

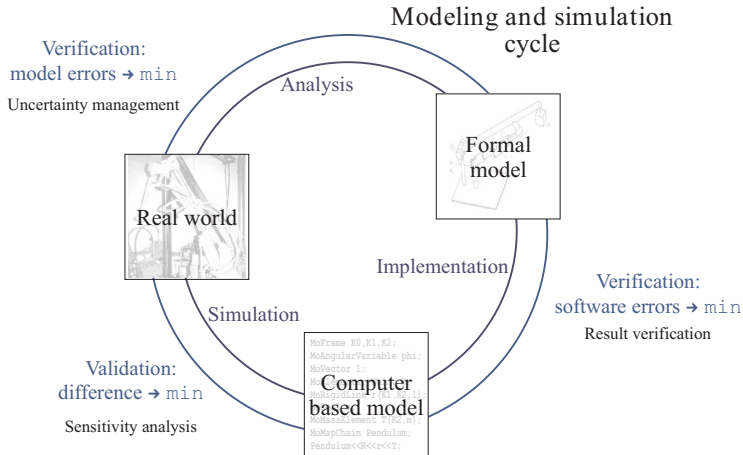
# VERICOMP: Comparing and Recommending Verified IVP Solvers

Ekaterina Auer, Andreas Rauh, and Lorenz Gillner

University of Applied Sciences Wismar

July 27, 2018

# Modeling/Simulation vs Verification/Validation

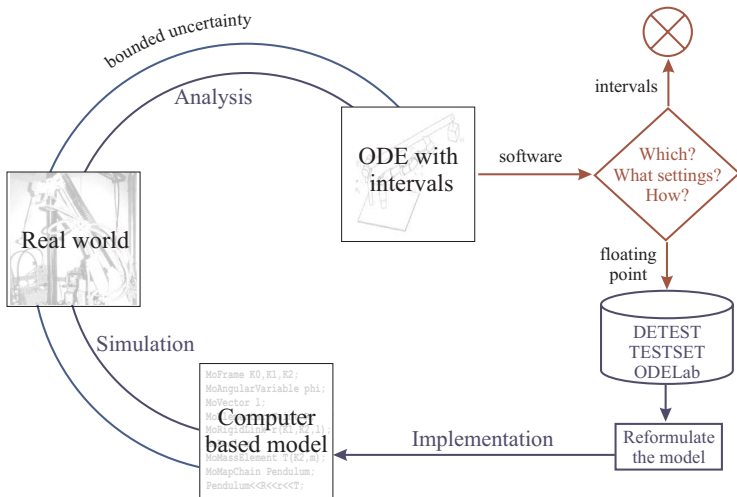


# Result Verification

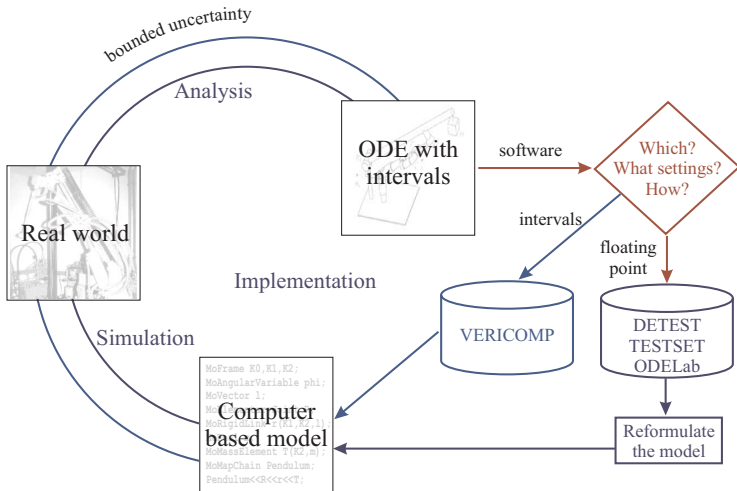
- Features:**
- Prove the correctness of the computed result
  - Take into account rounding or conversion errors
  - Account for the epistemic uncertainty (e.g. in measurements)
- Approaches:**
- interval and affine arithmetics,
  - Taylor models, ...
- Beginnings:**
- Dissertation by R. E. Moore, 1962

Result verification might help where other V&V techniques fail!

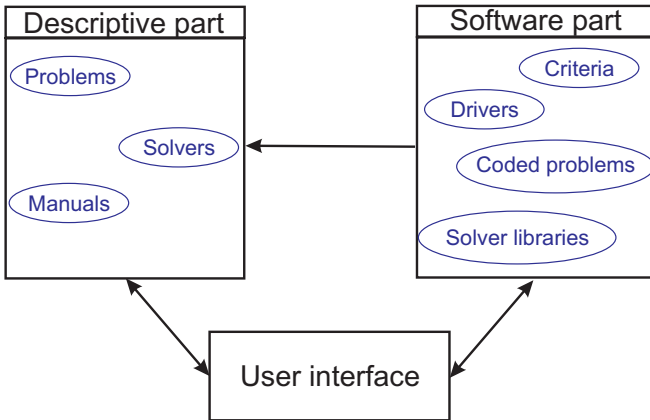
# Comparison of IVP Solvers



# Comparison of IVP Solvers



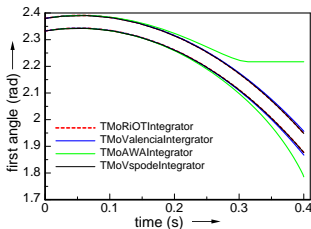
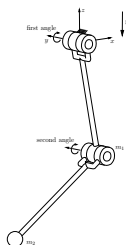
# Comparison of Floating-Point-Based Solvers



DETEST/STIFFDETEST, TESTSET, ODELAB

# Peculiarities of Comparing Verified Solvers

Example: A double pendulum with an uncertain initial angle



- different performance for problems with or without uncertainty
- the answer is an interval with a non-zero diameter
- possible break-down
- the answer is always reliable

## How Can VERICOMP Be of Use?

**Scenario 1:** Find the optimal solver for a problem

→ (or get a solver recommended)

**Scenario 2:** Compare the performance of your newly developed verified IVP solver with the existing ones (VNODE-LP, RiOT, ValEncIA-IVP)

**Scenario 3:** Collaborative analytics like ARCH-COMP

We show an application example for Scenario 1!



# A Possibility for VERICOMP's Application

ARCH Workshop



Applied verification for continuous and hybrid systems

→ Within the National Science Foundation-funded Cyber-Physical Systems Virtual Organization (CPS-VO)

**One aim:** Establishing a curated set of benchmarks submitted by academia and industry in the area

- Proposals for new benchmark problems; tool presentations

**Topics:**

- Tool executions/evaluations based on ARCH benchmarks
- Experience reports including open issues for industry

**Part of activities:** A competition [cps-vo.org/group/ARCH/FriendlyCompetition](https://cps-vo.org/group/ARCH/FriendlyCompetition)

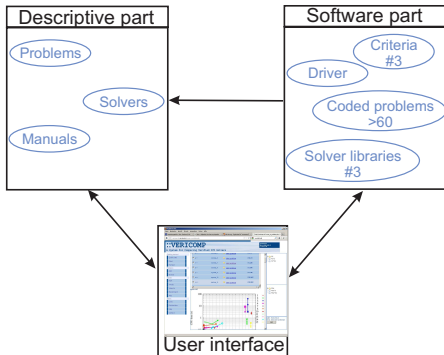
**Workflow:** Join a group → determine the set of problems (ARCH pdf repository) → submit results (via e-mail) → prepare a report ↪ **Manual!**



**VERICOMP would automate this workflow!**

# Framework VERICOMP: Structure and History

University of Duisburg-Essen



↓  
Online since 2010

↓  
Mainly volunteered work  
`vericomp.inf.uni-due.de`

↓  
**Hardware damage 2015**

↓  
Since 2017: reconstruction  
`vericomp.fiw.hs-wismar.de`

↓  
New design online (for problems) 😊

Used by developers of verified IVPS (e.g. DynIBEX 2016)

Dzetkusic, T. (2015). Rigorous integration of non-linear ordinary differential equations in Chebyshev basis. Numer. Alg. 69.1

djt Sandretto, J.; A. Chapoutot (2016). Validated Explicit and Implicit Runge-Kutta Methods, In: Reliable Computing 22,

E. Auer, A. Rauh, L. Gillner

University of Applied Sciences Wismar

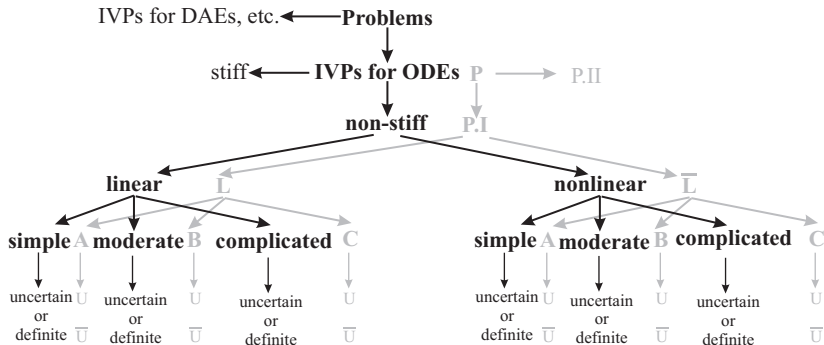
# Theoretical Basis: Problems

(Nonstiff) Initial value problems of the form:

$$\dot{x}(t) = f(x(t)), \quad x(t_0) \in [x^0] \quad ,$$

- $t_0 = 0, t \in [0; t_f] \subset \mathbb{R}$  for some  $t_f > 0$
- $[x^0] = [\underline{x}^0; \bar{x}^0]$
- $f$  can depend on parameters  $p$  with  $[p] = [\underline{p}; \bar{p}]$
- the problem is discretized
- the solution is  $[x_k]$  with  $x(t_k; 0, [x_0]) \subseteq [x_k]$

# Problems: Classification



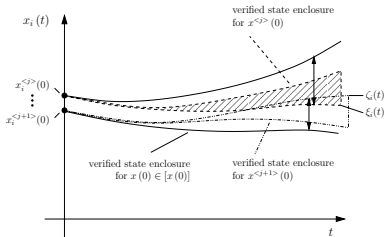
## Theoretical Basis: Criteria

- C1 Number of arithmetic operations at a time step
- C2 Number of function/ Jacobian, etc./ inverse matrix evaluations
- C3 Overhead
- C4 Wall clock time
- C5 User CPU time wrt. overestimation
- C6 Time to break-down  $t_{bd}$  for each solver
- C7 Total number of steps and number of accepted steps.

Each criteria can be weighted according to the application.

## Characterization of Overestimation for C5

1. Analytical solution  $x(t)$ : 
$$\max_{i=1}^n \{d([x_k]_i) - d(x_i(t_k))\}$$
2. No uncertainty: 
$$\max_{i=1}^n d([x_k]_i)$$
3. *Uncertainty in parameters*: 
$$\max_{i=1}^n \{|\overline{x}_i - \xi_i| + |\underline{x}_i - \zeta_i|\}$$



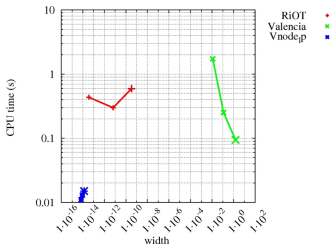
# Statistics

A WPD for ID 19 (formerly)

- Tables
- Work-precision diagrams
- Solution plots

Possible:

- Spider diagrams
- ...



**VERICOMP**  
A System for Comparing Verified IVP Solvers

ID: 19

---

Equations

```
x0' = 2.0*x0-2.0*x0**1
x1' = -x1*x0**1
```

Initial Values

```
1
3
```

Parameters

```
1 - f - 1
```

Class

```
FINLBNJ
```

Exact Solution

```
--f--
```

Description

```
B1 from DETEST
```

[show/hide statistics](#)

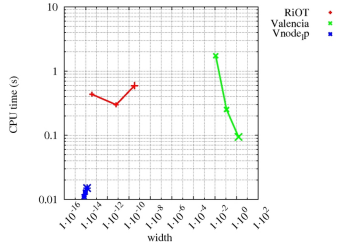
# Statistics

A WPD for ID 19 (formerly)

- Tables
- Work-precision diagrams
- Solution plots

Possible:

- Spider diagrams
- ...



RiOT										
settings	c1	c2	c3	c4	c5_time	c5_width	c6_time	c6_width	c7.1	c7.2
Order=18 Boundary="L0" Repeatability=10-15 Stop ctrl="AUTO" N_max=100 R_max=100 Local error tol=1e-11	✓	✓	✓	000.3163	000.3003	0.0000174000370-12	-1.0000000000000000e+01	0.07420387920039	✓	✓
results from new problems										
<pre>                 30RiOT_order=8_Boundary=""_equality=""_stop_ctrl=""_N_start=""_N_min=""_local_error_tol=""_tol=""                 30RiOT_order=18_Boundary="L0"_Repeatability="10-15"_stop_ctrl="AUTO"_N_max="100"_R_max="100"_local_error_tol="1e-11"_tol=""             </pre>										
Valencia										
settings	c1	c2	c3	c4	c5_time	c5_width	c6_time	c6_width	c7.1	c7.2
Stop=1000.00%	✓	✓	✓	000.7043	000.7043	0.2000012025	2.3100000000000000	1.0001022307	✓	✓
results from new problems										
<pre>                 18Valencia_order=18_Boundary="L0"_Repeatability="10-15"             </pre>										
Vnode_ip										
settings	c1	c2	c3	c4	c5_time	c5_width	c6_time	c6_width	c7.1	c7.2
Order=20 Tolerance=1e-11 HMin=1e-6	✓	✓	✓	000.0054	000.0054	2.0770021004070E-14	10.000000	1.0770310040000E-10	✓	✓
results from new problems										
<pre>                 20Vnode_ip_order=20_Tolerance="1e-11"_HMin="1e-6"_tol=""             </pre>										



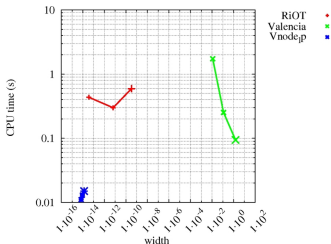
# Statistics

A WPD for ID 19 (formerly)

- Tables
- Work-precision diagrams
- Solution plots

Possible:

- Spider diagrams
- ...



New design

ODE:

**B1 DETEST**

**Equations:**

$$x_0' = 2.0 \cdot x_0 - 2.0 \cdot x_0 \cdot x_1$$

$$x_1' = -x_1 + x_0 \cdot x_1$$

**Variables:**  $x_0, \dots, x_1$

**Description:** The growth of two conflicting populations

**Initial value problems:**

Initial conditions for ID = 12. Class: PINLBNU

$x_0(0) = 1.0$       no parameters available

## Recommender: Formulation in VERICOMP

$$\max\{utility(U, K, G)\} \quad \text{with} \quad K = (P, E, S)$$

	Meaning	Meaning in VERICOMP
$U$	User	Problem
$E$	Entity set	Solvers
$G$	Recom. items from E	Recommended solvers
$K$	Context	$K = (P, S)$ ( $E$ is not dynamic)
$P$	User profile	Problem characteristics $\rightarrow$ classification
$S$	Situation	Applications (e.g. online/offline)
	Utility function	$\sum_{i=1}^7 w_i n(C_i(g)), g \in G, \sum_{i=1}^7 w_i = 1$

**Method:** Multiattribute utility collaborative filtering with C1-C7 ( $C_i(g)$ ) and weighting  $w_i$  according to  $S$

# Illustration: Biological Wastewater Treatment

## Simplified ASM1



$$\dot{S} = \frac{Q_W}{V_A} (S_W - S) - \mu(S, S_O) \frac{1}{Y} X$$

$$\begin{aligned} \dot{X} = & -\frac{Q_W}{V_A} X + \frac{Q_{RS,nom}}{V_A} (X_{Set} - X) + (\mu(S, S_O) - b) X \\ & + \frac{Q_{RS,nom}}{V_A} (X_{Set} - X) \Delta Q_{RS} \end{aligned}$$

$$\dot{S}_O = \frac{Q_W}{V_A} (S_{OW} - S_O) - \mu(S, S_O) \frac{1-Y}{Y} X + \frac{\rho_{O2}}{V_A} \left(1 - \frac{S_O}{S_{O,sat}}\right) u_{O2}$$

$$\dot{X}_{Set} = \frac{Q_W + Q_{RS,nom}}{V_{Set}} X - \frac{Q_{EX} + Q_{RS}}{V_{Set}} X_{Set} + \frac{Q_{RS,nom}}{V_{Set}} X \Delta Q_{RS}$$

Growth rate of bacteria:  $\mu(S, S_O) = \hat{\mu}_H \frac{S}{S+K_S} \frac{S_O}{S_O+K_{OS}}$

Uncertain parameters:

- ① the maximum bacteria growth rate  $\hat{\mu}_H$
- ② inflow concentration  $S_W$  of substrate
- ③ the initial system states

**Task:** Prevent dying of bacteria; ensure efficient purification with small  $S$

# Find the Right Software in VERICOMP!

## Step 0: Call VERICOMP

::VERICOMP News Getting started Contact Feedback Ekaterina Auer ↗

Full Test

**Problems**

Add

Browse

**Solvers**

RIOT

VNODE-LP

Valencia

Recommend

Add

### Welcome to VERICOMP



VERICOMP is a system for comparing and testing *verified solvers* for initial value problems. Verified solvers generate numerical sets that are mathematically proved to contain exact solutions.

### Our motivation

Obtaining *verified solutions* to IVPs for ordinary differential equations is *important in many application areas*, such as biomechanics or automatic control. Test sets and comparison systems for floating-point based solvers turned out to be very useful ([Test set](#), [ODELab](#)). Our hope is that a *similar framework for verified solvers* would promote their use.

### VERICOMP is different

Verified solvers have to be *compared differently* from their floating-point analogs. The main reason is that they perform unequally on problems with and without uncertainty. In either case, the result is an interval with a non-zero width, and it can happen, due to dependency and wrapping, that the considered solver does not reach the predefined integration time (possible break-down). Besides, the reliability of the result does not have to be assessed, because the obtained endpoints are mathematically proved to contain the true solution.



# Find the Right Software in VERICOMP!

## Step 1: Add to the Database

VERICOMP News Getting started Contact Feedback Ekaterina Auer

Full Test

**Problems**

Add

Browse

**Solvers**

RIOT

VNODE-LP

Valencia

Recommend

Add

**Miscellaneous**

### Add a new problem to the database

Fields marked by \* are obligatory.

**Dimension\*:** 1 | 10

Variable names*	Right side*	Initial values*	Parameter names:	Parameter values:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Name of the problem\*:**



**Description:**

**Exact solution(s):**

**Class\*:**  
PI -  L - I  A - :

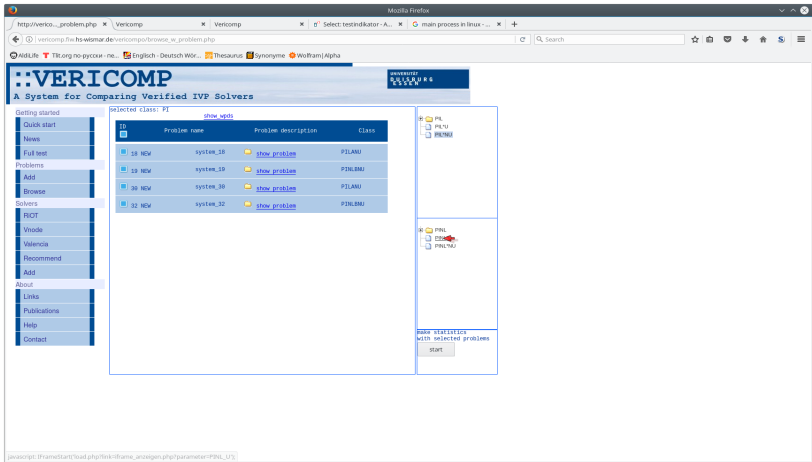
This is a real life problem

Submit



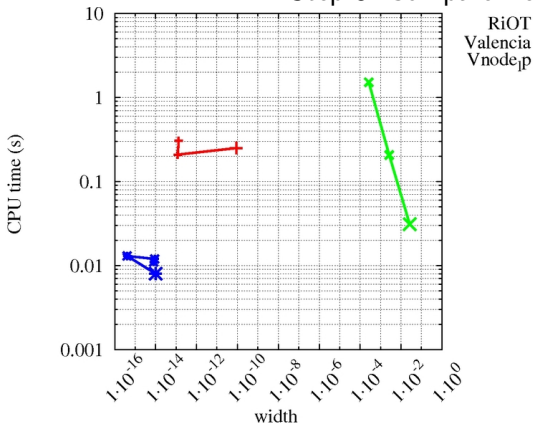
# Find the Right Software in VERICOMP!

## Step 2: Make Tests (Old Design)



# Find the Right Software in VERICOMP!

## Step 3: Compare Results



RiOT  
Valencia  
Vnode,p

Valencia-IVP with  
stepsize  
0.025, 0.0025, 0.0005

VNODE-LP with  
the order 10,  
15, 20

RiOT with the  
order 5, 10, 12

Also possible: Get a solver recommended without time consuming tests!

# Conclusions

## Results:

- The conceptual basis for comparisons of verified IVPS developed
- A problem/solver/statistics database (re)constructed
- The recommender formalism developed

## Future work:

- Full functionality with improved user interface
- Implementation of the recommender
- Possibility to add a new solver semi-automatically