Result Verification and Uncertainty Management in Engineering Applications

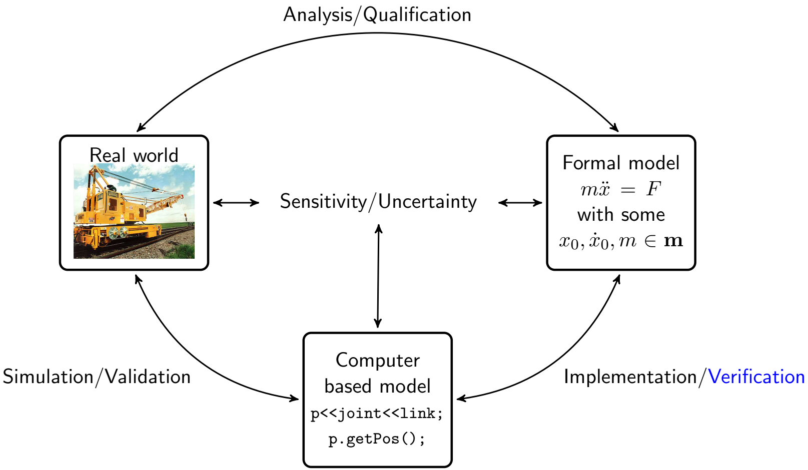
Ekaterina Auer

Joint work with
Wolfram Luther, Andreas Rauh, Stefan Kiel, Andrés Kecskeméthy

University of Applied Sciences Wismar
Universities of Duisburg-Essen and Rostock

September 22, 2014

Modeling and Simulation Cycle vs. V&V Activities



High requirements on safety and reliability → Need for V&V

Traditional V&V: A Short Overview

Beginnings: In the area of computational fluid dynamics

Standards: Generally: no true standard, only *guidelines*
Software V&V: IEEE 1012

Approaches: Formal methods for mission-critical tools
Syntactic methods otherwise

Necessary:

- Flexible and interoperable data types and libraries
- Adaptive self-learning algorithms
- Dealing with uncertainty or bad/missing data
- Uncertainty visualization
- Distributed and parallel computing, crowd sourcing
- User support...

Fact: Result verification can help where others fail!

Where Can Methods with Result Verification Help?

Qualification

- design/definition
- optimization
- dependability

Sensitivity/Uncertainty

- categorization
- quantification
- reliability

Verification

- code verification
- result verification
- software qualification
- accuracy assessment

Validation

- design of experiments
- validation metrics

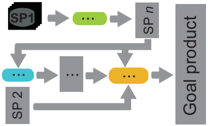
Goals:

- A procedure for V&V without simplifications/benchmarks
- Interoperable use of libraries (verified, probabilistic)
- User guidance for the V&V software choices

Outline

1. V&V guidelines
    - Process-oriented verification guidelines: Goals and tools
    - Illustration
2. Uncertainty and sensitivity analysis
    - Definitions, comparisons, illustrations
    - Applications from (bio)mechanics
3. User support: Comparing verified solvers for IVPs
4. Topics for further research

Process-Oriented Verification Guidelines: Goals

Workflow description	Requirements
	• functionalities of the software • parallelization/ real time • exchange formats, data transfer • documentation

↓

Computational & formal models: Definition, classification, logic

↓

Verification extent and congruence

- interoperability
- verification degree
- verification tools

↓

User support: Recommenders, comparison systems

Process-Oriented Verification Guidelines: Tools

Questionnaires:
- input data
- algorithms
- models
- output (data)

Verified software:

Arithmetics
- interval: e.g. CXSC, filib
- affine: e.g. YALAA
- Taylor models: e.g. RIOT

IVP solvers
- interval: e.g. VNODE-LP
- Taylor models: e.g. COSY VI

MSS
e.g. SMARTMOBILE for (bio)mechanics

Optimization
e.g. GLOBOPT (UNIVERMEC), ...

Verification Degree

From the lowest to the highest verification degree:

- C4 Standard floating (fixed) point arithmetic
- C3 IEEE 754 arithmetic, traditional sensitivity (e.g. Monte-Carlo)
- C2 Subsystems verified
- C1 The whole process verified (IEEE 754/P1788)

Additionally alongside the degree:

- \+ code verification
- − no uncertainty/sensitivity analysis

$Q_{\text{id}}q$ a quality indicator, $q \in [0, 1]$, id$\in \{\underline{n}ominal, \underline{u}ncertain\}$,

Possibilities for Defining a Quality Indicator

What factors contribute to the quality?

- The "width" of the uncertain result
- Maximum time intervals over which simulations are feasible
- Memory use, volume of data, ...

Example: A quality indicator for set-based simulations

- $q = n\left(\frac{\text{achieved integration time}}{\text{enclosure width}}\right)$
- n a normalizing function
- achieved integration time $= 1$ for pure kinematics

Questions:

- What about stochastic or fuzzy data types?
- How would crowd sourcing etc. influence it?

Tools: SmartMOBILE for (Bio)Mechanics

1. Verified kinematics/dynamics + uncertainty management
2. Free choice of the underlying arithmetic: templates + solvers

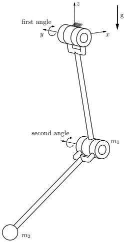
Type	Integrator	Purpose
`MoReal`	`MoAdams,...`	nonverified dynamics

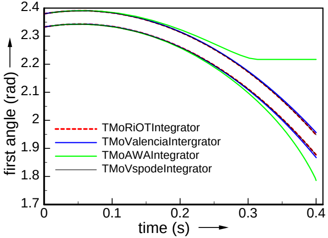
`TMoInterval`
`TMoFInterval`
`TMoTaylorModel`
`TMoTaylorModel`
	`TMoAWA`
	`TMoValencia`
	`TMoRiOT`
	`TMoVSPODE`
		verified dynamics of ODE based systems

`RDAInterval`	`---`	Taylor model based kinematics
`MoFInterval`	`MoIGradient`	verified equilibria kinematics with constraints
`MoSInterval`	`TMoValenciaS`	verified sensitivity
...	...	...

3. Converters MOBILE $\longrightarrow$ SmartMOBILE

Example: Dynamics of a Double Pendulum





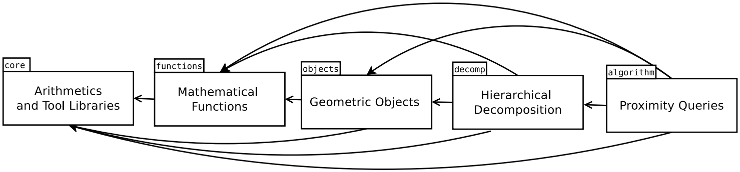

```
# define TMoInterval t;
TMoFrame<t> K0, K1, K2, K3, K4;
TMoAngularVariable<t> psi1, psi2;
// transmission elements
TMoVector<t> l1(0,0,-1), l2(0,0,-1);
TMoElementaryJoint<t> R1(K0,K1,psi1,xAxis);
TMoElementaryJoint<t> R2(K2,K3,psi2,xAxis);
TMoRigidLink<t> rod1(K1,K2,l1),rod2(K3,K4,l2);
t m1(1),m2(1);
TMoMassElement<t> Tip1(K2,m1),Tip2(K4,m2);
// the complete system
TMoMapChain<t> Pend;
Pend << R1<<rod1<<Tip1<<R2<<rod2<<Tip2;
// dynamics
TMoVariableList<t> q; q << psi1<<psi2;
TMoMechanicalSystem<t> S(q,Pend,K0,zAxis);
TMoAWAIntegrator I(S,0.0001,ITS_QR,15);
I.doMotion();
```

Strategy	TMoAWA (variable h)	TMoRiOT ($0.0002 \leq h \leq 0.2$)	TMoValencia ($h = 10^{-4}$)	TMoVSPODE (variable h)
Break–down	0.420	0.801	0.531	0.656
CPU Time*	5	285	22	10

* computed on 8 × Intel Xeon CPU 2.00GHz under Linux 2.6.25.14-69.fc8

UniVerMeC Instead of Templates

Unified Framework for Verified GeoMetric Computations



core *Adapter for underlying arithmetic libraries*

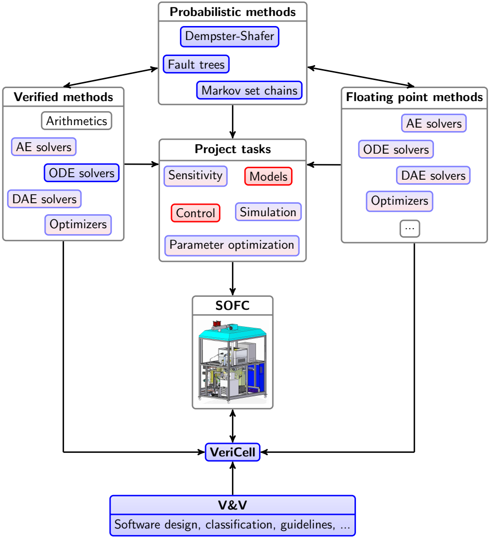
functions *Uniform representation for functions*

objects Implicit surfaces, *IVPs*

decomp Spatial decomposition, *Multisection schemes*

algorithms Distance computation, *Global optimization*, ...

Illustration of PO Guidelines: VERIIPC-SOFC Project



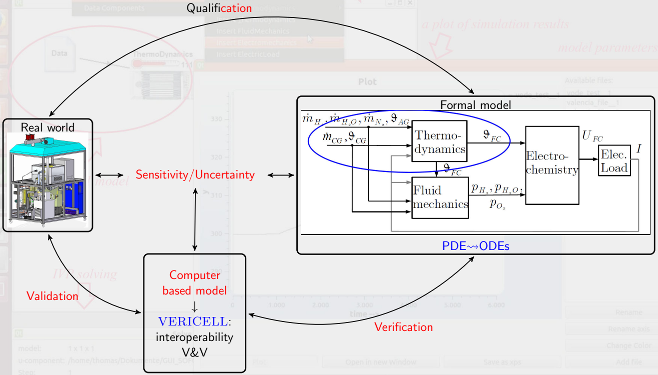
SOFCs convert chemical energy in electricity

+ high efficiency, flexibility wrt. fuel
– high operating temperature

Project goals:

- Models better suitable for control
- Verified methods for robustness
- Modeling/simulation in VERICELL

Illustration (Cont.): V&V Cycle for VERIIPC-SOFC



Interoperability

- Models added "on the fly"
- Verified and non-verified methods combined
- Different verified tools used in parallel

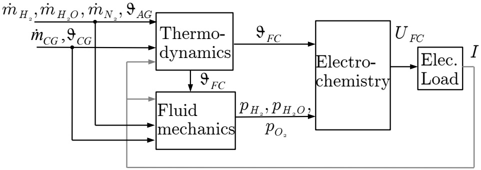
V&V analysis

- Sensitivity/uncertainty via verified methods
- Accurate validation possibilities

Illustration (Cont.): Questionnaire

Description of input data $\rightsquigarrow$ Tolerance of measurements

Initial temperatures ϑ_{CG}, ϑ_{AG} $(n_m = 2)$, mass flows of gases $\dot{m}$



Source: sensors (Eurotherm, Bronkhorst)

Description: ASCII file for T=19963 measurements

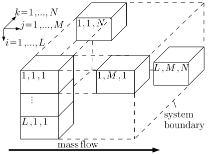
Pre-selection: low-pass filter

Accuracy: four digits for ϑ ± 1K;
$\pm 0.5\%$ of value ± 0.1 of range for $\dot{m}$

Representation: IEEE-754 double precision

Illustration (Cont.): Questionnaire

Description of models ⇝ Verification degree



$$\begin{aligned}
\dot{\vartheta}_{FC} &= \dot{m}_{H_2} \cdot \left(p_{\Delta H,2} \cdot \vartheta_{FC}^2 + p_{\Delta H,1} \cdot \vartheta_{FC} + p_{\Delta H,0}\right) + 6 \cdot p_A \cdot (\vartheta_A - \vartheta_{FC}) \\
&+ (\vartheta_{AG} - \vartheta_{FC}) \cdot (\dot{m}_{H_2} \cdot (p_{H_2,2} \cdot \vartheta_{FC}^2 + p_{H_2,1} \cdot \vartheta_{FC} + p_{H_2,0}) \\
&+ \dot{m}_{H_2O} \cdot (p_{H_2O,2} \cdot \vartheta_{FC}^2 + p_{H_2O,1} \cdot \vartheta_{FC} + p_{H_2O,0}) \\
&+ \dot{m}_{N_2} \cdot (p_{N_2,A,2} \cdot \vartheta_{FC}^2 + p_{N_2,A,1} \cdot \vartheta_{FC} + p_{N_2,A,0})) + I_{FC} \cdot p_{el} - \dot{m}_A \cdot \\
&\cdot (\vartheta_{FC} - \vartheta_{CG}) \cdot (77 \cdot p_{N_2C,0}/100 + 11 \cdot p_{O_2,0}/50 + 77 \cdot p_{N_2C,1} \cdot \vartheta_{FC}/100 \\
&+ 11 \cdot p_{O_2,1} \cdot \vartheta_{FC}/50 + 77 \cdot p_{N_2C,2} \cdot \vartheta_{FC}^2/100 + 11 \cdot p_{O_2,2} \cdot \vartheta_{FC}^2/50) \\
&= f(\vartheta_{FC}, p)
\end{aligned}$$

Formulas: M: ODEs (dim=1,3,9) with different arithmetics (FP/I/...)
P: Parameter identification (1–3)

$$J = \sum_{k=1}^{T} \sum_{j=1}^{n_m} \left(y_j(t_k, p) - y_{j,m}(t_k)\right)^2 \to \text{min}$$

Parameters: heat capacities of gases, enthalpies

Illustration (Cont.): Questionnaire

Description of algorithms $\rightsquigarrow$ Verification degree

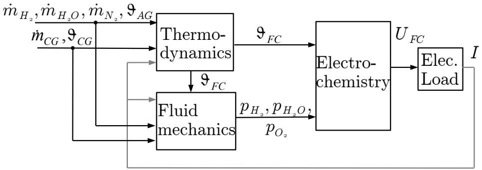
Depends on the kind of arithmetic and the type of identification

For example: M1.P2.A2 $\rightsquigarrow$ GLOBOPT (C2)

Type:	global optimization, numeric, iterative
Parallelization:	J parallelized for multi-cored CPUs and the GPU
Operations:	interval-based for $+, -, \cdot, /$, sqr
Sub-algorithms:	algorithmic differentiation
Sensitivity:	wide search spaces, many parameters

Illustration (Cont.): Questionnaire

Description of output data ⇝ Validation



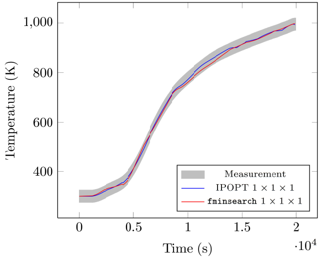
Depends on the kind of arithmetic

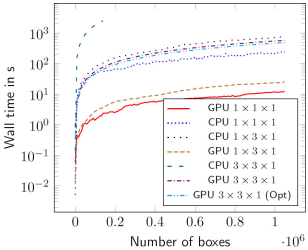
Data type/Accuracy: doubles or intervals

High accuracy: adjustment of measured data and simulations for different operating conditions

Exchange: ASCII file of double precision numbers

VERICELL: Results





- GPU speeds up the possibility *.P2.A2
- Verification degrees C3, C2
- Reliable validation possible
- *.P3.* is too slow at the moment even with FP
- *.P1.* leads to further simplifications in M and is too rough

Uncertainty and Sensitivity Analyses

Uncertainty analysis

Quantify the uncertainty in the the model output from the uncertainty in the input or vice versa

Direct:

- Probabilistic (e.g. Monte-Carlo, polynomial chaos expansion)
- Non-probabilistic (e.g. interval, fuzzy)

Inverse:

- Frequentist
- Bayesian

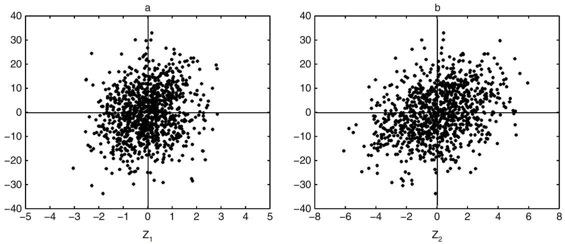
Sensitivity analysis

Apportion the uncertainty in the model output to different sources of uncertainty in the model input

- Scatter plots
- First derivative $\mathbf{s}_i = \dfrac{\partial x(\mathbf{p}_1, \ldots, \mathbf{p}_n)}{\partial p_i}$ for $p_i \in \mathbf{p}_i$

Sensitivity Analysis: Traditional vs. Interval Techniques

Example:* $Y = \Omega(Z_1 + Z_2)$, $Z_i \sim \mathcal{N}(0, \sigma_i)$, Z_1 less uncertain $(\sigma_1 < \sigma_2)$



$$S_i = \frac{\partial Y}{\partial Z_i} = \Omega$$

(the same in the interval case)

$\rightsquigarrow$ Z_1 is less influential

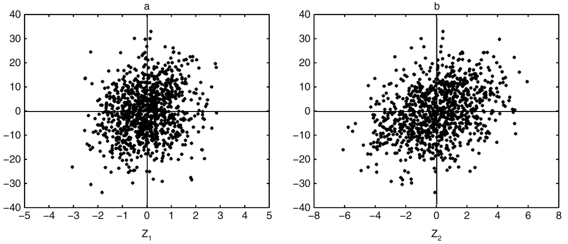
$\rightsquigarrow$ Z_1, Z_2 are equal

- Point derivatives can lead to wrong conclusions $\rightsquigarrow$ $S^\sigma = \dfrac{\sigma_i \partial Y}{\sigma_Y \partial Z_i}$
- Another view: S^σ mixes uncertainty and sensitivity up!

* A. Saltelli et al., *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, 2008

Sensitivity Analysis: Traditional vs. Interval Techniques

Example:* $Y = \Omega(Z_1 + Z_2)$, $Z_i \sim \mathcal{N}(0, \sigma_i)$, Z_1 less uncertain $(\sigma_1 < \sigma_2)$



⇝ scatter plots propagate the uncertainty similarly to interval extensions

$$S_i = \frac{\partial Y}{\partial Z_i} = \Omega$$

(the same in the interval case)

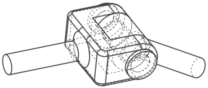
⇝ the model response is indeed equal

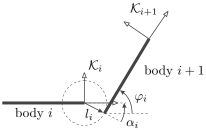
- Sensitivity is *the response of the model to the changes in parameters*
- Uncertainty is quantified by *propagating* it from input to output
- S^σ or similar notions *combine* both in one indicator

* A. Saltelli et al., *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, 2008

Example 1: Uncertainty at the Qualification Stage

`TMoSloppyJoint:`

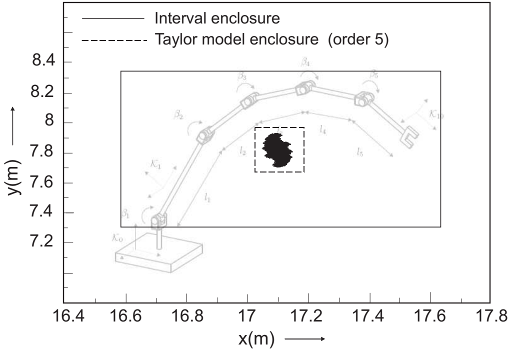




Parameter:

Lengths	$\pm 1\%$
Slackness	± 2mm
Angle	$\pm 0.1°$

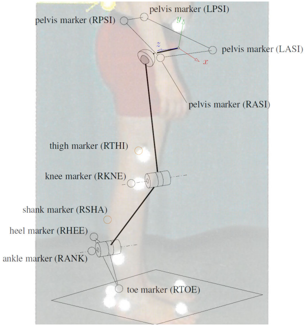
Results:



	x	y	Time (s)
`TMoInterval`	1.047	1.041	0.02
`TMoTaylorModel`	0.163	0.290	0.14

Example 2: Purely Parametric Uncertainty

Body segment motion



Reference $\mathbf{r} = \sum_{i=1}^{n} s_i \cdot \mathbf{p}_i$:

Parameters (mm):

knee width	120 ± 10
ankle width	80 ± 10
displacements	tangential/soft tissue ± 10
	normal ± 5

Femur length (mm):

	TMoRDA	INTERVAL
Knee, ankle	[377.6; 396.7]	$[0; \infty]$
Skin displacement	[0.000; 621.4]	no answer

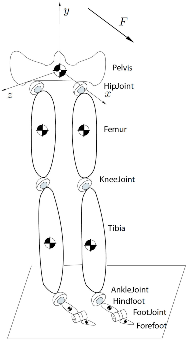
Point sensitivity of femur wrt.

Knee	Ankle	Tangential	Normal	Soft
0.4	-0.3	-2	0.7	1.4

Knee, Ankle: ±7mm; Tangential, Normal, Soft: ±37.5mm

Example 3: Stabilization of Stance

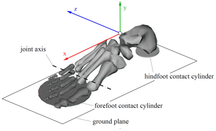
Human skeleton



⇓

Mass parameters

Foot contact



Hunt-Crossley contact

⇓

Contact parameters

Stance stabilizer

PID controller

$$Q = K_p \cdot \varphi + K_d \cdot \omega + K_i \int \varphi dt$$

⇓

Force parameters

Example 3 (Cont.): Uncertain Parameters

Piecewise functions: absolute value $|x|$, sign $sgn(x)$, contact

Force parameters:

ω	$[0.5, 6.28]$s^{-1}
F_x	$[0, 200]$N
F_y	$[0, 50]$N

Contact parameters:

r_{ff}	$[0.04, 0.2]$m	r_{hf}	$[0.02, 0.15]$m
e_N	$[0.01, 0.2]$	μ_{st}	$[0.5, 2.0]$
e_T	$[0.01, 0.2]$	μ_d	$[0.08, 2.3]$

Uncertain mass parameters:

pelvis mass	m_{pelvis}	$[35, 65]$kg
position of the mass	p_x	$[0.05, 0.1]$m
	p_y	$[0.1, 0.5]$m
	p_z	$[-0.05, 0.05]$m

Example 3 (Cont.): Influence on Equations of Motion

$$M(q;t)q'' + \underbrace{b(q,q';t) = Q(q,q';t)}_{f},\ \text{dof=26}$$

Parameters of interest: m_{pelvis}, p_x, F_x, $m_{rfemur} = 10.34$kg

$$[w_1\ w_2\ w_4\ w_6] = [[0,200]\ [-940.00,-595.69]\ [-31.89,31.89]\ [-50.17,45.49]]$$

Sensitivity (interval/nominal)

	m_{pelvis}	p_x	m_{rfemur}	F_x
w_1	0.0	0.0	0.0	1
w_2	-9.8	0.0	-9.8	0.0
w_4	[-0.5,0.5]/0	0.0	0.7848	0.0
w_6	[-9.81,0.5]/-0.25	[-637.66,-343.34]/-490.5	0.5	0.0

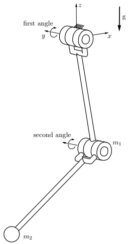
Overview of the (Bio)mechanical Examples

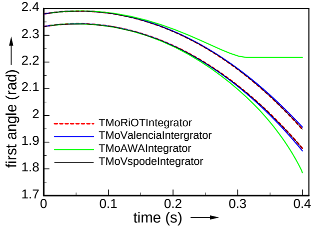
Methods with result verification were used during quantification, verification and validation stages of the V&V cycle as well as for the sensitivity and uncertainty analysis

	V&V	Verification degree	To-Do
Tree type mechanisms			
Double pendulum	✓	C1, $Q_u 0.29$	
Five arm manipulator	✓	C1-, $Q_u 0.92$	
Femur length	✓	C1, $Q_n 0.99$	validation
Stance stabilization	✓	C1, $Q_u 0.27$	dynamics
Muscle activation	✓	C1, $Q_n 0.99$, $Q_u 0.27$	validation
Closed loop mechanisms			
Four bar explicit	✓	C1-, $Q_n 0.99$	
Four bar implicit	✓	C1, $Q_n 1$	dynamics

Comparing Verified IVP Solvers

Dynamic simulations are common in engineering ⇒ IVP



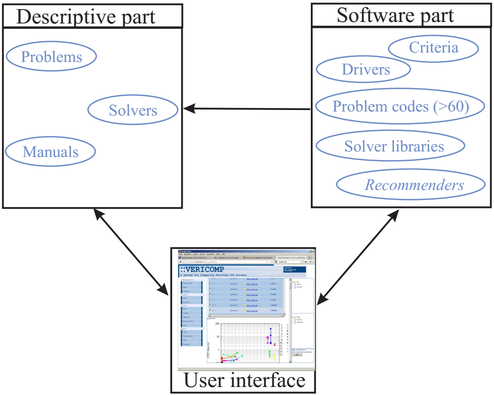


Dynamics of a double pendulum with an uncertain initial angle

Solvers perform differently ⇒ Which is better?

- Different performance for problems with/without uncertainty
- The answer is an interval with a non-zero diameter
- Possible break-down
- The answer is always reliable

Framework VERICOMP



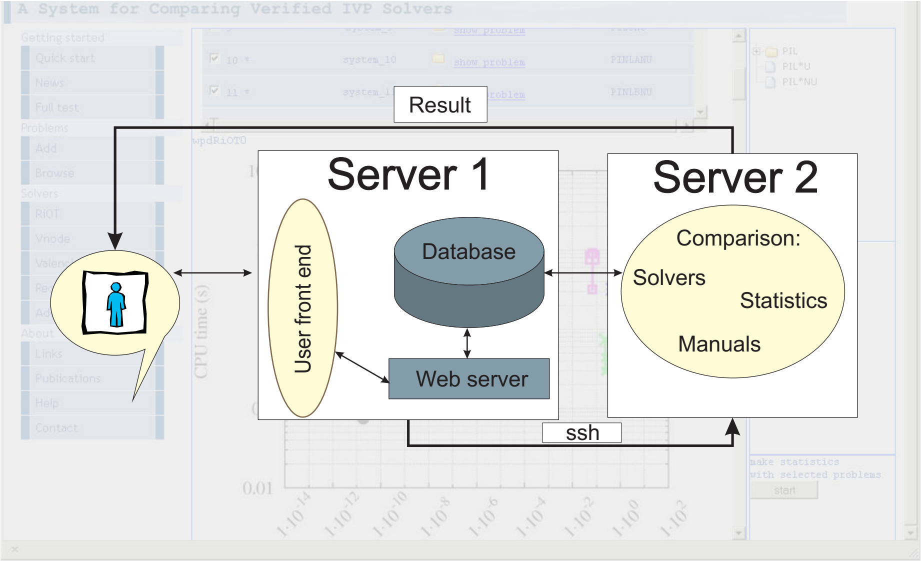
Problems:

- PI Non-stiff
- L/$\overline{\text{L}}$ Linear/Non-linear
- A/B/C Simple/Moderate/Complex
- U/$\overline{\text{U}}$ Uncertain/Nominal (definite)

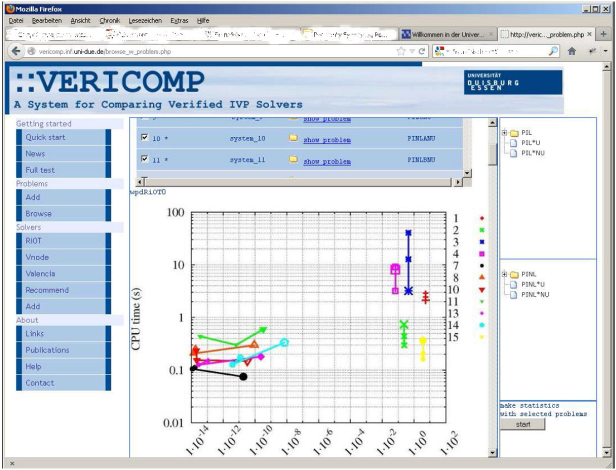
Criteria (weighted):

- C1 # Operations (per time step)
- C2 # Function etc. evaluations
- C3 Overhead
- C4 Wall clock time
- C5 User CPU time wrt. overestimation
- C6 Time to break-down t_{bd}
- C7 # Total steps/accepted steps

VERICOMP: Conceptual Implementation



VERICOMP: Functionality



Features:

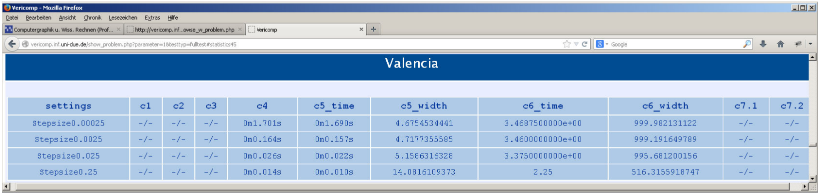
- standard/manual tests
- add a problem to the database
- browse the database
- create/browse WPDs (default/selected)

Statistics:

- tables with C4$= 1/t_\mathrm{c}$, C5$= 1/(e_\mathrm{u} \times t_\mathrm{u})$, C6$= t_\mathrm{bd}/e_\mathrm{bd}$
- work-precision diagrams

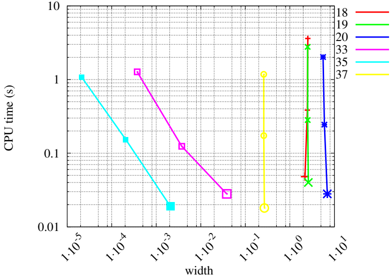
VERICOMP: Statistics

Tables:



settings	c1	c2	c3	c4	c5_time	c5_width	c6_time	c6_width	c7.1	c7.2
stepsize0.00025	-/-	-/-	-/-	0m1.701s	0m1.690s	4.6754534441	3.4687500000e+00	999.982131122	-/-	-/-
stepsize0.0025	-/-	-/-	-/-	0m0.164s	0m0.157s	4.7177355585	3.4600000000e+00	999.191649789	-/-	-/-
stepsize0.025	-/-	-/-	-/-	0m0.026s	0m0.022s	5.1586316328	3.3750000000e+00	995.681200156	-/-	-/-
stepsize0.25	-/-	-/-	-/-	0m0.014s	0m0.010s	14.0816109373	2.25	516.3155918747	-/-	-/-

WPDs:



Visualization of C5:

user CPU time vs e_u

- A difference between the exact and obtained solution widths at t_{out}
- $\overline{\text{U}}$ and the rest of the cases: solution width at t_{out}

VERICOMP: Automatic Problem Descriptions

VERICOMP DATABASE EXAMPLES, EXAMPLE ID(S): 1

GENERATED AUTOMATICALLY, CODE BY E. AUER

Equations:

$$\dot{x}_0 = -p_0 x_0$$

Initial conditions for ID=1:

$$x_0(0) = 1$$

Parameter values for ID=1:

$$p_0 = [-2, -1]$$

The exact solution for ID=1:

$$x_0 = e^{([1,2]x_0)}$$

Desired integration interval: $t_0 = 0$, T =not specified

Description of the problem: Classification for ID=1: PILAU: non-stiff, linear, simple, with uncertain parameters.

Plots

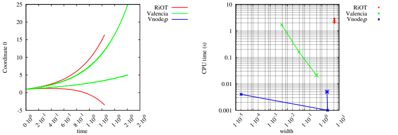


FIGURE 1. Solution plot(s) and WPD for ID=1

1

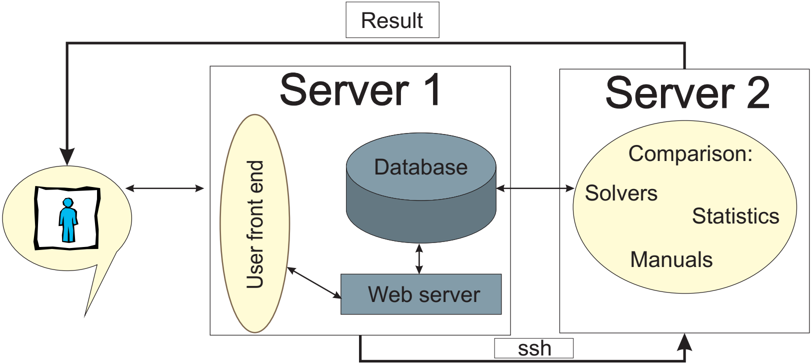
Features:

- LaTeX code for pdfs
- generated from problem descriptions in the database
  - equations
  - parameters (+ IV)
  - exact solution
  - description
- needs data on solution trajectories and WPDs generated by `statistics`

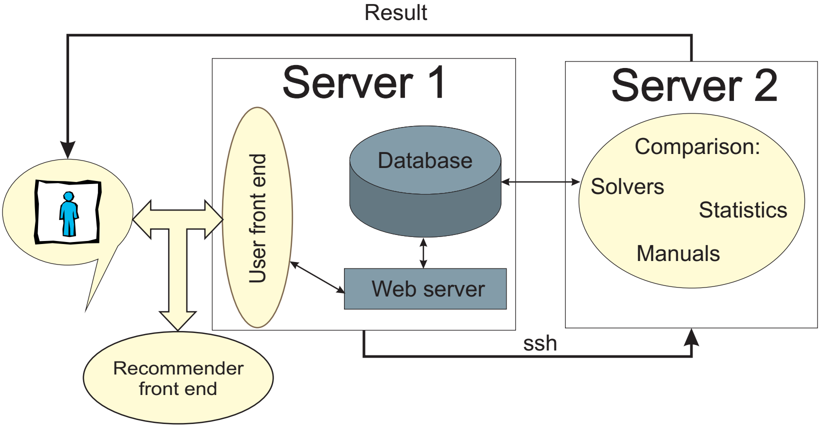
Interesting questions:

- automatic layout
- formatting of long equations

VERICOMP: Recommender



VERICOMP: Recommender



The recommendation can be validated by running the usual tests!

Basics of a Recommender System for IVPS

$\max\{utility(U, C, G)\}$ with $K = (P, E, S)$ and $n = \dfrac{1}{1 + e^{1-(x-40)/10}}$

	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	$\chi(g) = \sum_{i=1}^{7} w_i n(C_i(g))$, $g \in G$, $\sum_{i=1}^{7} w_i = 1$

Method: MAUT with C1-C7, weighting w_i according to S

Similarity: Depends on linearity, complexity, uncertainty

Example 1: VERICOMP Recommender

Task: Recommend a solver for a non-linear, simple problem with uncertainty

Situation: Simulate the problem online over short time intervals
$\rightsquigarrow (w_4\ w_5\ w_6) = (0.4\ 0.4\ 0.2)$

Similarity: $\mu(u)$

PI.$\overline{\text{L}}$.A.U:= $\{3, 33\}$

ID=3:

$$\begin{aligned}\dot{x} &= -x^3/2\\ x(0) &= [0.5, 1.5]\end{aligned}$$

ID=33:

$$\begin{aligned}\dot{x}_0 &= 1\\ \dot{x}_1 &= x_1 \cos(x_0)\\ x_0(0) &= 0\\ x_1(0) &= [0.9, 1.25]\end{aligned}$$

Solvers: RIOT, VALENCIA, VNODE with 3 types of settings each

Example 1: VERICOMP Recommender (Cont.)

Test run data on the problems with IDs 3 and 33

Solver	ID 3					ID 33				
	$t_c(s)$	$t_u(s)$	e_u	t_{bd}	e_{bd}	$t_c(s)$	$t_u(s)$	e_u	t_{bd}	e_{bd}
RIOT 5	3.270	3.197	0.448	10	0.130	3.597	3.466	0.811	10	0.20
RIOT 10	13.030	12.763	0.443	10	0.057	0.860	0.842	0.811	10	0.20
RIOT 15	40.883	40.607	0.443	10	0.055	0.918	0.886	0.811	10	0.20
VALENCIA 1	0.045	0.042	2.987	1.300	5.85	0.260	0.257	0.850	10	309.55
VALENCIA 2	0.287	0.282	2.905	1.17	3.69	1.528	1.521	0.815	10	249.32
VALENCIA 3	2.794	2.780	2.897	1.19	3.77	90.844	90.726	0.812	10	243.87
VNODE 15	0.014	0.009	0.887	6.36	151.77	0.047	0.041	0.811	10	0.203
VNODE 20	0.014	0.007	0.987	3.81	218.18	0.047	0.042	0.811	10	0.203
VNODE 25	0.015	0.009	1.138	2.59	270.42	0.046	0.039	0.811	10	0.203

VNODE 15 should be the best for the chosen situation and problem!

Recommendation:

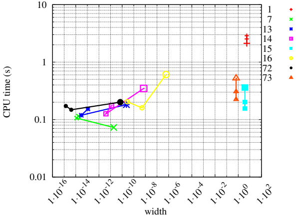
g_i	Cr4 (3/33)	Cr5 (3/33)	Cr6 (3/33)	$\frac{x_3+x_{33}}{2}$
RIOT 5	0.007/0.007	0.007/0.007	0.936/0.500	0.149
RIOT 10	0.007/0.008	0.007/0.008	0.999/0.500	0.155
RIOT 15	0.007/0.007	0.007/0.008	0.999/0.500	0.155
VALENCIA 1	0.058/0.009	0.014/0.010	0.007/0.007	0.020
VALENCIA 2	0.009/0.007	0.008/0.007	0.007/0.007	0.007
VALENCIA 3	0.007/0.007	0.007/0.007	0.007/0.007	0.006
VNODE 15	0.89/0.05	0.99/0.11	0.01/0.48	0.462
VNODE 20	0.89/0.05	0.99/0.11	0.01/0.48	0.461
VNODE 25	0.84/0.05	0.99/0.13	0.01/0.48	0.453

VNODE 15 is actually recommended!

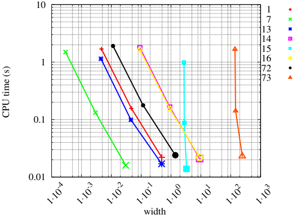
Example 2: Studying the Performance of a Solver

How well does VNODE perform in comparison to RiOT and ValEncIA for the class of simple systems PI.*.A.*?

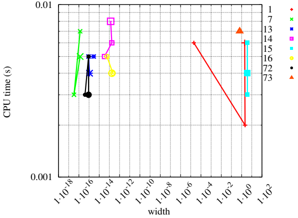
RiOT



ValEncIA



VNODE



- Solvers perform differently for this class of problems
- C5 is considered the most important; e_u is exact
- ValEncIA performs in a similar way for all problems
- VNODE is the best solver for this class of problems

Conclusions

Developed PO verification guidelines help in the overall V&V process!

Necessary for V&V were:

- Flexible and interoperable data types and libraries
  → SmartMOBILE, UniVerMeC, Vericell
- Adaptive self-learning algorithms
- Dealing with uncertainty or bad/missing data
  → (Bio)mechanics in SmartMOBILE
- Uncertainty visualization
  → Future work!
- Ubiquitous distributed and parallel computing, crowd sourcing
  → CPU/GPU parallelization in Vericell
- User support...
  → Vericomp

Interoperability remains an important issue!

Outlook: Biomechanics

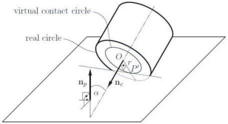
Interesting: foot contact problem

Reasons: switches → non-smooth functions

discontinuity sticking, etc...

verified treatment is infrequent

Advantages: e.g. contact area modeling with intervals



Tools: such tools as e.g. FLOW* are verified but need reimplementation for integration into SMARTMOBILE

Outlook: VERICELL and VERICOMP

Modeling, simulation and control of SOFC (VERICELL):

- Possibilities to speed up *.P3.* (full verification without approximation)
- Simplifications having analytical solutions that describe the system adequately
- More measurements for temperature $\rightsquigarrow$ better results for higher dimensions
- Computerized models for further subsystems

User support (VERICOMP):

- The system is not being maintained at the moment
- Automatic addition of solvers is difficult but necessary
- Solution of automatic layout problems necessary