

# Numerical Verification Assessment in Computational Biomechanics

Ekaterina Auer and Wolfram Luther

Department of Computer Science and Applied Cognitive Science  
University of Duisburg-Essen  
`auer,luther@inf.uni-due.de`

**Abstract.** In this paper, we present several aspects of the recent project PROREOP, in which a new prognosis system is developed for optimizing patient-specific preoperative surgical planning for the human skeletal system. We address verification and validation assessment in PROREOP with special emphasis on numerical accuracy and performance. To assess numerical accuracy, we propose to employ graded instruments, including accuracy tests and error analysis. The use of such instruments is exemplified for the process of accurate femur reconstruction. Moreover, we show how to verify the simulation results and take into account measurement uncertainties for a part of this process using tools and techniques developed in the project TellHIM&S.

**Keywords:** Numerical verification assessment, validation, uncertainty, result verification

## 1 Introduction

In the United States, there is a long tradition in the Computational Fluid Dynamics (CFD) community [17] and the American Society of Mechanical Engineers [2] of designing methodologies and of implementing and testing tools for the verification and validation (V&V) assessment. Recently, these efforts have been broadened to include application processes in biomechanics and biomedicine [1]. The authors of the overview papers cited define the terms verification and validation in the context of modeling and simulation, software engineering and numerical mathematics in engineering and physical applications. Moreover, they develop requirements for categorizations and classifications of processes as a result of precise assessment procedures. However, the known assessment methodologies do not provide a definitive step-by-step V&V procedure immediately applicable by the engineer. In the understanding of the key researchers in this field, all-encompassing procedures for obtaining proofs of correctness do not exist, and V&V activities can only assess the correctness or accuracy of specific (parts of) processes examined.

This voluntary limitation to a small part of the whole modeling, verification and validation cycle is not surprising. In CFD and solid mechanics, process models are based predominantly on continuous mathematics and the use of spatial

and temporal discretization or iterative solution methods (possibly with insufficient convergence). In such models, huge equation systems or partial differential equations are solved or finite element methods applied. The usual verification methodology is to look for analytical solutions and theoretical proofs of existence and convergence, which are generally difficult to obtain in the above-mentioned cases. One possible solution to these problems is to use computer-aided proofs, real number algorithms or algorithms with result verification. A recent collection of papers [9] demonstrates further ways of developing robust and reliable numerical software. However, most of these methods do not occur in standard V&V methodologies to a noticeable degree, not even as interval data types in the discussion of the uncertainty concept as highlighted in [13]. Interval arithmetic and result verification are not addressed at all. In fact, the verification instrument proposed there focuses only on analytic or benchmark solutions, making it very limited.

This community is not aware that several environments for validated modeling and simulation of the kinematics and dynamics of various classes of mechanical systems (e.g., NiceMOBILE or SmartMOBILE) already exist [4]. This is confirmed by the astonishing fact that numerical verification using interval tools was not considered in Verisoft (<http://www.verisoft.de/>), a recent project funded by the German Federal Ministry of Education and Research, either — even though Verisoft focuses on verifying the design of integrated computer systems. The correct functionality of systems is to be proved in selected application scenarios by using mathematical formalisms, formal program verification and model checkers.

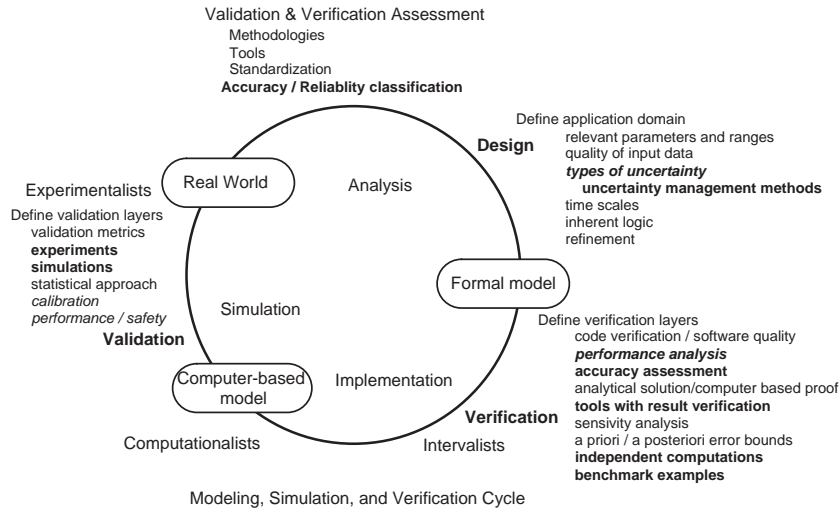
First and foremost, the existing methodologies deal with accuracy measured a posteriori in relation to benchmark solutions or model problems. Our suggestion is to perform elements of V&V assessment during the designing step. It should begin with the specification of the process and its subprocesses, the design of the building-blocks and software modules, the definition of interfaces and data flows and, finally, the selection and adaptation of appropriate data types and algorithms. Furthermore, it is important to define the tolerances present within the input parameters to obtain a certain accuracy of computation.

The paper begins with a short introduction to the modeling, simulation and verification cycle and defines several concepts important for further considerations. Then it proposes a new numerical verification taxonomy and discusses accurate femur reconstruction as an example. We show how to employ recently developed methodologies to transfer this process into a higher verification class and how to use SMARTMOBILE to verify the results of one of its parts.

## 2 Verification Assessment in Mechanics

In this section we describe three major steps in the verification and validation assessment process (cf. Figure 1). The first step is to analyze the real world problem and to design a formal model of the system under consideration. The

application domain is described, relevant parameters and their ranges defined and types of uncertainty in the model and its parameters identified.



**Fig. 1.** Validation & Verification Assessment and Addressed Topics (in bold) [19].

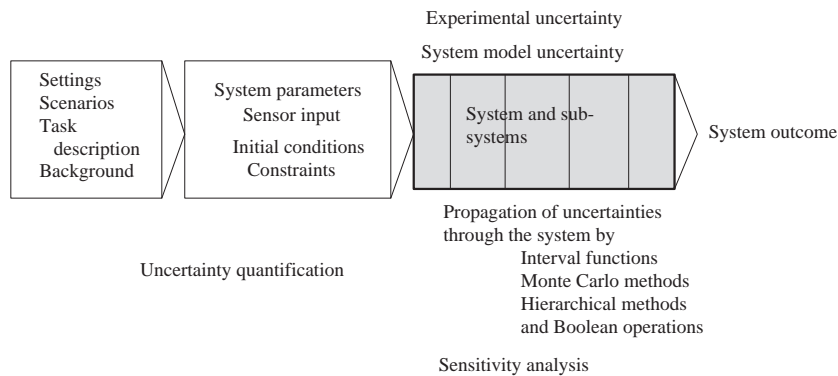
The second step is verification, which pursues two major goals: code verification — that is, finding logical and programming errors in the code — followed by numerical verification. We are mainly interested in the latter. We are currently developing a four-tier hierarchy that specifies the degree of verification of a given numerical program. A selection of graded instruments helping to identify the class to which the program belongs consists of instruments for subsystem cases and benchmark problems, for independent control computations and for the calculation of a priori or a posteriori error bounds. In addition, we use tools for sensitivity analysis to identify parameters that have a significant influence on the result as well as tools with result verification or computer-aided proofs of the existence of the solution.

Benchmark and subsystem cases are also used in the third and final step, validation. This step addresses model fidelity, defines a validation metric and compares the outcomes of simulations and experiments.

There are tasks in the cycle from Figure 1 which cannot be associated with only one step in the V&V assessment. One example is calibration which concerns the identification and adjustment of the model parameters. Usually, calibration is carried out to achieve a high degree of correspondence between the model and the experimental outcome. Similarly, performance issues influence both the simulation and the numerical algorithms, and the uncertainty problem affects the parameters, the model and/or the experiments.

We would like to emphasize that the notion of reliability is not used only in verified numerics but also in mechatronic systems. In the latter case, reliability refers to the probability that the system will provide correct service within a certain time period. System reliability goes hand in hand with further issues, e.g., safety and availability. Often, dependability is used as a synonym for reliability. Kochs [12] explains mechatronic dependability as a "qualitative and quantitative assessment of performance with regard to reliability and safety and taking into consideration all relevant attributes and factors." Dependability describes the ability of the system to provide specified services to the user.

To obtain system reliability, the V&V assessment has to deal with uncertainty. There are two types of uncertainty. Experimental uncertainty depends on the precision of measurements and can be characterized by varying initial values or parameters following a probability distribution, which may be unknown. Model or epistemic uncertainty causes unpredictable system behavior as a consequence of missing knowledge about the system or the environment [18].



**Fig. 2.** Uncertainty analysis.

Figure 2 shows methods for propagating uncertainties throughout the system. The imprecision in the outcome can be measured by providing bounds enclosing all possible results or by using probability theory or, alternatively, Dempster-Shafer belief theory [13]. If the probabilities are unknown, Bayes theory can be used. In most cases, it is necessary to compute accurate enclosures of interval functions  $f([a, b])$  where  $f$  denotes a system function and  $[a, b]$  is an arbitrary interval or interval vector built with machine numbers  $a$  and  $b$  to model the uncertainty on input parameters. This problem may be solved by using an interval optimization solver providing  $\inf f([a, b])$  and  $\sup f([a, b])$  or interval tools supporting the rounding to  $\pm\infty$ . Otherwise, Monte Carlo methods can be applied; they yield no overestimation, but they are time-consuming and the bounds obtained are not guaranteed [10].

At first glance, numerical verification based on interval enclosures does not provide any information on the fidelity of the computational model since the relationship between model and experiment is not being investigated at the verification stage of the V&V cycle in Figure 1. However, in several applications, we observed that, if the result enclosures obtained were wide or the verification procedure was unsuccessful, the models were often poor for that parameter range. In [14] we studied the identification and simulation of a hydraulic differential cylinder with servo valve. As a controller, a Sugeno-Takagi fuzzy model was used. It was shown that numerical verification was possible in cases where conventional simulation also yielded good results. Therefore, failure to verify a simulation is usually an indicator of a need to improve the corresponding model.

### 3 A Taxonomy For Numerical Verification Assessment

In this section we propose a numerical verification taxonomy. Furthermore, we give some guidelines for performing a verification assessment analysis. These guidelines help to characterize a computational model and its implementation from the point of view of numerical verification.

We introduce four classes, from lowest to highest certification standard. The lowest class covers only the use of a standardized floating point arithmetic and a detailed documentation of the results. The remaining higher classes address the fidelity of the model translation into a programming language and examine whether the numerical model implementation accurately represents the conceptual description of the model.

**Class 4:** The process implementation uses standard floating-point or fixed-point arithmetic; results are not verified.

**Class 3:** The system is subdivided into subsystems. The numerical implementation of the process uses at least standardized IEEE (P)754 floating-point arithmetic. Furthermore,

- sensitivity analysis is carried out for uncertain parameters; alternatively uncertainty is propagated throughout the subsystems using methods like Monte Carlo;
- a priori/posteriori error bounds are provided for important subprocesses; alternatively, self-correcting algorithms are used or numerical stability is proved; condition numbers are computed, and failure conditions identified.

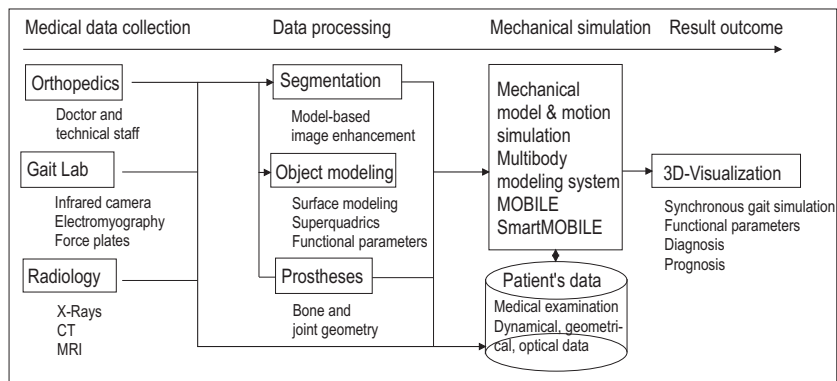
**Class 2:** Relevant subsystems are implemented using tools with result verification or delivering reliable error bounds. The tools use language extensions for scientific computation with standardized floating-point, (enhanced) interval, multiple precision (multiword) or stochastic arithmetic; the actual precision is computed at run-time according to the needs of input data and the predicted outcome. The convergence of numerical algorithms is proved via existence theorems, analytical solutions, computer-aided proofs or fixed-point theorems.

**Class 1:** Uncertainty is quantified and propagated throughout the process using interval or ensemble computing. Model parameters are optimized by calibration. The whole system is verified using tools with result verification. Basic numeric algorithms and (special) functions are certified. Alternatively, real number algorithms, analytical solutions or computer-aided existence proofs are used. Performance issues are addressed. Numerical verification is accompanied by code verification. Software and hardware comply with the IEEE 754 and follow a proposed interval standard.

This taxonomy helps us to establish standardized descriptions of application and validation domains, data capture and visualization methods, software building-blocks, data flow through the system, standardized interfaces and model descriptions within a heterogeneous hardware and software environment.

#### 4 A Case Study: Accurate Femur Reconstruction

In parallel to the considerations about the taxonomy from section 3, assessment management tools for biomechanical applications have been developed and used within the modeling process of parts of the human skeletal system to support hip and lower-limb surgery. An overview of the complete process flow is provided in Figure 3.



**Fig. 3.** Process flow in the PROREOP project.

Using a recently developed questionnaire, accuracy aspects of the data flow and of important subprocesses and algorithms are identified and analyzed. The algorithms and data exchange types are described in a standardized manner: input and output values with their significant digits, types of uncertainties and characteristic parameters of the algorithms with an impact on the accuracy and reliability (numerical data types, conversions, precision loss, cancellation, discretization and truncation errors, error accumulation, stability and condition

numbers, test cases) are calculated or estimated. This allows us to determine the level of V&V in the process according to the developed V&V taxonomy.

More precisely, our goal in this section is to determine how a parameterized model for efficient and accurate femur reconstruction using model-based segmentation and superquadric shapes [7] fits into the proposed V&V taxonomy. Computational model verification starts with an analysis of the collected patient data coming from medical examination in the gait lab and the radiology department, which is subsequently used in the PROREOP process flow. To quantify the quality of input data and the process uncertainty, the following questions have to be answered:

- Which data source(s) (raw data) is used to acquire the initial data? (MRI, CT, X-RAY, etc.)
- How is the initial data described, and are there any requirements for a generic data type definition in XML?
- Which kind of data selection/fusion was used? (Kind of sensor, camera, MRI, etc.)
- How accurate is this initial data? (Type of noise, one- or two-sided distribution, percentage of wrong or missing values, failure and redundancy, kind of pre-calculation by firmware program as used in MRI systems)
- How is the raw and initial data described? Is there a common global coordinate system?
- Is there a potential deficiency in the modeling process (model error), unknown data or a lack of knowledge concerning parameters or constraints?

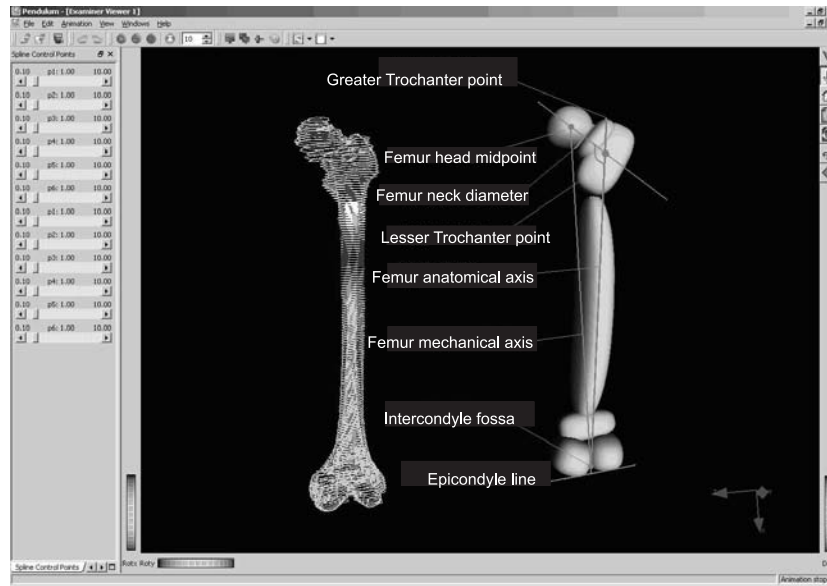
Then, the ability of the conceptual model and its translation into a computer-based representation has to be judged and the computational strategy characterized. Typical questions here concern the type of algorithms used (iterative, recursive, stochastic, geometric).

Next, a detailed description of mathematical operations actually used in the algorithm is collected. Additionally, concise information about the initial and resulting data types for these mathematical operations is gathered.

Finally, the outcome has to be analyzed: In which format is the resulting data described, and is there any need for a general format? Are there any known or anticipated failures in the resulting data or a systematic loss of precision?

This methodology was applied to study the process for segmenting, reconstructing and measuring a femur bone on the basis of a 3D MRI scan. The goal was to automatically extract important geometric bone features. (These features concerned geometrical measurements highlighted in Figure 4.)

The process was split into several building-blocks. First, a thresholding method to convert the original 3D MRI images to 3D binary images containing only bones and other tissues of the same intensity was used. The second step provided a region growing method to eliminate most tissues that were not bones. After region growing, the shaft of the femur was already sufficiently segmented, but the femur ball needed additional processing. Therefore, in the third step, we used a VRML model of a standard femur to further refine the binary image.



**Fig. 4.** Femur bone and SQ-model with typical features.

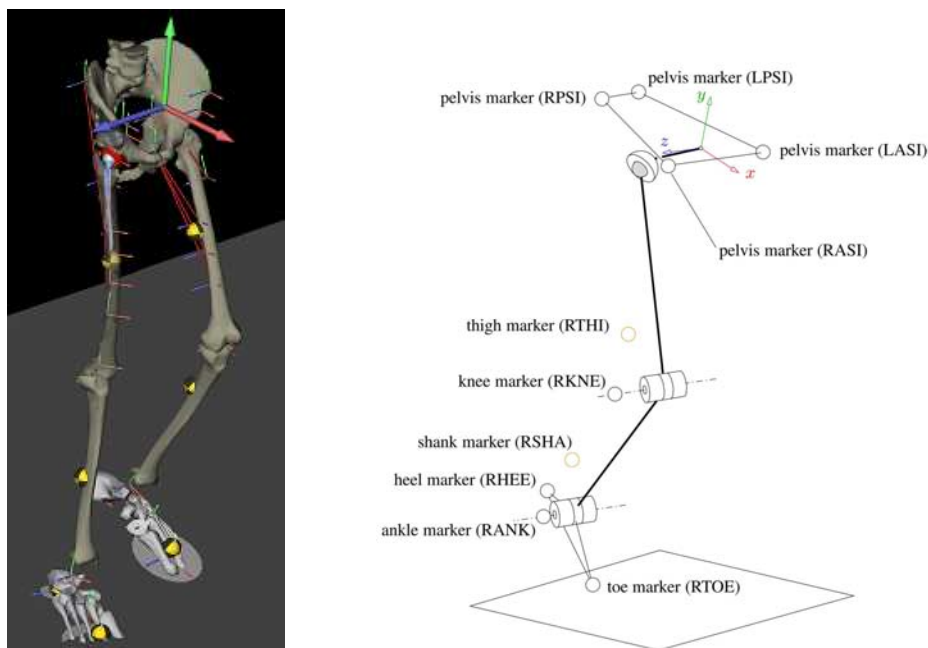
Then, a patient-specific superquadric (SQ) bone model was built. The basic SQ type possesses only eleven parameters, the type with local deformations eighteen parameters. In both cases the optimization process that fits the SQ to the data cloud contained considerably fewer parameters than B-spline surfaces. By using a solid modeling technique, we constructed a tree-similar object with SQ leaves. A split and merge approach was applied to reduce the number of SQs while maintaining the quality of the fit. Candidates for merging were chosen in such a way that the feature-oriented SQ model of the femur was constructed (Figure 4). From this model, significant points and quantities, like the mechanical length or the center of the femur head, could easily be extracted by using the orientation of the SQ within a global coordinate system and basic operations on the parameters. Together with the SQ-based approach, a manual extraction of the visualized patient data and a parallel calculation based on the VRML model delivered three independent computations of the patient-specific bone features and justified the implementation to be put into Class 3 of the V&V taxonomy [7].

The reconstruction of the bones of the hip and lower limbs is then used together with marker data coming from a gait lab to build a patient-specific mechanical model and motion simulation. To this end, reasonable bounds for the knee and hip joint positions are needed.



## 5 Verification of Body Segment Motion in SMARTMOBILE

One of the not yet solved problems in the project PROREOP is the characterization of artifacts induced by skin motion that directly influence the position of markers with respect to the bones and joints during the experiments in the gait lab (see Figure 5). We propagated uncertainties induced by these artifacts by applying the modeling and simulation tool SMARTMOBILE to the identification of body segment motion. SMARTMOBILE provides results guaranteed to be correct within the constraints of the considered computational model of a mechanical system.



**Fig. 5.** Skeleton-hip prosthesis — markers relevant for identification of body segment motion.

In this section, we give a short description of SMARTMOBILE and then turn to the problem of identification of body segment motion from marker trajectories. In conclusion, we mention the recent developments concerning validated sensitivity analysis.

SMARTMOBILE [4] is a C++ object-oriented software for verification of various classes of mechanical systems. This tool is based on MOBILE [11] which employs usual numerics. Models in both tools are executable C++ programs built of the supplied classes for transmission elements such as rigid links for

modeling of rigid bodies, for scalar or spatial objects such as coordinate frames and for solvers such as those for differential equations.

SMARTMOBILE is one of the first integrated environments providing result verification for kinematic and dynamic simulations of mechanical systems. The advantage of this environment is its flexibility due to its template structure: the user can choose the kind of (non)verified arithmetics according to his task. An overview of arithmetics available in SMARTMOBILE at this moment is given in Table 1. However, advanced users are not limited to them and are free to plug in their own implementations if they follow the general instructions given in [3].

MOBILE users are assisted in switching to SMARTMOBILE by several converters. The first type helps to transform new MOBILE packages into SMARTMOBILE templates via series of automatically generated LINUX scripts. The second type converts already existing MOBILE models into the form defined by SMARTMOBILE. Elements generated by both converter types might require a heuristic improvement by the user, if they contain code fragments transformation of which cannot be automatized, for example, non-verified equation solvers.

For most kinematical problems, it is sufficient to use the basic data type from Column 3 of the Table 1 as the parameter of all the template classes used for a particular model. The main idea for dynamic and special kinematical tasks such as finding of system equilibria is to use pairs basic data type/corresponding solver (Columns 3 and 4). Here, the basic data type should be constructed in such a way as to allow us to apply the given solver. That is, it should automatically deliver, for example, all the derivatives the solver requires.

Our experience shows that the general tendency as to what kind of arithmetic to use is as follows. If only a reference solution is of interest, floating point arithmetics with `MoReal` and a usual numerical integrator such as Runge-Kutta's can be employed for dynamic simulations. If the user is interested in fast verification of a relatively simple system with little uncertainty, interval-based pairs are of use. Taylor arithmetic should be chosen mostly for offline simulations with more uncertainty [4].

**Table 1.** Arithmetics supplied with SMARTMOBILE.

Description	Arithmetic	Kinematic	Dynamics
reference	floating point	MoReal	MoRungeKutta,...
based on VNODE [15]	intervals	TMoInterval	TMoAWA
based on VALENCIA-IVP [5]	intervals	TMoFInterval	TMoValencia
based on RiOT [8]	Taylor	TMoTaylorModel	TMoRiOT
based on COSY [6]	Taylor	RDAInterval	—
equilibrium states	intervals	MoFInterval	MoIGradient
sensitivity with VALENCIA-IVP	intervals	MoSInterval	TMoValenciaS

Now we consider the problem of identification of body segment motion using marker trajectories. A subtask of this problem, for which a new algorithm has been developed recently, is the reconstruction of the hip joint position from

positions of markers fastened to specified places on a patient's leg (Figure 5, right side). The corresponding model is purely kinematic. At first, the segment frame motion is obtained by orthogonalizing the bone and joint axes sequentially. In the second step, the model parameters and the motion of the model segments are adjusted to the marker trajectories using nonlinear optimization. For the purposes of verification, fitting task of this second stage was simplified in such a way as to be explicitly solvable.

The data on marker positions contains uncertainties which appear, for example, due to various kinds of skin displacement during motion. The task of quantifying the influence of such uncertainties on the simulation result was to be solved in SMARTMOBILE. The uncertainties of interest amounted to  $\pm 10$  mm in both knee and ankle widths. Besides, uncertainties due to skin displacements underneath markers were to be taken into account. The displacement tangential to skin could amount to up to  $\pm 10$  mm, normal to skin up to  $\pm 5$  mm, and marker displacement due to soft tissue movement was again  $\pm 10$  mm. The nominal values of these parameters in the model were 120 mm for the knee width and 80 mm for the ankle width. The femur length obtained with MOBILE for the system without uncertainties was 0.3863 m (rounded up to the fourth digit after the decimal point). We were interested in the influence of the uncertainties on the hip joint position and the length of the femur bone. Only the right leg was considered; the algorithm works for the left leg analogously.

At first, we transformed the corresponding MOBILE package into SMARTMOBILE. After this semi-automatic procedure, we tried to use interval arithmetic on the model obtained in this way. These first results were discouraging: to begin with, we could not work with the above mentioned uncertainties. Already an uncertainty of  $\pm 0.6$  mm in the knee width gave us the result interval of over 30 m (!) in diameter, which was of course meaningless in the context. This indicated, on the one hand, that the automatically generated classes had to be improved by an expert, and, on the other hand, that a different kind of verified arithmetic had to be used.

There were several code fragments which could be rewritten in such a way as to produce less overestimation. The main problem with the old code, however, was the routine which computed the solution of a linear system of equations  $Ax = b$  directly by inverting the matrix  $A$ ,  $x = A^{-1}b$ , which is a well-known source of overestimation in the interval case. It was not the task of our converter program to detect such code fragments, and, generally, this cannot be easily done. After having solved this system by a corresponding routine from PROFIL/BIAS for intervals and having performed other smaller improvements, we could reduce diameter of the resulting interval to approximately 20 m for the initial uncertainty of  $\pm 0.6$  mm in the knee width, which was still not good enough from the practical point of view.

Our next step was to use Taylor models instead of intervals for this simulation. However, we were confronted with limitations imposed by Taylor arithmetic libraries RIOT and COSY. The former does not implement the inverse sine function  $\arcsin x$  which was employed in the original algorithm; the parameters of

the latter concerning memory management had to be thoroughly tuned to make it work with the big amount of measurement data. The results of simulations with COSY are shown in Table 2. (They are rounded to the fourth digit after the decimal point.) Taylor models were bounded by the LDB algorithm from COSY (the linear dominated bounder) to obtain upper and lower bounds on the overall uncertainty. Although this simulation took more CPU time than the interval one<sup>1</sup> and considerably more time than the non verified simulation in MOBILE, it was still presumed to be faster than the corresponding series of Monte Carlo simulations.

**Table 2.** Verified identification of body segment motion under uncertainties: hip joint position and femur length in m.

Uncertainty	Position (x,y,z)	Femur length
Knee, ankle	([-0.0805;-0.0697], [-0.0602;-0.0452], [0.0403; 0.0701])	[0.3776; 0.3967]
Skin displacement	([-0.3351;0.1939], [-0.3628; 0.2319], [-0.0066; 0.1158])	[0.0000; 0.6214]

We see from Table 2 that the uncertainties in knee and ankle widths result in the uncertainty of approximately 20 mm in the femur length (Line 2). The overall uncertainty due to skin movement has a greater impact on the system: the diameter of the best possible enclosure of the femur length is approximately 622 mm. This result indicates the need to perform all corresponding measurements with great care if the proposed algorithm is to be used. On the other hand, it might be worth while to devise an algorithm that would be less sensitive to marker displacements.

The displacement due to soft tissue movement has the biggest influence, the displacement normal to skin the smallest, as shown in Table 3. Here, the femur length was measured under  $\pm 5$ ,  $\pm 10$  and  $\pm 20$  mm uncertainty individually for each kind of displacement.

**Table 3.** Sensitivity of the model with respect to marker displacements due to skin movements: femur length (m).

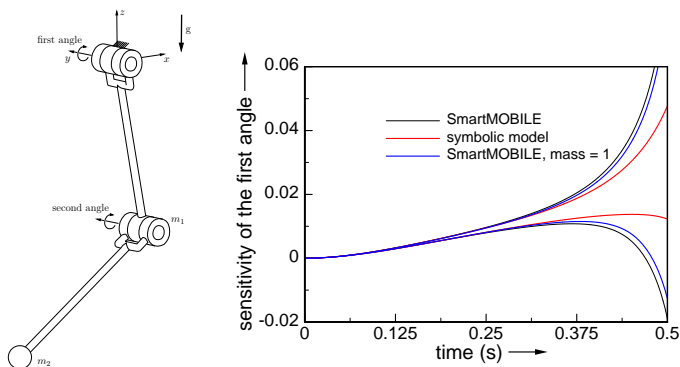
Marker displacement	[-5;5]	[-10;10]	[-20;20]
tangential to skin	[0.3492;0.4203]	[0.3008;0.4576]	[0.1146;0.5468]
normal to skin	[0.3742;0.3985]	[0.3279;0.4413]	[0.3335;0.4427]
soft tissue	[0.3593;0.4125]	[0.3279;0.4413]	[0.0000;0.6330]

The fact that Taylor arithmetic was so much more successful than the interval one is not astonishing since the proposed algorithm in MOBILE has a lot of cancellation in the sense described in [16]. However, since what we were

<sup>1</sup> 36.1 s versus 0.4 s for a simulation with  $\pm 10$  mm uncertainty in the knee width on Intel Xeon CPU 2GHz under Linux 2.6.23.14-115.fc8

interested in was essentially an enclosure of the range of a function, the actual final results were negatively influenced by the use of the LDB algorithm bounding the range of a Taylor model. Therefore, our future task will be to employ a (Taylor model based) optimizer instead. Another interesting direction will be to compute the overall uncertainty using a measure based on differential sensitivity. A tool similar to the one described in the next paragraph for dynamic systems can be employed for this purpose. In this way, we will be able to quantify the overestimation in the SMARTMOBILE simulation. The employment of result verification for accurate femur reconstruction allows us to place it in Class 2 of the V&V taxonomy proposed in section 3.

A new development in SMARTMOBILE important for V&V analysis of dynamic systems is the class `TMoValenciaSIntegrator`, which provides validated sensitivities of all states with respect to parameters of interest. This class is based on the corresponding algorithm from VALENCIA-IVP. Sensitivity in this case is understood as the partial derivative of a given state with respect to a certain parameter. As an example, we consider the double pendulum from [5] with the uncertainty of  $\pm 1\%$  in the first initial angle. In Figure 6, its sensitivity with respect to the first mass  $m_1$  is shown. The blue curve shows results obtained with SMARTMOBILE for  $m_1 = 1$  kg, the grey one for  $m_1 \in [0.99; 1.01]$  kg. For comparison, results for the corresponding symbolic equations with  $m_1 \in [0.99; 1.01]$  kg from VALENCIA-IVP are represented by the red curve.



**Fig. 6.** Sensitivity of the double pendulum wrt. the first mass.

The enclosures obtained in VALENCIA-IVP are tighter, which is not very remarkable since the symbolic model contains less numerical operations and is therefore less prone to overestimation. The enclosures for certain and uncertain masses do not differ much over this time interval. It lets us conclude that mass uncertainty does not contribute as much as the uncertainty in the initial conditions to the overall overestimation which shows itself in the continuous widening of enclosure widths over time. The sensitivity is increasing in the considered in-

terval, but its values are small. That leads to the conclusion that this parameter does not influence the simulation for  $t \in [0; 0.5]$  substantially.

## 6 Conclusions and Further Work

In this paper we proposed a new taxonomy of numerical verification and provided guidelines for performing a verification assessment analysis that allowed us to associate a computational model and its implementation with a certain V&V class. The methodology was demonstrated by the example of accurate femur (motion) reconstruction. A part of this process could be verified using SMARTMOBILE.

In the future, we plan to place this research in a more general context and work on an assessment framework in the form of step-by-step instructions for performing an assessment analysis to classify (bio)mechanical processes and their computational realizations.

## References

1. A. E. Anderson, B. J. Ellis, and J. A. Weiss. Verification, validation and sensitivity studies in computational biomechanics. *Computer Methods in Biomechanics and Biomedical Engineering*, 10(3):171–184, June 2007.
2. ASME Committee (PT60) on Verification and Validation in Computational Solid Mechanics. Guide for verification and validation in computational solid mechanics, 2006.
3. E. Auer. *SmartMOBILE: A framework for reliable modeling and simulation of kinematics and dynamics of mechanical systems*. PhD thesis, Universität Duisburg-Essen, WiKu Verlag Dr. Stein, 2007. ISBN: 978-3-86553-240-4.
4. E. Auer and W. Luther. SMARTMOBILE — An environment for guaranteed multibody modeling and simulation. In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics ICINCO*, 2007. ISBN: 978-972-8865-87-0.
5. E. Auer, A. Rauh, E. P. Hofer, and W. Luther. Validated Modeling of Mechanical Systems with SMARTMOBILE: Improvement of Performance by VALENCIA-IVP. In *Proc. of Dagstuhl Seminar 06021: Reliable Implementation of Real Number Algorithms: Theory and Practice*, Lecture Notes in Computer Science LNCS 5045, pages 1–28, 2008.
6. M. Berz and K. Makino. COSY INFINITY 9.0. Programmer’s manual. Technical Report MSUHEP 060803, Michigan State University, 2006.
7. R. Cuyppers, Z. Tang, W. Luther, and J. Pauli. A parametrized model for efficient and accurate femur reconstruction using model-based segmentation and superquadric shapes. In *I-ASTED*, Baltimore, April 2008.
8. I. Eble. RiOT. Available through the author: <http://iamlasun8.mathematik.uni-karlsruhe.de/~ae08/>.
9. Bo Einarsson, editor. *Accuracy And Reliability in Scientific Computing*. Society for Industrial and Applied Mathematics, U.S., 2005.
10. Chr. Hörsken and H. Traczinski. Modeling of multibody systems with interval arithmetic. In Walter Krämer and Jürgen Wolff v. Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 317–328. Kluwer, 2001.

11. A. Kecskeméthy. *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen*. PhD thesis, Gerhard Mercator Universität Duisburg, 1993.
12. H.-D. Kochs. Key factors of dependability of mechatronic units: Mechatronic dependability. In *28th Annual International Computer Software and Application Conference (COMP-SAC 2004)*, pages 584–586, Hong Kong, China, 2004. IEEE Computer Society 2004.
13. V. Kreinovich, J. Beck, C. Ferregut, A. Sanchez, G. R. Keller, M. Averill, and S. A. Starks. Monte-Carlo-type techniques for processing interval uncertainty, and their engineering applications. In *Proceedings of the Workshop on Reliable Engineering Computing*, pages 139–160, Savannah, GA, September 2004.
14. W. Luther, E. Dyllong, D. Fausten, W. Otten, and H. Traczinski. Numerical verification and validation of kinematics and dynamical models for flexible robots in complex environment. In U. Kulisch, R. Lohner, and A. Facius, editors, *Perspectives on Enclosure Methods*, pages 181–199. Springer, 2001.
15. N. S. Nedialkov. *The design and implementation of an object-oriented validated ODE solver*. Kluwer Academic Publishers, 2002.
16. A. Neumaier. Taylor forms — use and limits. *Reliable Computing*, 9:43–79, 2002.
17. W. L. Oberkampf, T. G. Trucano, and C. Hirsch. Verification, validation, and predictive capability in computational engineering and physics. Technical Report SAND2003-3769, Sandia National Laboratories, 2003.
18. ESReDA Project Group on Uncertainty Management. *Uncertainty in Industrial Practice — A guide to quantitative uncertainty management*. Wiley, Chichester, United Kingdom. To appear.
19. S. Schlesinger. Terminology for model credibility. *Simulation*, 32(3):103–104, 1979.