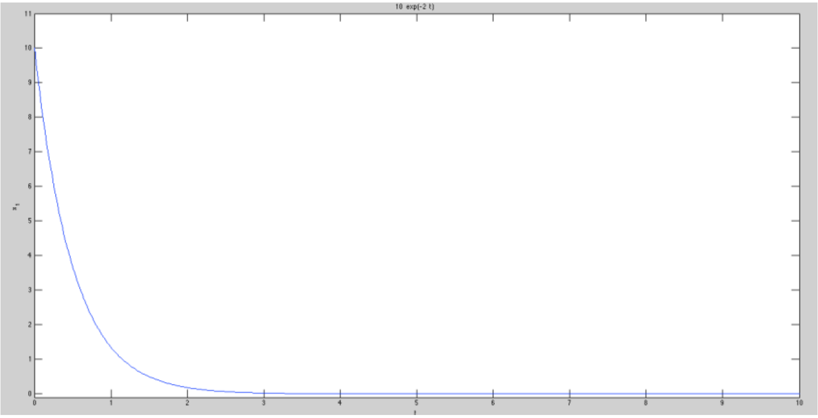
( )

The **error** increases with the **distance** from . Thus, we pick:

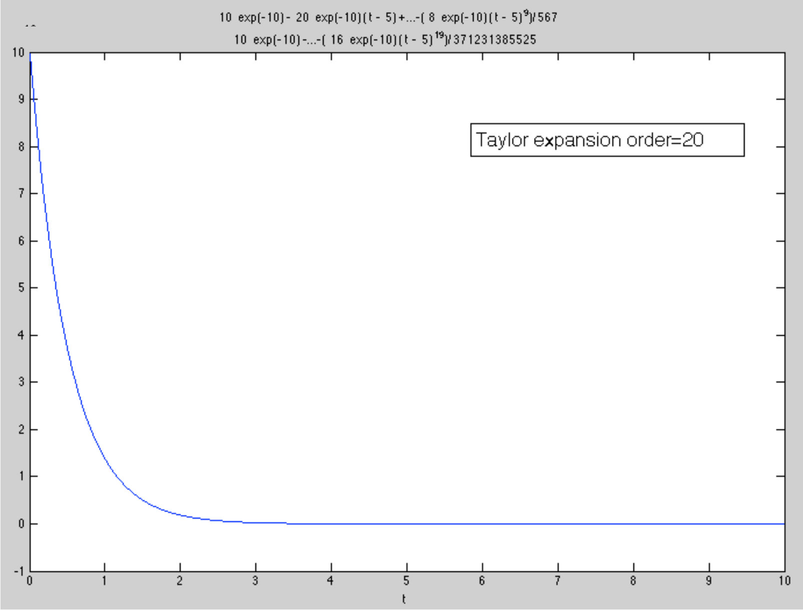
–

Taylor is **approximation**. What is the **bound on error** over [0, T)?

THE SALT LEVEL IN TANK SYSTEM X1

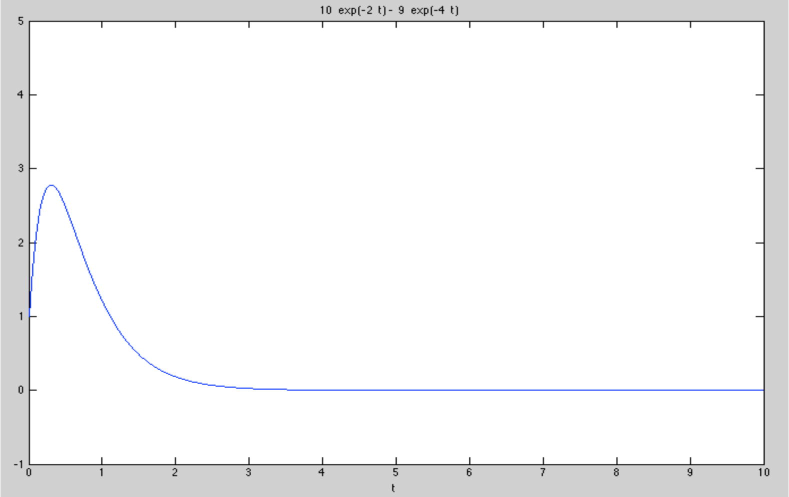


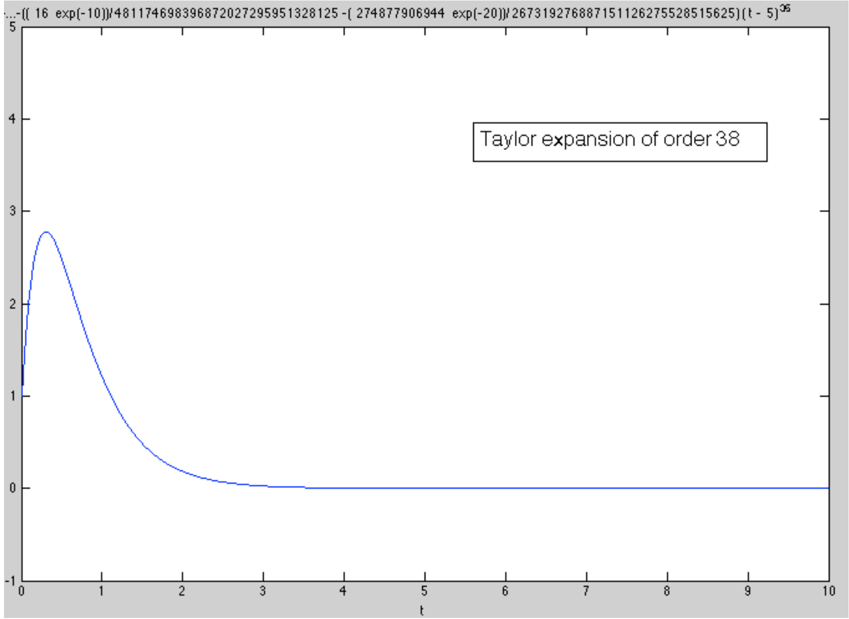
Salt level x1 over [0, 10) in
Original System, K1 = 2



Salt level x1 over [0, 10)
In Taylor expansion

THE SALT LEVEL IN TANK SYSTEM X2





Salt level x1 over [0, 10) in Original System, K2 = 4

T=0.2848
Y=2.8080

MULTIVARIATE TAYLOR APPROXIMATION

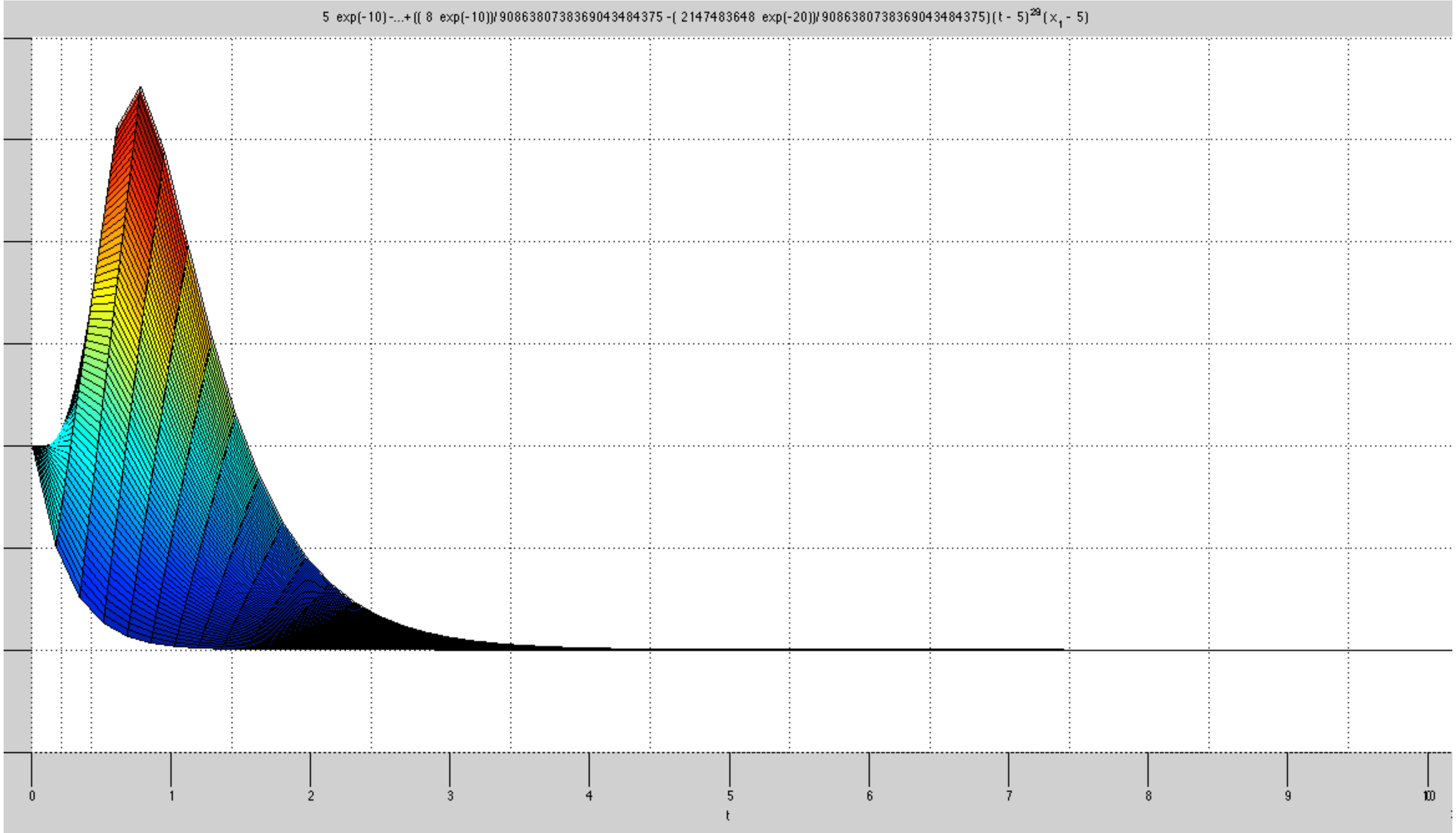
Similarly for evaluating the changes **over more than one** (time) dimension, we use **multivariate** Taylor approximation.

$$f(\boldsymbol{x}) = \sum_{|\alpha| \le k} \frac{D^{\alpha} f(\boldsymbol{a})}{\alpha!} (\boldsymbol{x} - \boldsymbol{a})^{\alpha} + \sum_{|\alpha| = k} h_{\alpha}(\boldsymbol{x}) (\boldsymbol{x} - \boldsymbol{a})^{\alpha},$$

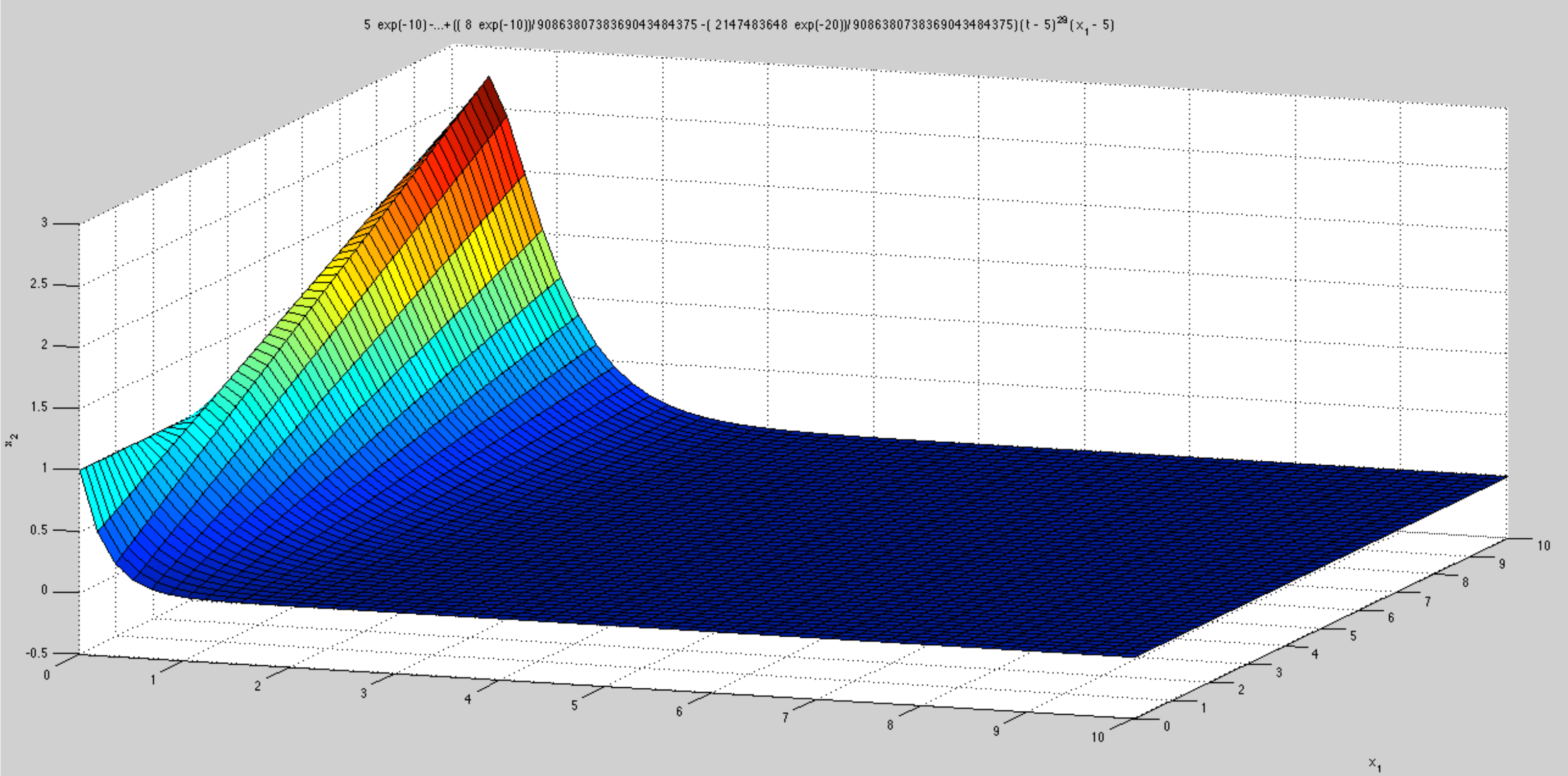
$$\text{and} \quad \lim_{\boldsymbol{x} \to \boldsymbol{a}} h_{\alpha}(\boldsymbol{x}) = 0.$$

Thus, the proposed method can be used to **enforce safety properties** involving a **linear combination of constraints**

MULTIVARIATE ANALYSIS OF



MULTIVARIATE ANALYSIS OF – (CONT'D)



TAYLOR REMAINDER

Lagrange form:

$$0 \leq x \leq T$$

$$R_k(x) = \frac{f^{(k+1)}(\xi_L)}{(k+1)!}(x-a)^{k+1}$$

$$a = \text{expansion point} = \frac{T}{2}$$

$$x \leq \xi \leq a$$

This is a function with a **maximum** in [0, T). Let:

$$1 \leq n \leq \infty : |f^{(n)}| \leq M$$

Under the assumption that the system does not evolve too quickly. Then

$$R_k(x) \leq \frac{M}{(k+1)!}(x-a)^{k+1}$$

N.B. ( ) grows **very quickly**.

Even for large we **converge** very fast.

VERIFYING THE APPROXIMATION

Lemma:

$$\text{If for all } 0 \le t \le T, 0 < \epsilon, Y(t) + \epsilon \le H \text{ and } |Y(t) - X(t)| \le \epsilon \text{ then } X(t) \le H$$

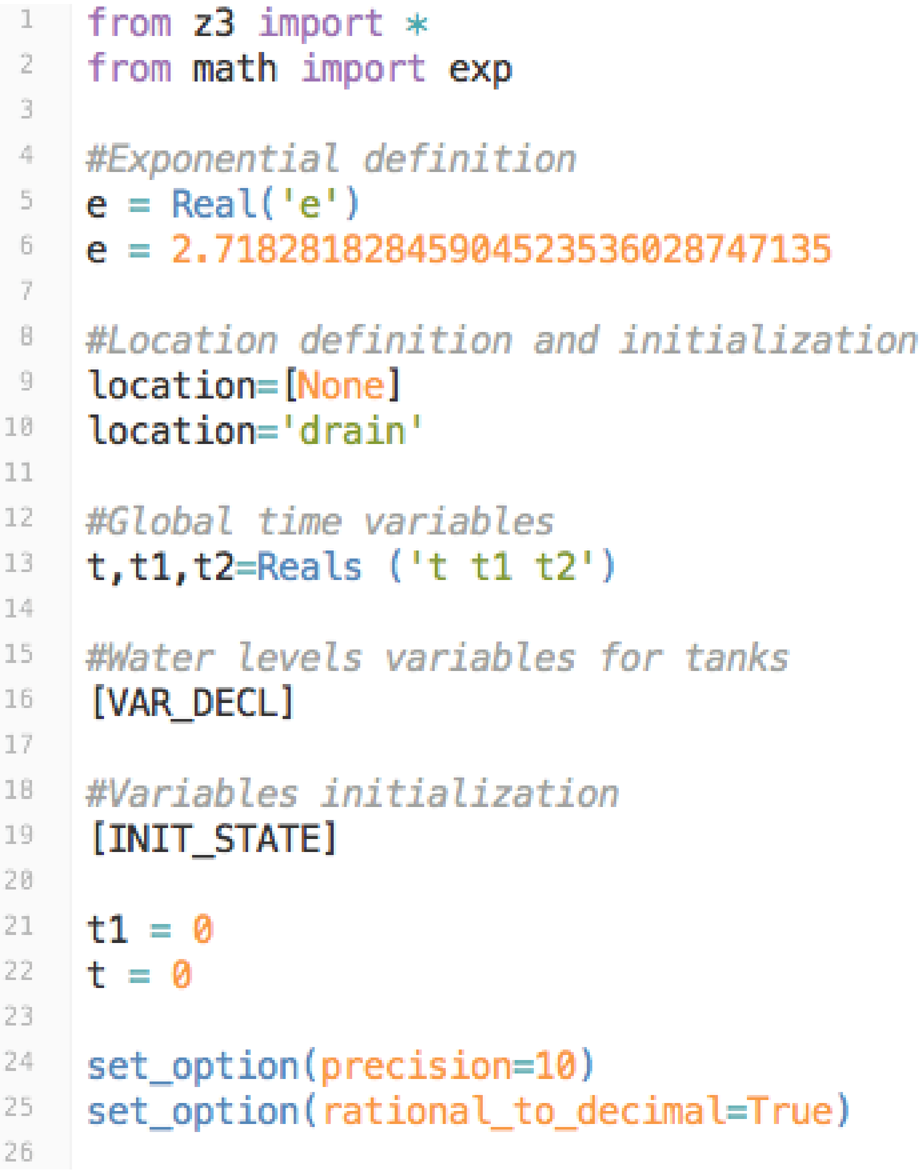
1. We **compute the maximum** of Y (Taylor approximation).
2. If ( ) **is safe**, then **is safe**.
3. Otherwise either:
   1. Our approximation is **too crude**.
   2. The system is **unsafe**.
4. **Refinement:** increase Taylor expansion **order**

PART II:
LAPSES FALSIFICATION

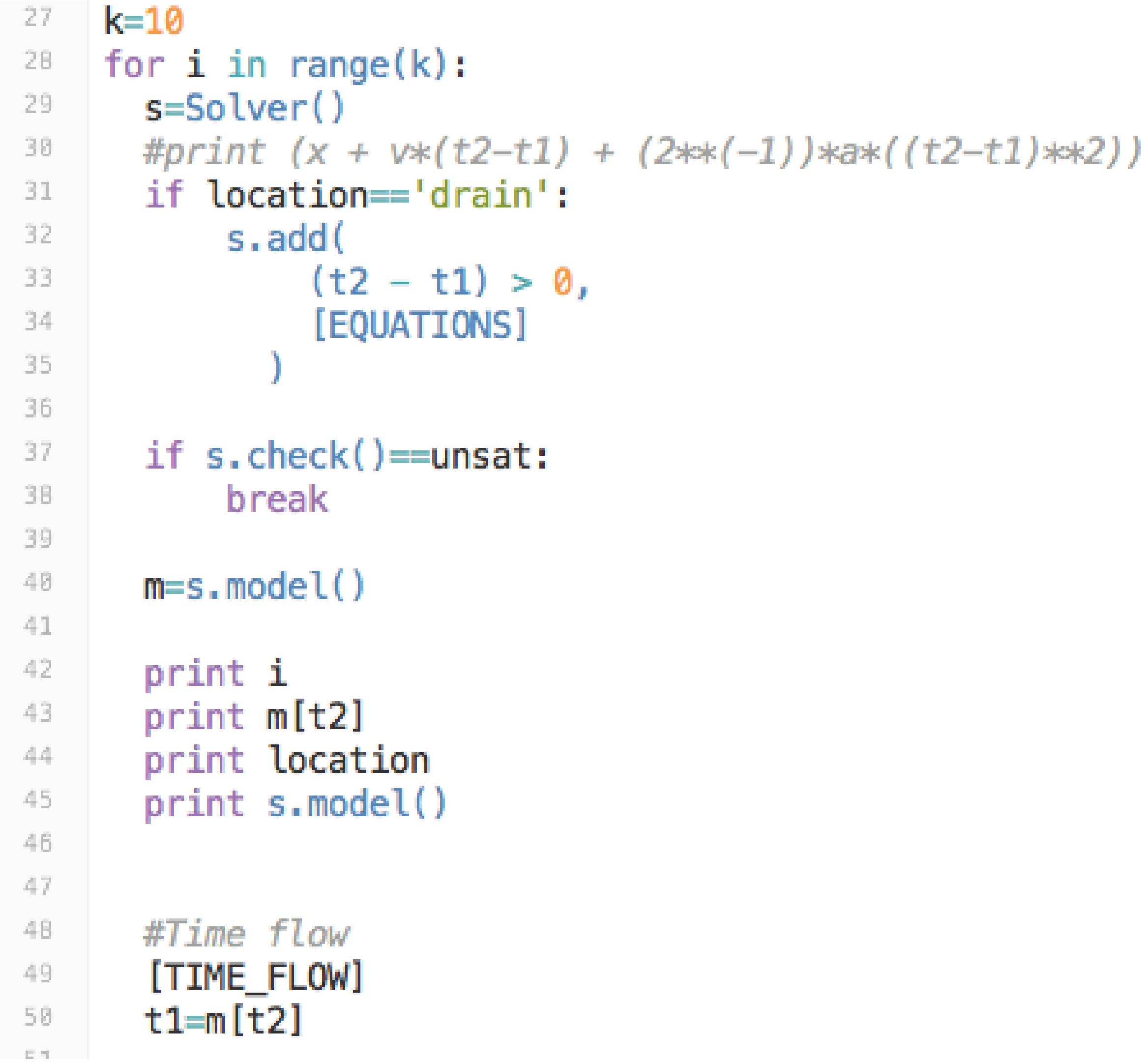
LAPSES:
FINDING VIOLATING TRACE

System model generation in Z3

- **Template for modeling the system in Python**
- **System equation generation using computer algebraic systems**
- **Given the result to Z3 to simulate**
- **Z3 outputs:**
  - SAT,
  - Unknown

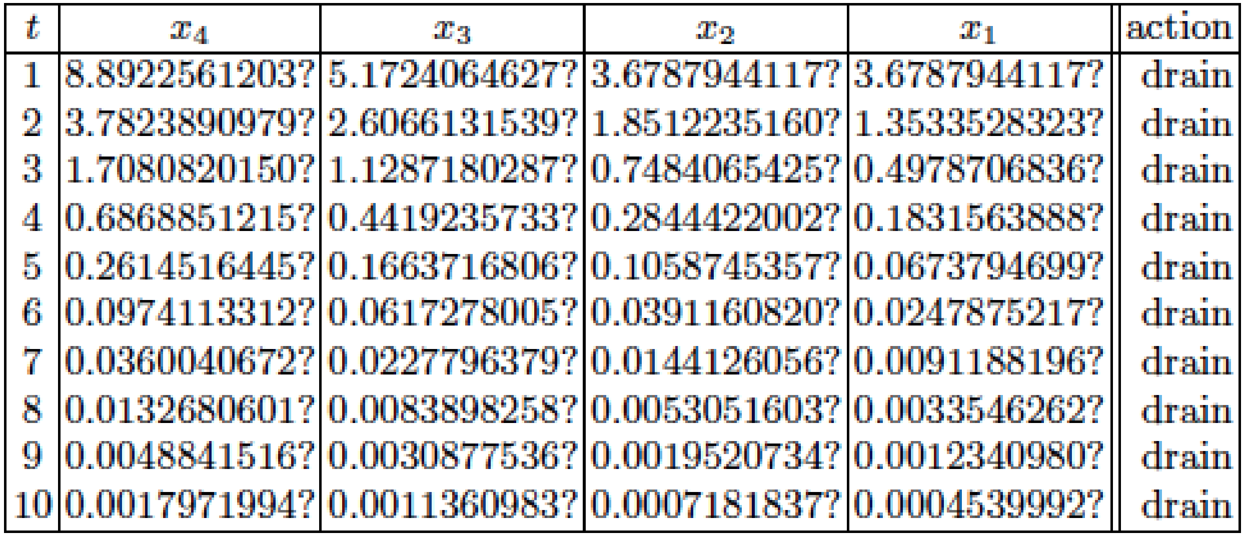


```
from z3 import *
from math import exp

#Exponential definition
e = Real('e')
e = 2.71828182845904523536028747135

#Location definition and initialization
location=[None]
location='drain'

#Global time variables
t,t1,t2=Reals ('t t1 t2')

#Water levels variables for tanks
[VAR_DECL]

#Variables initialization
[INIT_STATE]

t1 = 0
t = 0

set_option(precision=10)
set_option(rational_to_decimal=True)

```



```
k=10
for i in range(k):
  s=Solver()
  #print (x + v*(t2-t1) + (2**(-1))*a*((t2-t1)**2))
  if location=='drain':
      s.add(
          (t2 - t1) > 0,
          [EQUATIONS]
        )

  if s.check()==unsat:
      break

  m=s.model()

  print i
  print m[t2]
  print location
  print s.model()


  #Time flow
  [TIME_FLOW]
  t1=m[t2]
```

EXAMPLE EXECUTION



t	x_4	x_3	x_2	x_1	action
1	8.8922561203?	5.1724064627?	3.6787944117?	3.6787944117?	drain
2	3.7823890979?	2.6066131539?	1.8512235160?	1.3533528323?	drain
3	1.7080820150?	1.1287180287?	0.7484065425?	0.4978706836?	drain
4	0.6868851215?	0.4419235733?	0.2844422002?	0.1831563888?	drain
5	0.2614516445?	0.1663716806?	0.1058745357?	0.0673794699?	drain
6	0.0974113312?	0.0617278005?	0.0391160820?	0.0247875217?	drain
7	0.0360040672?	0.0227796379?	0.0144126056?	0.0091188196?	drain
8	0.0132680601?	0.0083898258?	0.0053051603?	0.0033546262?	drain
9	0.0048841516?	0.0030877536?	0.0019520734?	0.0012340980?	drain
10	0.0017971994?	0.0011360983?	0.0007181837?	0.0004539992?	drain

Table 2. A sample execution of Tank system with $N = 4$ generated by lapses and computed by Z3Py

FUTURE WORKS

Computing the upper bound on derivatives of

$$1 \leq n \leq \infty : |f^{(n)}| \leq M$$

Extend small model theorem (SMT) for continuous dynamics

TOOL REMARKS

- **Requirement:**
  - SMT solver + Nonlinearity support + easy debugging

- **Initially we used iSAT for nonlinear systems.**
  - Claims it supports nonlinearity
    - But it is limited to the most basic systems.
  - Didn’t worked for us

- **Moving on to Z3**

Z3 - LIMITATIONS

- Z3 worked well for simulation. Supported basic nonlinearity
- Didn’t support nonlinear equations with existential quantifiers
- We didn’t want to substitute the nonlinear with linearized system
- So we constructed our own approximation using Taylor.

THANK YOU

PROVING SAFETY

Invariant Generation for Systems

Approximating systems using Taylor approximation

Finding the bound on approximation error

If approximation error is within the error bound

Is safety verified? :)

Else

Refine approximation

VERIFICATION IN Z3

```
t = Real('t')
x = Real('x')

s = Solver()
s.set(auto_config=False, mbqi=False)

s.add( ForAll(t, x<30),
    x<10,
    x>0,
    t>0,
    t<5)

# Display solver state using internal format
print s.sexpr()
print s.check()
```