Verification and Validation for Femur Prosthesis Surgery

Ekaterina Auer, Roger Cuypers, Eva Dyllong, Stefan Kiel, Wolfram Luther[1]

Computer and Cognitive Sciences (INKO)
University of Duisburg-Essen
Duisburg, Germany
Email: {auer, cuypers, dyllong, kiel, luther}@inf.uni-due.de

Abstract. In this paper, we describe how verified methods we are developing in the course of the project TellHim&S (*Interval Based Methods For Adaptive Hierarchical Models In Modeling And Simulation Systems*) can be applied in the context of the biomechanical project PROREOP (*Development of a new prognosis system to optimize patient-specific preoperative surgical planning for the human skeletal system*). On the one hand, it includes the use of verified hierarchical structures for reliable geometric modeling, object decomposition, distance computation and path planning. On the other hand, we cover such tasks as verification and validation assessment and propagation of differently described uncertainties through system models in engineering or mechanics.

Keywords: Graphical interface construction, superquadrics, 3D modeling, biomedical engineering

1 Introduction

Worldwide, there exist many tools for total hip replacement (THR) planning which are utilized in real life surgeries. These tools employ 3D surface models to assist surgeons during operations or to allow them to operate virtually [1], [2]. However, the traditional methods have several drawbacks. For example, surgeons often have to use 2D imagery for 3D reconstruction or employ standardized bone and muscle models scaled only roughly for the individual patient.

The aim of a recent project PROREOP *Development of a new prognosis system to optimize patient-specific preoperative surgical planning for the human skeletal system* (2007-2008) has been to overcome the drawbacks of a classical computer assisted 2D or 3D prosthesis surgery planning. Its important parts were the implementation of a tolerance-compliant selection procedure based on appropriate prosthesis features and the modeling of a bone-prosthesis fitting into the medullary space of the already routed femoral shaft. In a subproject concerning reliable bone modeling we addressed verification and validation (V&V) assessment in PROREOP with special emphasis on numerical accuracy and performance. We employed patient-specific MRI-, CT- and X-Ray data of the human

Dagstuhl Seminar Proceedings 09471
Computer-assisted proofs - tools, methods and applications
http://drops.dagstuhl.de/opus/volltexte/2010/2513

pelvis and lower limbs in combination with analytically described models with robust geometrical parameters.

Studies on robustness and verification-specific aspects in PROREOP would not have been possible without the research made in another current project, TellHim&S — *Interval Based Methods For Adaptive Hierarchical Models In Modeling And Simulation Systems* (since 2006). On the one hand, it covers the use of verified hierarchical structures for reliable geometric modeling, object decomposition, distance computation and path planning. On the other hand, we study the possibilities of numerical result verification for dynamic multibody systems with the help of the appropriate modeling and simulation software. Biomechanics is an important new field where the approaches developed in TellHim&S can (and need to) be applied. For example, they help to deal with uncertainties in an effective way. In this paper, we describe the research in both parts of this project and demonstrate afterward how the results can be used in the PROREOP context.

The mechanical interdependencies of the musculoskeletal system can only be reconstructed by employing data collected in a gait lab in addition to static measurements and image sequences analysis. Typical parameters for THR are leg length, femoral offset and the angle between the axes of the femoral neck and shaft (so-called extraosseous aspects of the reconstruction [3]), prosthetic acetabular center, femur length, fit-and-fill limitations as well as contact constraints (intraosseous aspects concerning e.g. the femoral canal). The center of the prosthetic femoral head should coincide with the center of the acetabular cup.

All these parameters are influenced by uncertainty. Among all different kinds of uncertainty, two general types can be distinguished. Aleatory uncertainty refers to a variability type similar to that arising in games of chance. It cannot be reduced by further empirical study. Epistemic (reducible) uncertainty refers to the incertitude due to the lack of knowledge. An example is the absence of evidence about the probability distribution of a parameter. This is the type of uncertainty we have to deal with in PROREOP.

The imprecision in the outcome can be quantified by providing bounds enclosing all possible results, by using probability theory or Dempster-Shafer evidence theory (DST). For the first possibility, a range of tools is offered by the program SmartMOBILE [4]. For example, it allows the user to compute an enclosure for the length of the femur bone given the measuring uncertainties in the positions of markers attached to a human leg in order to identify the bone segment motion. Steps toward full application of the DST in the biomechanical context were made by implementing the DSI Toolbox (Dempster-Shafer with intervals) [5]. It is based on the previously developed IPP Toolbox [6] and designed to define, aggregate and evaluate precise Dempster-Shafer structures by using directed rounding and verified methods made accessible in MATLAB by the INTLAB library [7].

To represent surfaces of bones and muscles, there exist several modeling methods, which include approaches based on splines, hierarchic volumes or scene graphs. In the already mentioned subproject of PROREOP, we relied primarily on hybrid hierarchic and scene graph based models in addition to accurate distance algorithms. In particular, we considered CSG trees the leaves of which contained additional information represented by implicit equations, polyhedra, superquadrics (SQ) or free formed surfaces. Here, the methods developed in TellHim&S for working with such trees as well as proximity queries and distance computations between interval-based CSG-octrees can be applied. All these methods are implemented using floating point arithmetic and an interval based adaptive hierarchic model in parallel. If the global and local properties of this type of models are used, it is possible to determine functional parameters and valid points automatically and accurately by using surface data that is simultaneously segmented and marked up with respect to object components. These data should be completed by mechanical and material parameters in order to construct distance measures between bones and endoprostheses.

To cover all the above mentioned topics, we structure the paper as follows. After this introduction to the background of our process verification approach, we overview the application context briefly. Then, we describe reliable hierarchical structures and provide an example of their use in biomechanics. In section 4, we touch upon our verification and validation management scheme with the four-tier classification for subprocesses briefly and introduce the newly developed DSI Toolbox and the verified modeling and simulation tool SmartMOBILE. In section 5, we concentrate on the uncertainty management using the example of the femur prosthesis. We conclude by recapitulating the main results in the project TellHim&S and providing a perspective for the future research.

2 A Short Overview of the Application Context

The main goals of PROREOP were to construct a repository with three-dimensional bone and muscle segmentation data, to realize a kinematical feature extraction from superquadric bone models, to implement tolerance-compliant selection procedure based on appropriate prosthesis features, and to model a bone-prosthesis fitting into the medullary space of the already routed femoral shaft. The common practice while selecting an optimal implant is to chose the biggest available components that still fit into the carved femoral canal. The fitting process comes down to minimizing the distance between the canal and the implant stem. Because the set of available combinations of components is usually a discrete space and the number of reasonable options is small, the process can be optimized by starting multiple implant-canal fitting processes in parallel and selecting the best available combination on completion.

In this section, we describe two subprocesses in PROREOP. First, we overview the task of bone modeling with superquadrics. Then we turn to the fitting process.

2.1 Bone Modeling with Superquadrics

Superquadrics are a family of geometrical shapes that are defined by the implicit equation

$$F(x,y,z)=\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}}+\left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}}=1\ , \qquad (1)$$

where (a_1, a_2, a_3) is the spatial scaling, $(\varepsilon_1, \varepsilon_2)$ are the roundness and (x, y, z) are the position of a point in 3D space. The formula (1) is commonly referred to as inside / outside function (IO). It can be used to formulate certain types of distance functions efficiently. Due to the wide variety of shapes that can be represented by SQs, they are a convenient option for the modeling, decomposition and measurement of human bones and prostheses. Moreover, IO-functions allow us to check for collisions within CSG trees easily. Besides, we can use them as a basis for spatial selection tools.

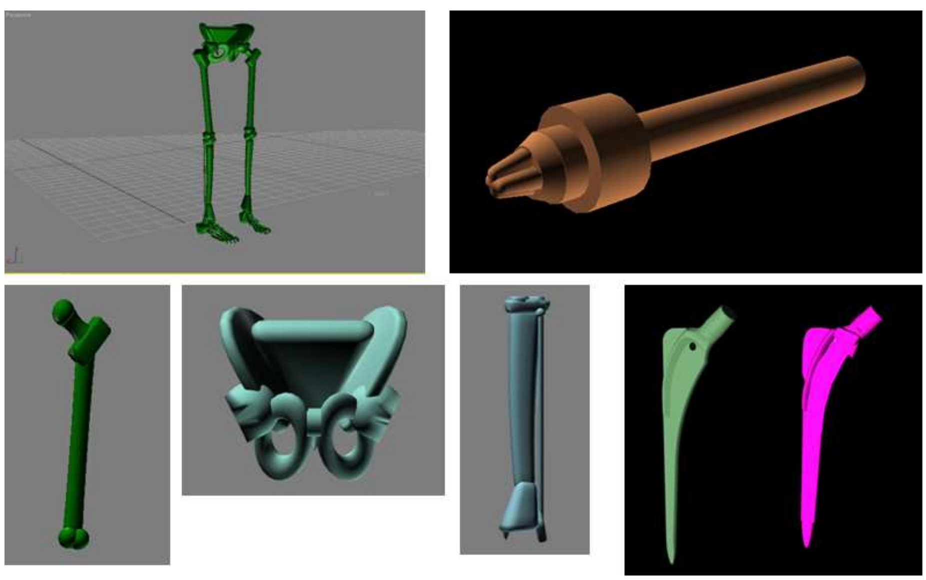

Fig. 1. Overview of bones/implants/tools reconstructed using superquadrics

Every model in Fig. 1 consists of a collection of superquadric shapes, whose global and local structure is defined by mathematical constraints on the components' parameters which are based on anatomical and mechanical interdependencies between adjacent bone or prosthesis parts. The models are reconstructed from point data extracted from patient-specific MRI-, CT- and X-Ray data. This data is segmented using support vector machine (SVM) methods, then decomposed and converted to the model with an SQ-based semi-automatical or automatical algorithm [8]. The global and local properties of this type of models can be used to determine functional bone and prosthesis parameters and valid points automatically and accurately by using surface and decompositional data. This data can be completed by mechanical and material parameters in order to construct distance measures between bones and endoprostheses. Currently, we can determine the anatomic size of the femur length with an absolute error of 1/10 percent.

2.2 Bone-Prosthesis Fitting

The general aim of a THR surgery is to align the implant so that the fitness with the femoral stem is maximized. Among the criteria that define the fitness value are:

- Minimization of the distance between the mechanically relevant areas of the implant and the medullary space;
- Maximization of implant component size without exceeding the space given by the uncovered medullary space opening.

Basically, to find an optimal fit, it is necessary to solve a minimization problem for the distance between the implant and the femoral stem. Depending on the geometric model used for the implant and the femoral bone, this involves the repeated calculation of the distance between multicomponent SQ models or a multicomponent SQ model and a point cloud. What we are looking for is a transformation $T = (x_w, y_w, z_w, \varphi, \theta, \psi, s)$ that minimizes the corresponding distance integral. Here, (x_w, y_w, z_w) is the global position, (φ, θ, ψ) the orientation in ZYZ Euler angles, and s the scaling of the implant. The stem size is considered to be constant. The distance minimization problem is stated as:

$$\underset{T \in \mathcal{R}^7}{\text{minimize}}\ F(T) \equiv \int\!\!\int_S |d(p_M, T_{p_I})| \, dS \tag{2}$$

$$\text{mit } d(p_M, T_{p_I}) \geq 0 \qquad \forall p_I \ , \tag{3}$$

where S is the implant surface, p_I is a point on the implant surface, p_M is a point on the medullary space canal an d is the distance model used for the stem/implant distance. Among the adequate distance models for SQs are the Euclidean and Radial Euclidean Distance.

Since the implant has to be navigated through the opened spongiosa and medullary space, more bone substance has to be removed than is necessary for a tight fit of the implant, especially in the areas of neck and most proximal shaft. Therefore, the general proximity demands are reduced to those of a tight closeness in areas which are critical for a stable fit, like the distal part of the shaft. This requires the decomposition of the implant into different areas with respective priorities. The above method was applied in several tests. The mean difference between the positioned implant and the bone was measured to be between 1 mm (critical areas) and 5 mm (less critical areas). An extensive clinical evaluation of the proposed methods and a comparison to manual methods has yet to be done.

3 Hierarchical Models

In the first part of TellHim&S, we focus on interval methods for hierarchical structures. This includes but is not limited to geometric modeling, verified decomposition of models, distance computation and collision detection.

During the past stages of TellHim&S, we mostly developed CSG-models with quadric primitives and octree-encoded objects. Although both methods are suitable for modeling of a wide variety of shapes, they are inadequate for application in the scope of such complex applications as PROREOP, for which more elaborate modeling primitives are required. However, complex primitives can suffer from overestimation if evaluated with standard interval arithmetic (IA). The use of more sophisticated arithmetics such as affine arithmetic or Taylor models often reduces this effect. In the first step of the current TellHim&S phase, we enhance our hierarchical decomposition structures with these new arithmetics and prepare them for the new modeling types.

In the second step, we plan to extend CSG-trees with arbitrary implicit objects such as superquadrics, hyperquadrics (HQ) or free form surfaces as primitives. Nowadays a lot of models are described with polyhedrons. Although we can already represent convex polyhedral models using the implicit surface approach as an intersection of hyperplanes, we intend to add special handling routines for these models considering their wide use in practice (section 3.4)

Many geometrical tasks become more complicated as the complexity of models grow. For example, verified distance computation between two bones modeled with SQs or HQs is very demanding in general. Using hierarchical decomposition techniques, we subdivide the model into simpler parts. In this way, we can solve the task by working with simple structures. Additionally, it is easier to prune parts of the search region not containing a solution in this case.

For these complex objects, simple subdivision schemes such as interval-octrees often yield large areas of uncertainty, where we can not decide whether they belong to the object. To obtain a tight enclosure of the distance, we need appropriate decomposition methods which reduce this effect. While we do not intend to move away from the axis-aligned box structure formed by an interval-octree, we plan to integrate pruning techniques for reducing the uncertainty (section 3.5). These methods would allow us to preserve the advantages of the axis-aligned octree-cells, while improving the accuracy without increasing the maximum subdivision depth.

In this section, we describe our integrated framework for geometrical computations in detail and provide an example for its application within the context of PROREOP.

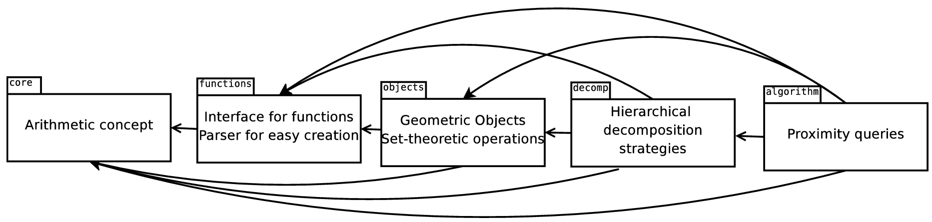

Fig. 2. Basic structure of the framework for verified geometric computations

3.1 An Integrated Framework

The subdivision methods and verified arithmetics can be combined in many ways (cf. Table 1). Users have to choose an adequate method depending on their current goals (e.g. speed, accuracy, space) and the considered models.

Table 1. Possible combinations of subdivision techniques with different arithmetics

Subdivision \ Arithmetic	IA	AA	TM
Octree	X	X	X
LIETree	-	X	X

Therefore it is necessary to implement a framework which would allow us to combine the techniques easily. It would simplify the testing process and ensure that test results are *comparable*. Below, we propose such framework covering all project goals. We need to support modeling, verified decomposition and implementation of algorithms (cf. Fig. 2). The framework consists of five components:

`core` Foundation of the framework, *arithmetic concept*
`functions` Interface for definition of single-valued analytic functions
`objects` Combination of functions to semi-analytic sets using CSG-like set-theoretic operations
`decomp` Various decomposition strategies
`algorithms` Applications for example distance computations, collision detections

All goals of our project can be mapped to components in our framework as shown in Fig. 3. Note that most goals are handled by more than one component. In subsections 3.2-3.6, we describe the five components in detail.

3.2 The `core`-Package

The `core`-package contains thin wrappers for several external libraries. Its main purpose is to provide an interface for different arithmetic types to handle them

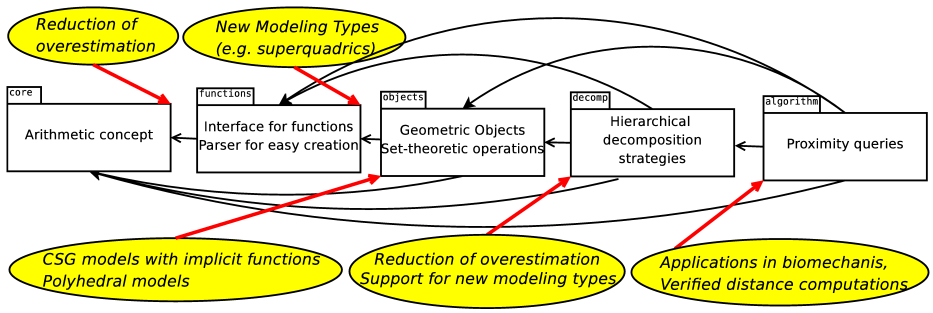

Fig. 3. The framework and the TellHim&S goals (yellow)

as uniformly as possible. We implement it by defining a *concept*. The concept describes operations which should be provided by any arithmetic type in our framework. Functions from a higher layer (e.g. `function`-package) can work on any arithmetic as long as operations used by them belong to the common set defined by the concept. Owing to the use of C++ templates, we do not have to employ an abstract base class for the different arithmetic types. This helps to avoid the virtual dispatch call which would be necessary for every arithmetic operation otherwise.

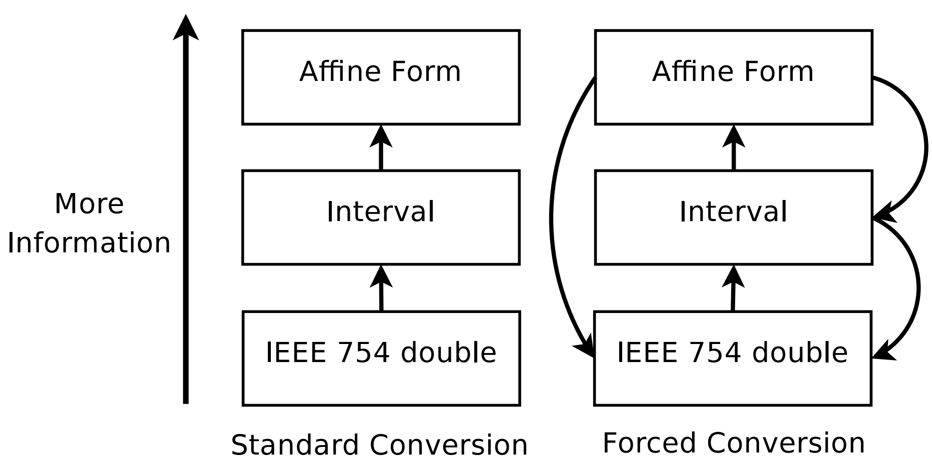

Fig. 4. Standard and forced conversion operators

Another important point is the conversion between different arithmetic types. Currently we support standard IEEE 754 double, interval arithmetic and affine arithmetic. We define two conversion operators: standard and forced conversion. The standard operator supports only conversions which do not result in a loss of information. For example, we can convert a double value to a (degenerate) interval. However, it is impossible to convert an affine form to an interval without the loss of the dependency information in general. Such conversions can be made only using the forced conversion operator (Fig. 4).

3.3 The `function`-Package

The main purpose of the `function`-package is to define single-valued analytic functions. In our current implementation, implicit functions for object modeling are defined here. The package consists mainly of a set of interfaces that define operations to be supported by a function in our framework. These operations include evaluation of a function for every supported arithmetic and several auxiliary methods for obtaining textual representations, dimensions of the input vectors, etc.

Functions are defined using a template class which can be parametrized with the actual logic via a functor. The functor object contains only the evaluation logic and should be implemented using a template function. This implementation detail allows the user to add new arithmetic types easily if the functors are appropriatly designed according to the arithmetic concept. Future implementations will ensure this constraint automatically.

Moreover, we provide a parser to simplify generation of functions. This allows us to test various decomposition schemes on different surfaces easily. Besides, automatic differentiation using fadbad++ ([9]) is provided for the functions generated through the parser.

3.4 The `objects`-Package

The `objects`-package has two main purposes. First, we provide modeling tools for more complex models than implicit surfaces. Second, it adds an extra abstraction layer for new input sources (e.g. polyhedral models) to the framework.

The first purpose is achieved by providing set-theoretic operations. They allow us to combine the surfaces generated by implicit equations to more complex models. Here the set-theoretic operations are the commonly used CSG-operations union, intersection, difference and negation.

```
<<interface>>
IGeoObj
-------------------------------------------------------------------
+<<const>> primitives(): unsigned
+<<const>> functions(): unsigned
+<<const>> cf(): const IFunction&
+<<const>> simplify(in args:core:arith::ivector): auto_ptr<IGeoObj>
```

Fig. 5. Interface for a geometric object in our framework

To cover the second task, it is necessary to implement an abstract and uniform interface for a geometric object. The `IGeoObj` interface shown in Fig. 5 treats every object as a set and defines a characteristic function on this set. The characteristic function can be evaluated with every arithmetic provided by the `core`-package. Using this method, we can incorporate modeling types not easily representable by the implicit surface approach into the framework. Polyhedral models are of special interest here. Currently, we can represent polyhedra as an intersection of halfspaces defined by implicit hyperplane equations. In the future, we will provide a more sophisticated implementation at a higher level via the `IGeoObj` interface.

3.5 The `decomp`-Package

Various tasks can be simplified using hierarchical decomposition schemes. We can use different techniques depending on the requirements on accuracy, speed and the modeling technique. The `decomp`-package includes all supported decomposition schemes. Decompositions should work with the `IGeoObj` in general. This makes them independent from the internal object representation.

Currently, we support binary trees for uniform spatial subdivision. These trees work similar to interval octrees. The nodes represent axis-aligned boxes. Three different node types are supported. Black nodes are areas which are certainly filled by the object, white nodes are empty areas. Gray nodes represent uncertain areas: It is unknown whether a part of the object lies here. Usually the decomposition stops if the desired accuracy or a maximum subdivision depth is reached.

In general, the accuracy of the decomposition depends on the area covered by gray leaf nodes. We improved the accuracy significantly using a new decomposition technique based on implicit linear interval estimations [10]. For medium subdivision depths (15,20), the average reduction of the uncertain area[1] ranged between 13% and 25%.

3.6 The `algorithms`-Package

The four previously described layers form the foundation for high level algorithms. We provide them in the `algorithms`-package. Currently, the focus lies on algorithms for verified distance computation. However, collision detection and path planning are also in the scope of our project because hierarchical structures can be used effectively for this tasks.

3.7 Application in PROREOP

As outlined in section 2, the THR procedure is of special interest in PROREOP. One of the tasks is to calculate the shortest distance between all points on the acetabulum and the femur model. Using the example below, we show how each subprocess in this task can be assigned a layer in our framework.

[1] The area of gray leaf nodes.

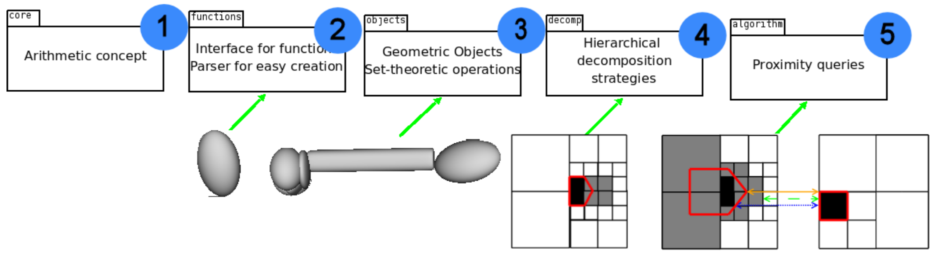

Fig. 6. Application of the framework in context of PROREOP

The SQ parameters resulting from the fitting process (section 2.2) can be used for modeling SQs using implicit functions and the `function`-package (cf. Fig. 6, item (2)). We can evaluate the resulting functions with every arithmetic type supported by the `core`-package (1). In the next step we combine several SQs employing set-theoretic operations to semi-analytic sets using the `object`-package (3). These more complex surfaces form a good model of the bones. Verified distance computation between such complex models can be very demanding. However, using the `decomp`-package it is possible to break complex objects down into much simpler parts (4). If we use an interval octree, we obtain axis-aligned boxes which form a verified enclosure of the original object. In the last step, we calculate a verified enclosure of the distance between the two original models (5). We have already outlined this algorithm in [8].

4 V&V Management and Tools

In the second part of the project TellHim&S, we focus on dynamic (mechanical) systems. We study the ways to verify their modeling and simulation on a computer appropriately and to offer techniques for their analysis (e.g. sensitivity) and validation. An important part of the project is the framework Vericomp [11] for comparing and assessing verified initial problem solvers for ordinary differential equations. However, Vericomp is not the main focus in this paper; our goal is to present those methods from TellHim&S which are useful in the biomechanical context.

There is a long tradition, for example, in the Computational Fluid Dynamics (CFD) community [12], of designing methodologies and of implementing and testing tools for the verification and validation assessment. The authors define the terms verification and validation in the context of modeling and simulation, software engineering and numerical mathematics. 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. One of the goals of TellHim&S is to enhance these schemes with our expertise from the field of guaranteed computations, beginning with the designing step.

There are three major steps in the verification and validation assessment process (V&V cycle [13]). The first step is to analyze the real world problem and to design a formal model of the system under consideration. 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. The third and final step, validation, addresses model fidelity, defines a validation metric and compares the outcomes of simulations and experiments.

In the project TellHim&S, various tasks in this cycle were treated from the verified point of view. Additionally, several decision-making aspects were formalized and applicability of the formalization shown using biomechanical applications (in particular, in PROREOP). This section gives a short overview of the results and points out new developments. We begin by giving a possible classification of processes with respect to their degree of verification. Next, we present a new library DSI Toolbox designed to define, aggregate and evaluate interval Dempster-Shafer structures in MATLAB. It addresses uncertainty management in case when definite bounds for parameters cannot be derived. Finally, we outline main features of SmartMOBILE, a library for verified modeling and simulation of mechanical systems, which offers verified methods on each stage of V&V cycle for systems with bounded uncertainties. Note that *verified* uncertainty management is addressed separately in Section 5.

4.1 V&V Classification

In [13], we introduced the following classification for processes, given from lowest to highest certification level.

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 to overcome uncertainties; 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.

An an example of how the developed systematics can be used in a biomechanical context, we consider accurate femur reconstruction using model-based segmentation and SQ shapes [14], a subtask in PROREOP. Computational model verification starts with an analysis of the collected patient data coming from medical examination in the gait lab and the radiology department. With the help of a specialized 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, allowing us to determine the level of V&V in the process according to the classification above [13].

According to [13], the process is 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 is used. The second step provides a region growing method to eliminate most tissues that are not bones. After region growing, the shaft of the femur is already sufficiently segmented, but the femur ball needs extra-processing. Therefore, in the third step, we use a VRML model of a standard femur to further refine the binary image. Then, a patient-specific superquadric bone model is built. From this model, significant points and quantities, like the mechanical length or the center of the femur head, can 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 delivers three independent computations of the patient-specific bone features and justifies the implementation to belong into class three of the V&V taxonomy. 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. However, it is necessary to take into account artifacts induced by skin motion that directly influences the position of markers with respect to the bones and joints during the experiments in the gait lab. In subsection 5.1, we show how this initial uncertainty can be propagated through the system to quantify the uncertainty in the outcome.

4.2 DSI-Toolbox — a Toolbox for Dempster-Shafer Analysis

The significance of the Dempster-Shafer theory [15] for modeling and propagating uncertainty has grown recently [6]. It allows us to combine evidence from different experts or other sources and provides a measure of confidence that a given event occurs. A special feature of this theory is the possibility to characterize uncertainties arising because of the lack of knowledge as discrete probability assignments. Due to the presence of imprecision, it is only possible to compute a lower and an upper bound (belief and plausibility) of the probability. However, the few existing DST implementations, for example, the IPP Toolbox [6], rely on floating point arithmetic and do not exploit to the full extent the inherently interval-based nature of the theory. With IPP as a basis, we developed a new verified implementation called DSI Toolbox (Dempster Shafer with intervals) for Matlab to work with rigorous DST structures that rely on interval calculus and directed rounding [5]. DSI uses INTLAB [7] for basic interval functionalities. It contains both functions from the IPP Toolbox, which were rewritten to take into account all rounding errors and adjusted to intervals, and newly designed functions.

The main task of the new toolbox is to guarantee correctness of the solution. For that purpose, we take care of all rounding errors that might occur during the computation by enclosing real numbers in their corresponding machine intervals. Note that we do not take into account the modeling error present in DST or other probability based methods. To optimize the CPU time, we compute all steps using vector-matrix operations in MATLAB/INTLAB.

With the help of DSI, we can define DST structures either directly by their focal elements with masses (routine `dsistruct`) or by cumulative distribution functions such as the triangular or the Weibull distribution (`dsitriangleinv`, `dsiweibullinv`). Given evidence can be aggregated by Dempster's rule or (weighted) mixing method, and the original and resulting belief and plausibility functions can be plotted.

As a short introduction to the DSI, consider the following example. Two experts give estimations about a robot failure. The first expert provides an assessment in form of a triangular distribution function. The important feature which the DSI Toolbox offers in this case is the possibility to define it with an uncertain mode and lower/upper bounds. In Fig. 7, left, the solution space of the triangular distribution with lower bound $[1, 2]$, upper bound $[10, 12]$, mode $[4, 6]$ and 2^{12} samples is shown. This space lies between the belief and the plausibility function computed using corresponding upper and lower bounds.

The second expert provides an assessment in form of a BPA directly. Using the DSI Toolbox, we define the BPA by the routine

```
dsistruct(
   [infsup(1,3),2/6;infsup(1.5,6),1/6;
   infsup(5,15),3/6]).
```

Here, `infsup(x,y)` is the standard INTLAB function to define an interval in infimum-supremum notation. In this example, the first focal element $[1, 3]$ has

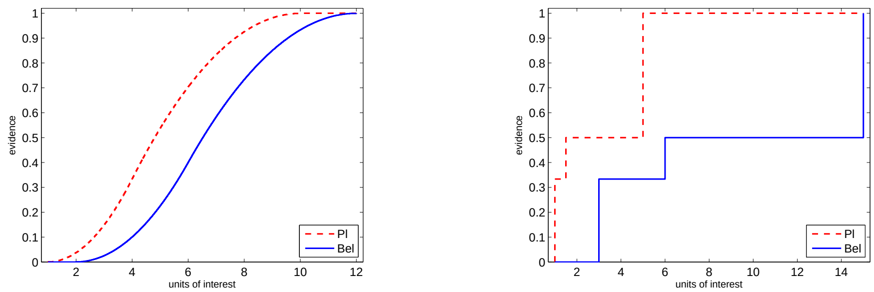

Fig. 7. Triangular distribution with lb=[1, 2], ub=[10, 12] and mode=[4, 6] (left) and the solution space for expert two

the mass 2/6, the second ([1.5, 6]) the mass 1/6 and the third ([5, 15]) the mass 3/6. In Fig. 7, right, the solution space of this BPA is shown.

To aggregate these two structures, we use Dempster's rule and mixing. In Fig. 8, the results of the application of Dempster's rule and unweighted mixing for the two BPAs from our example are shown. The BPAs and their aggregation by Dempster's rule are computed in 6.932 seconds on an Intel Core 2 DUO @ 2.1 GHz platform with 2 GB RAM. The overall CPU time for computing the BPAs and their unweighted mixing is 0.0925 seconds only (on the same platform).

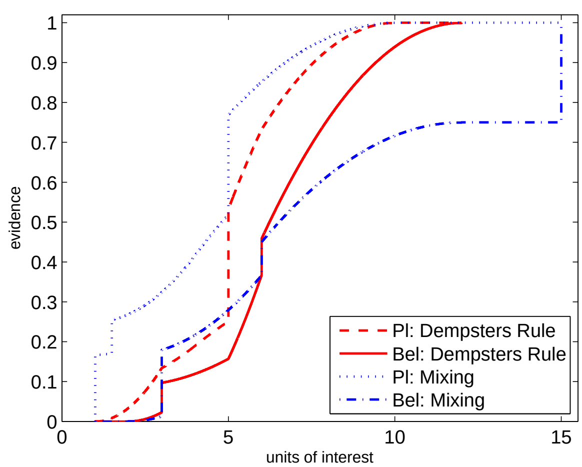

Fig. 8. Aggregation by Dempster's rule and unweighted mixing

In [5], we showed first results for the application of the DSI Toolbox in a practical context. It was possible to perform a simple fault tree analysis with DST structures as well as propagate DST based uncertainty through an artificial system with a non-monotonous system function. In both cases, DSI was faster than the floating point based IPP Toolbox from which it originated. Besides, we demonstrated that the correct results could be obtained more easily using DSI in the second case.

4.3 SmartMOBILE — A Tool for Verified Modeling and Simulation of Mechanical Systems

SmartMOBILE [4] is a C++ object-oriented software for verification of various classes of mechanical systems based on MOBILE [16] 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, scalar or spatial objects such as coordinate frames and 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 the 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 2. However, advanced users are not limited to them and are free to plug in their own implementations if they follow the general instructions from [17].

For most kinematical problems, it is sufficient to use the basic data type from Column 3 of the Table 2 as the parameter of all the template classes used for a particular model. The main idea for dynamical and special kinematical tasks such as finding of system equilibria is to use pairs basic data type/corresponding solver (Columns 3 and 4). 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 arithmetics should be mostly chosen for offline simulations with considerable uncertainty [4].

Aside from verified modeling and simulation, SmartMOBILE offers techniques for sensitivity analysis and uncertainty management described in detail in the next section.

5 Uncertainity Management

The experience of the last decades shows that while the design process in many application fields becomes shorter due to time-to-market pressure, the requirements on numerical accuracy and performance grow stricter. However, engineers lack precise knowledge regarding the process and its input data in early design

Table 2. Arithmetics supplied with SmartMOBILE.

Description	Arithmetic	Kinematic	Dynamics
reference	floating point	MoReal	MoRungeKutta,...
based on VNODE [18]	intervals	TMoInterval	TMoAWA
based on ValEncIA-IVP [19]	intervals	TMoFInterval	TMoValencia
based on RiOT [20]	Taylor	TMoTaylorModel	TMoRiOT
based on COSY [21]	Taylor	RDAInterval	—
equilibrium states	intervals	MoFInterval	MoIGradient
sensitivity with ValEncIA-IVP	intervals	MoSInterval	TMoValenciaS

stages. Therefore, to assess how reliable a system is, they have to deal with uncertainty. That is the reason why methods to propagate uncertainties through the system gain more and more importance.

The overall imprecision in the outcome can be specified by providing upper and lower bounds on all possible results using interval or other verified methods. As a further option, the Dempster-Shafer theory can be used as described in section 4.2. In this section, we use several subtasks from PROREOP to show how the uncertainty can be propagated in a verified way.

5.1 Identification of Body Segment Motion Using Marker Trajectories

We consider the problem of the reconstruction of the hip joint position from positions of markers fastened to specified places on a patient's leg, a task described in more detail in [22]. The corresponding model is purely kinematic. The data on marker positions contains measurement errors which appear, for example, due to skin movement during motion. These uncertainties are empirically proved to be within ± 10 mm for each the marker displacement tangential to skin and the one due to soft tissue movement. The marker displacement normal to skin can be up to ± 5 mm. Besides, both knee and ankle widths with nominal values of 120 mm and 80 mm, respectively, are also measured with an error of ± 10 mm.

In [13], we showed that the length of the femur was in the interval $[377.6; 396.7]$ if we considered only imprecisions in knee and ankle widths. Marker displacements caused an enclosure of almost 622 mm in diameter which is less meaningful in real life cases. This result can be interpreted in three ways. Fists interpretation is that all corresponding measurements have to be performed with great care if the proposed algorithm is to be used. The second is that an algorithm less sensitive to marker displacements has to be devised. The third presupposes that the overestimation in the verified algorithm we used is too big to characterize the problem properly.

The latter implication can be ruled out by taking into account a measure on overestimation provided by the reference uncertainty in this case:

$$[u] = \sum_{i=1}^{n} \frac{\partial f(p_1, \ldots, p_n)}{\partial p_i} \times [p_i] \ .$$

Here, the sensitivity $s = [s_1, \ldots, s_n]$ is a `double`-based value of the partial derivative of $f(p)$, $p = [p_1, \ldots, p_n]$. We computed s in SmartMOBILE using algorithmic differentiation and the floating point data type `F<double>`. Sensitivities for marker displacements have the highest absolute values which indicates that the large diameter of the enclosure is not entirely due to overestimation but really results from the high sensitivity of the original model to this kind of parameters.

5.2 A Simplified Muscle Activation Model

The model under investigation (Fig. 9, left) represents a simplified subsystem of the human leg described in more detail in [23], [24]. It consists of pelvis, thigh and shank. To drive the model in forward dynamics simulations, the muscle *biceps femoris short head* is included, which is responsible for knee flexion. For the purposes of the first verified study, the overall model is simplified so that it is everywhere continuously differentiable. For example, the force law of the involved muscle model is not Hill-type anymore because it contains non-smooth functions. Besides, the activation function is allowed to be negative as well as the force to be positive, which does not take into account mechanical constraints. Under these restrictions, the simulated results do not quite fit with the results obtained in the gait lab (cf. Fig. 9, right, for the knee angle).

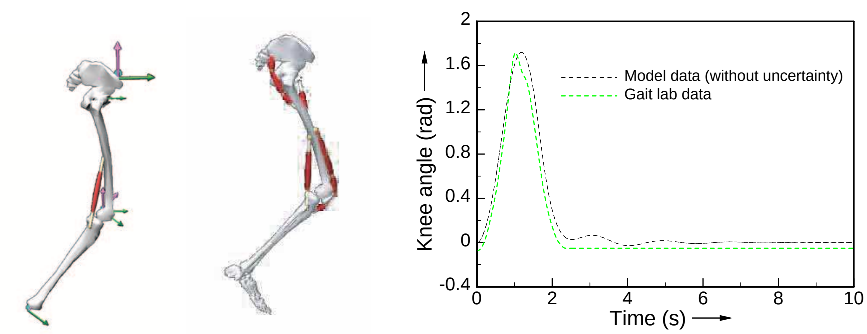

Fig. 9. The considered subsystem of the human leg: simplified model (left), original model (center), comparison with the data from the gait lab (right, without uncertainties)

Since most of the model parameters cannot be measured exactly, the task consists in investigating how uncertainty in parameters influences the outcome of simulations. The parameters of interest are the thigh length, the shank length, and the point of the muscle insertion at the hip (its z and y coordinates). In [24], [25], we identified thigh and shank lengths as the most influential parameters. To prove this, we computed the verified sensitivities of the solution with respect to all parameters of interest using the class `TMoValenciaSIntegrator` in SmartMOBILE. The results for the sensitivity of the knee angle are shown in Fig. 10. We notice again that the curves for the sensitivity with respect to thigh and shank lengths have a significantly greater absolute value in each point than those for muscle insertion and are also more prone to overestimation.

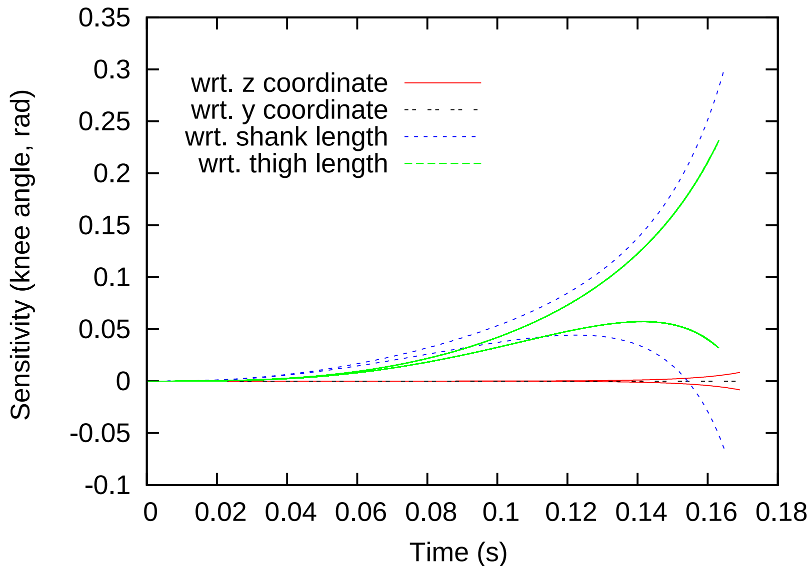

Fig. 10. Verified sensitivity of the knee angle to four uncertain parameters

The next step in verifying this model is to provide means of working with non-smooth functions. In this way we will be able to improve the correspondence of the simulation to the gait lab data. As a part of TellHim&S, we implemented a class `pwFunc` for computation of enclosures and first derivatives for piecewise functions defined in the following way (cf. Fig. 11, left):

$$\begin{cases} f_0(x), \text{ if } c_{-1} = -\infty < x \leq c_0, \\ f_1(x), \text{ if } c_0 < x \leq c_1, \\ \dots \quad \dots \\ f_{n-1}, \text{ if } c_{n-2} < x \leq c_{n-1}, \\ f_n, \quad \text{ if } c_{n-1} < x < c_n = +\infty \end{cases} .$$

For such functions, we define the first derivative as shown below.

$$\begin{cases} f_i'(X), & \text{if } X \subseteq (c_{i-1}, c_i], \\ \bigcup\limits_{k=i+1}^{j-1} f_k'([c_{k-1}, c_k]) \cup f_i'([\underline{x}, c_i]) \cup f_j'([c_{j-1}, \overline{x}]), & \text{if } X \subseteq (c_{i-1}, c_j] \end{cases} . \quad (4)$$

Note that the function $f(x)$ should be continuous in $x = c_i$, $i = 0, \ldots, n$. Besides, $f(x)$ is not differentiable in general, and $f'(X)$ for $X \ni c_i$ encloses both left and right derivatives if implemented as in Eq. (4).

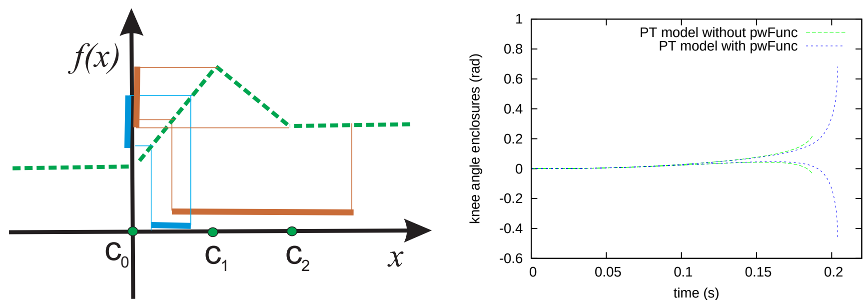

Fig. 11. Considered non-smooth functions (left) and simulation results for the enhanced simple model (right)

As a first test for this implementation, we enhanced the simplified muscle model by not allowing the activation function to be negative or greater then one and the force to be positive:

$$\begin{aligned}&0 \leq a(t) = A_1 e^{-c_1(t-t_1)} + A_2 e^{-c_2(t-t_2)} \leq 1\\ &F \leq 0 \ .\end{aligned}$$

We used our class in combination with the initial value problem solver `TMoValenciaIntegrator`.

The simulation results for the thigh length equal to 0.45 m with the uncertainty of 0.2% of the nominal value are shown in Fig. 11, right. Note that this uncertainty is consistent with the achievable measurement precision (cf. section 2.1). As expected, the use of the constrains improved the length of maximum simulation period (blue lines) in comparison to the previously used model (green lines).

6 Conclusions and Outlook

In this paper, we described the advances in the project TellHim&S funded by the German research council. In the geometry-related part of the project, we developed an integrated framework connecting all its aspects. It allows the user to employ different kinds of arithmetics, geometric primitives and decomposition strategies interchangeably in dependence on the application at hand. In the mechanics-related part of TellHim&S, we enhanced the V&V management by using verified instruments and developed a tool for characterizing epistemic uncertainty with the help of interval based Dempster-Shafer theory. We showed how our methods can be applied in a biomechanical context using subtasks from the project PROREOP. For the first time we performed a verified analysis of the dynamics of a mechanical system model which incorporated non-smooth functions.

Our future goals coincide with the uncovered aspects of TellHim&S. In the first part, we plan to extend the supported arithmetic types to Taylor models, integrate more input sources for objects and optimize existing algorithms for the new hierarchical structures. In the second part, we will concern ourselves with the refinement of the V&V assessment for biomechanical processes, the development of means for more complex fault tree analysis with Dempster-Shafer structures and intervals and the optimization of our implementation for non-smooth functions with respect to overestimation. In both parts, we will consider possible applications in biomechanics such as bone-prosthesis fitting or stance stabilization.

References

1. Lattanzi, R., Viceconti, M., Zannoni, C., Quadrani, P., Toni, A.: Hip-Op: an innovative software to plan total hip replacement surgery. Medical informatics and the internet in medicine **27**(2) (2002) 71–83
2. De Momi, E., Pavan, E., Motyl, B., Bandera, C., Frigo, C.: Hip joint anatomy virtual and stereolithographic reconstruction for preoperative planning of total hip replacement. In: International Congress Series. Volume 1281., Elsevier (2005) 708–712
3. Glassman, A.: Preoperative planning for primarytotal hip arthroplasty. Operative Techniques in Orthopaedics **5**(4) (1995) 296–305
4. Auer, E., Luther, W.: SmartMOBILE and its Applications to Guaranteed Modeling and Simulation of Mechanical Systems. In: Lecture Notes in Electrical Engineering. Volume 24. (2009)
5. Auer, E., Luther, W., Rebner, G., Limbourg, P.: A Verified MATLAB Toolbox for the Dempster-Shafer Theory. In: Proc. of the Workshop on the Theory of Belief Functions. (2010) Submitted.
6. Limbourg, P.: Imprecise Probabilities for Predicting Dependability of Mechatronic Systems in Early Design Stages. dissertation, University of Dusburg-Essen (November 2007)
7. Rump, S.: INTLAB - INTerval LABoratory. Developments in Reliable Computing (1999) 77–104
8. Cuypers, R., Kiel, S., Luther, W.: Automatic Femur Decomposition, Reconstruction and Refinement Using Superquadric Shapes. In: Proceedings of the IASTED International Conference. Volume 663. (2009) 59
9. Stauning, O., Bendtsen, C.: fadbad++ web page `http://www.fadbad.com/ (Accessed on 18.01.2010)`.
10. Bühler, K.: Implicit linear interval estimations. In: Proceedings of the 18th spring conference on Computer graphics, ACM (2002) 132
11. Auer, E., Rauh, A.: Toward Definition of Systematic Criteria for the Comparison of Verified Solvers for Initial Value Problems. In: Proc. of 8th Intl. Conference on Parallel Processing and Applied Mathematics PPAM 2009, Wroclaw, Poland (2009) In Print.

12. Oberkampf, W.L., Trucano, T.G., Hirsch, C.: Verification, validation, and predictive capability in computational engineering and physics. Technical Report SAND2003-3769, Sandia National Laboratories (2003)
13. Auer, E., Luther, W.: Numerical Verification Assessment in Computational Biomechanics. In: Numerical Validation in Current Hardware Architectures. (2008)
14. Cuypers, R., Tang, Z., Luther, W., Pauli, J.: A parametrized model for efficient and accurate femur reconstruction using model-based segmentation and superquadric shapes. In: I-ASTED, Baltimore (April 2008)
15. Dempster, A.: Upper and lower probabilities induced by a multivalued mapping. Annals of Math. Statistics **38** (1967) 325–339
16. Kecskeméthy, A.: Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen. PhD thesis, Gerhard Mercator Universität Duisburg (1993)
17. Auer, E.: SmartMOBILE: A Framework for Reliable Modeling and Simulation of Kinematics and Dynamcis of Mechanical Systems. PhD thesis, Universität Duisburg-Essen, WiKu Verlag Dr. Stein (2007) ISBN: 978-3-86553-240-4.
18. Nedialkov, N.S.: The design and implementation of an object-oriented validated ODE solver. Kluwer Academic Publishers (2002)
19. Auer, E., Rauh, A., Hofer, E., Luther, W.: 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 (2008)
20. Eble, I.: Über Taylor-Modelle. PhD thesis, Universität Karlsruhe (2007)
21. Berz, M., Makino, K.: COSY INFINITY 9.0. Programmer's manual. Technical Report MSUHEP 060803, Michigan State University (2006)
22. Tändl, M., Stark, T., Erol, N., Löer, F., Kecskeméthy, A.: An Object-Oriented Approach For Simulation Of Human Gait Motion Based On Motion Tracking. Special Issue of the International Journal of Applied Mathematics and Computer Science AMCS, "Verified Methods: Applications in Medicine and Engineering" **19**(3) (2009) 455–467
23. Strobach, D., Kecskeméthy, A., Steinwender, G., Zwick, B.: A Simplified Approach for Rough Identification of Muscle Activation Profiles via Optimization and Smooth Profile Patches. In: CD Proceedings of the International ECCOMAS Thematic Conference on Advances in Computational Multibody Dynamics, Madrid, Spain, ECCOMAS (June 21 – 24 2005)
24. Auer, E., Tändl, M., Strobach, D., Kecskeméthy, A.: Toward Validating a Simplified Muscle Activation Model in SmartMOBILE. In: CD Proceedings of SCAN 2006, IEEE Computer Society (2007)
25. Auer, E., Luther, W.: Uses of New Sensitivity and DAE Solving Methods in SmartMOBILE for Verified Analysis of Mechanical Systems. Special Issue of the International Journal of Applied Mathematics and Computer Science AMCS, "Verified Methods: Applications in Medicine and Engineering" **19**(3) (2009) 455–467