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

- Verification: model errors $\rightarrow \min$
- Uncertainty management
- Validation: difference $\rightarrow \min$
- Sensitivity analysis
- Implementation
- Analysis
- Formal model
- Real world
- Computer based model
- Result verification
- Software errors $\rightarrow \min$

MoFrame K0,K1,K2;
MoAngularVariable phi;
MoVector l;
MoElementaryJoint R;
MoRigidLink r(K1,K2,l);
MoMassElement T(K2,m);
MoMapChain Pendulum;
Pendulum<<R<<r<<T;

E. Auer, A. Rauh, L. Gillner
University of Applied Sciences Wismar
VERICOMP 2.0
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

Real world

Computer based model

bounded uncertainty

Simulation

Analysis

ODE with intervals

Implementation

Which?
What settings?
How?

intervals

software

floating point

DETEST TESTSET ODELab

Reformulate the model
Comparison of IVP Solvers

Motivation

VERICOMP

Applications

Conclusions

Real world

bounded uncertainty

Analysis

Implementation

ODE with intervals

software

Which?

What settings?

How?

floating point

intervals

Computer based model

MoFrame K0,K1,K2;
MoAngularVariable phi;
MoVector l;
MoReal m;
MoMassElement T(K2,m);
MoMapChain Pendulum;
Pendulum<<R<<r<<T;

Which?

What settings?

How?

Floating point

intervals

VERICOMP

DETEST

TESTSET

ODELab

Reformulate the model

E. Auer, A. Rauh, L. Gillner

University of Applied Sciences Wismar

VERICOMP 2.0
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

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
- 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

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)


Theoretical Basis: Problems

(Nonstiff) Initial value problems of the form:

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

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

IVPs for ODEs
- stiff
- non-stiff

IVPs for DAEs, etc.

linear
- simple
- moderate
- complicated

nonlinear
- simple
- moderate
- complicated

Certainty:
- uncertain
- or
- definite

E. Auer, A. Rauh, L. Gillner
University of Applied Sciences Wismar
VERICOMP 2.0
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 \( x(t) \):
\[
\max_{i=1}^{n}\{d(\lfloor x_k \rfloor_i) - d(x_i(t_k))\}
\]

2. No uncertainty:
\[
\max_{i=1}^{n} d(\lfloor x_k \rfloor_i)
\]

3. Uncertainty in parameters:
\[
\max_{i=1}^{n}\{|\bar{x}_i - \xi_i| + |\underline{x}_i - \zeta_i|\}
\]
Statistics

A WPD for ID 19 (formerly)

→ Tables
→ Work-precision diagrams
→ Solution plots

Possible:
→ Spider diagrams
→ ...
Statistics

A WPD for ID 19 (formerly)

→ Tables
→ Work-precision diagrams
→ Solution plots

Possible:
→ Spider diagrams
→ ...

E. Auer, A. Rauh, L. Gillner
University of Applied Sciences Wismar
VERICOMP 2.0
Statistics

A WPD for ID 19 (formerly)

→ Tables
→ Work-precision diagrams
→ Solution plots
Possible:
→ Spider diagrams
→ ...

New design
Recommender: Formulation in VERICOMP

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

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Meaning in VERICOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U) User</td>
<td>Problem</td>
</tr>
<tr>
<td>(E) Entity set</td>
<td>Solvers</td>
</tr>
<tr>
<td>(G) Recom. items from (E)</td>
<td>Recommended solvers</td>
</tr>
<tr>
<td>(K) Context</td>
<td>(K = (P, S)) ((E \text{ is not dynamic}))</td>
</tr>
<tr>
<td>(P) User profile</td>
<td>Problem characteristics (\rightarrow) classification</td>
</tr>
<tr>
<td>(S) Situation</td>
<td>Applications (e.g. online/offline)</td>
</tr>
</tbody>
</table>

Utility function

\[
\sum_{i=1}^{7} w_i n(C_i(g)), \quad g \in G, \quad \sum_{i=1}^{7} w_i = 1
\]

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

E. Auer, A. Rauh, L. Gillner
University of Applied Sciences Wismar
VERICOMP 2.0
Illustration: Biological Wastewater Treatment

Simplified ASM1

\[
\dot{S} = \frac{Q_W}{V_A} (S_W - S) - \mu (S, S_O) \frac{1}{Y} X \\
\dot{X} = - \frac{Q_W}{V_A} X + \frac{Q_{RS,nom}}{V_A} (X_{Set} - X) + (\mu (S, S_O) - b) X \\
\quad + \frac{Q_{RS,nom}}{V_A} (X_{Set} - X) \Delta Q_{RS} \\
\dot{S_O} = \frac{Q_W}{V_A} (S_{OW} - S_O) - \mu (S, S_O) \frac{1 - Y}{Y} X + \rho O_2 \frac{V_A}{V_{Set}} \left(1 - \frac{S_O}{S_{O,sat}}\right) u_{O2} \\
\dot{X}_{Set} = \frac{Q_W + Q_{RS,nom}}{V_{Set}} X - \frac{Q_{EX} + Q_{RS}}{V_{Set}} X_{Set} + \frac{Q_{RS,nom}}{V_{Set}} X \Delta Q_{RS}
\]

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

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 intervals of solvers are mathematically closed to contain the true solution.
Find the Right Software in VERICOMP!

**Step 1: Add to the Database**

<table>
<thead>
<tr>
<th>Full Test</th>
<th>Problems</th>
<th>Add</th>
<th>Browse</th>
<th>Solvers</th>
<th>RIOT</th>
<th>VNODE-LP</th>
<th>Valencia</th>
<th>Recommend</th>
<th>Add</th>
<th>Miscellaneous</th>
</tr>
</thead>
</table>

Add a new problem to the database

Fields marked by * are obligatory.

<table>
<thead>
<tr>
<th>Dimension*:</th>
<th>Variable names*:</th>
<th>Right side*:</th>
<th>Initial values*:</th>
<th>Parameter names*:</th>
<th>Parameter values*:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name of the problem*:</th>
<th>Description:</th>
<th>Exact solution(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class*:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI - L 1 A -</td>
</tr>
</tbody>
</table>

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

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