

# Integration of reliable algorithms into modeling software

W. Luther<sup>1</sup>, G. Haßlinger<sup>2</sup>, E. Auer<sup>1</sup>, E. Dyllong<sup>1</sup>, D. & H. Traczinski<sup>1</sup>

<sup>1</sup> University of Duisburg-Essen

Department of Computer Science

47048 Duisburg, Germany

{luther,auer,dyllong,traczinski}@informatik.uni-duisburg.de

<sup>2</sup> Deutsche Telekom, T-Systems

Am Kavalleriesand 3, 64307 Darmstadt, Germany

gerhard.hasslinger@telekom.de

**Abstract.** In this note we discuss strategies that would enhance modern modeling and simulation software (MSS) with reliable routines using validated data types, controlled rounding, algorithmic differentiation and interval equation or initial value problem solver. Several target systems are highlighted. In stochastic traffic modeling, the computation of workload distributions plays a prominent role since they influence the quality of service parameters. INoWaTIV is a workload analysis tool that uses two different techniques: the polynomial factorization approach and the Wiener-Hopf factorization to determine the workload distributions of GI/GI/1 and SMP/GI/1 service systems accurately. Two extensions of a multibody modeling and simulation software were developed to model kinematic and dynamic properties of multibody systems in a validated way. Furthermore, an interface was created that allows the computation of convex hulls and reliable lower bounds for the distances between subpaving-encoded objects constructed with SIVIA (Set Inverter Via Interval Analysis).

**Keywords.** Reliable algorithms, stochastic traffic modeling, multibody modeling tools, geometric modeling

## 1 Modeling software: three case studies

The techniques of validated computing have proved their merits in many scientific and engineering applications [13]. They are based on solid theoretical studies in mathematics and computer science [2]. In this paper we discuss different strategies to integrate reliable and validating algorithms into stochastic, mechanical or geometric modeling and simulation software.

### 1.1 Stochastic traffic modeling

Semi-Markov processes (SMP) provide a useful modeling approach for evaluating the performance of computer and telecommunication systems and can be effectively applied in many other fields as well. A SMP combines a Markov chain for transitions

in a finite state space and a state-specific distribution that characterizes possible increase and decrease of the workload in the system between certain points in time.

The computation of workload distributions of service systems in telecommunication networks is essential for determining the quality of service parameters for various types of data transfer traffic. Depending on the model, there are different ways to determine workload distributions. We investigate polynomial and Wiener-Hopf factorization as efficient approaches for classical general independent GI/GI/1 and discrete time semi-Markovian server systems. We show that subproblems can be non-trivially solved by a computer program. Both methods are integrated into a recent tool INoWaTIV (Interval-based numerical software with result verification of the workload distribution for Internet traffic). Our approach is first to obtain a numerical solution of the steady state workload distribution using an extension of the algorithm of Grassmann and Jain [12]. After a verification step, the guaranteed workload distribution can be computed. Both steps are carried out by using INoWaTIV.

## 1.2 Modeling of multibody systems

One of the most challenging application fields for reliable computing is the modeling of a robot in a complex environment. Numerical rounding, uncertainty in sensor measurements, and discretization contribute to significant error accumulation during non-validated computations. This happens, for example, while the collision positions of free arms and joint angles or the payload of the end-effector are computed. Similar problems exist in the context of robot localization and local path planning as well as in parameter estimation and identification of a non-linear controller for multibody systems with many degrees of freedom. For this reason we sought to introduce strategies that would enhance modern multibody modeling and simulation software (MMSS) with accurate routines that use validated data types, controlled rounding and algorithmic differentiation.

Particular attention was paid to the object-oriented MMSS MOBILE [14]. This tool was adapted to validation, which resulted in two reliable extensions: NiceMOBILE (Numerics with Intervals and Error Control in MOBILE) [2], [20] and SmartMOBILE (Simulation and Modeling of dynAmics in MOBILE: Reliable and Template-based) [3], [4]. The former is intended to support validated kinematics and can choose from two libraries for interval computations [22]. The latter incorporates libraries for algorithmic differentiation as well as the adapted interval initial value problem (IVP) solver VNODE [18] and can therefore be used for validated modeling of dynamic properties of multibody systems. The template-based structure of SmartMOBILE facilitates integration of arbitrary reliable data types instead of intervals, should the necessity to do so arise.

## 1.3 Geometric modeling

Along with the above-mentioned research, studies of reliable algorithms for computing distances between two objects were carried out. To cope with the lack of accuracy for elementary functions and rounding error accumulation inflicted by standard

IEEE 754 arithmetic, we first represented initial data for these objects as machine numbers. Furthermore, standard arithmetic operations and precise scalar product were preferred, and nested iterations avoided. Under these general design rules, a set of basic routines was developed. This set accurately computes the distance between points and edges or rectangles, as well as between (non)convex polygons. Verified enclosures of distances were obtained by using interval versions of the usual algorithms [6], [10]. Based on these routines, accurate algorithms that determine the distance between a sensor point and a polyhedron, two polyhedra and two axis-aligned or unaligned octrees [9] were developed. One of the results of this research was the C++ package PolyLib [19], which implements an interface between several of the above mentioned routines and MOBILE. The newly developed modeling tool ZMIN constructs and visualizes hierarchical tree structures and includes an interface to the SIVIA system [13]. Moreover, ZMIN interfaces with PolyLib and SIVIA, allowing us to accurately compute convex hulls and guaranteed lower bounds for distances of objects represented by subpavings.

## 2 Integration of accurate algorithms into MSS

Before discussing the integration of accurate algorithms into MSS, it is necessary to clarify general strategies for validating the results of modeling tools already in use. Fig. 1 illustrates the stages (represented by arrows) of the enhancement approach. An arrow's length represents the degree to which the validated software and MSS intertwine.

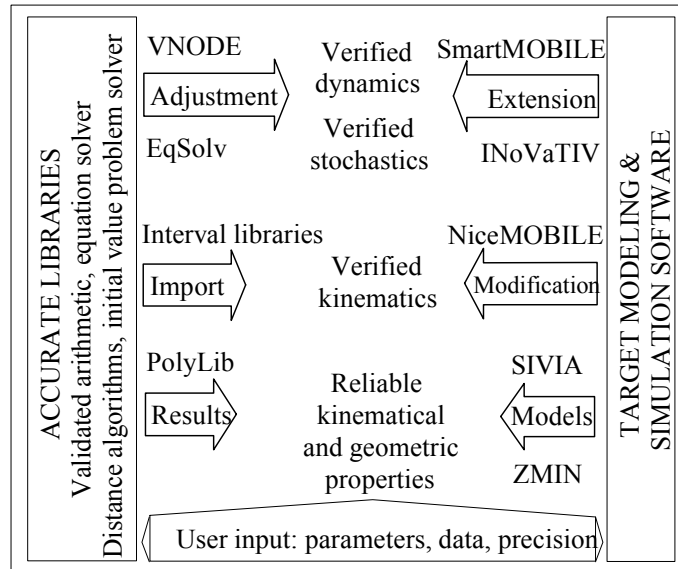


Fig. 1. Integration schemes

The first integration stage requires almost no merging (from either side). MSS and accurate algorithms exchange results, the former providing the required models and the latter the characteristics sought. In this manner, accurate kinematic properties can be obtained. This mechanism is used by PolyLib, for example, to compute collision-free trajectories for a moving end-effector or distances between polyhedral objects.

The second stage consists in introducing validated data types and algorithms into the MSS. This process requires substantial modifications in the target modeling software, causing a higher merging degree. Verified kinematic properties of the system being modeled can be now obtained, as in the case of NiceMOBILE, which uses an interval library and automatic error control [16] to modify MOBILE's code in order to provide guaranteed results.

The third stage presupposes that the second has been carried out. Now MSS has to be substantially extended or completely redesigned to feature validation. For example, it might become necessary to use enhanced modeling elements or templates that allow us to change basic data types and algorithms according to the task. In this case, not only MSS but also the validated algorithms have to be adjusted. After the third stage, both constituents form an almost inseparable whole, allowing us to obtain verified dynamic properties of systems. Examples of such programs are SmartMOBILE, which uses modified VNODE to compute solutions to the equations of motion resulting from an enhanced model, and INoWaTIV, which employs modified equation solver and interval matrix algorithms.

In the following sections, we present three case studies of modeling and simulation software for stochastic, mechanical and geometric systems.

### 3 Verified computation of the workload distribution

In [11] we have presented a short introduction to traffic modeling in telecommunication networks. The classical approach in queueing and service systems is to consider random variables for the interarrival times of events corresponding to the arrivals of packets, flows, connections or other units relevant for network elements. Two basic characteristics of the stochastic behavior of traffic are the distribution function of considered random variables and the autocorrelation of the process. To model the distribution function of arriving data as well as the autocorrelation function, semi-Markov processes can be used. These processes extend the well-known Markov models with finite state space. Furthermore, they provide memoryless states with exponential or geometrical distribution, such that each state is associated with an arbitrary state specific distribution. For a process  $SMP(m)$  with  $m$  states in the underlying chain, the autocorrelation has the form of a superposition of  $m - 1$  geometrical terms including complex coefficients.  $SMP(m)$  models have been successfully used within a fitting procedure for a given video trace to handle video traffic multiplexing as described in [21]. Since the results have influence on service level agreements, the analysis of such models and the modeling itself should be done as exactly as possible. Consequently, interval arithmetic is applied to guarantee the results of the analysis and to validate the method. Hence, we are able to obtain reliable information about data delay and loss probabilities.

### 3.1 Polynomial factorization approach

In the following, we consider a GI/GI/1 service system in the Kendall notation with general independent (GI) arrival and service processes and only one service station.  $A_n, S_n$  denote the interarrival and service times of the  $n$ -th arrival,

$$0 \leq A_n \leq g, 0 \leq S_n \leq h, U_n = S_n - A_n, W_n = \max(W_{n-1} + U_{n-1}, 0),$$

$W_n$  the waiting time of the  $n$ -th request. Then the fact that

$$Pr(W_n = i) = \sum_k Pr(W_{n-1} = k) Pr(U_{n-1} = i - k), i > 0,$$

leads to  $w(i) := \lim_{n \rightarrow \infty} Pr(W_n = i)$ , the stationary distribution. The workload of a GI/GI/1 server denotes the time required to process all requests present.

For the stationary distribution  $w$  we obtain the characteristic system equation [21]

- $w(k) = \sum_{-h \leq i \leq g} w(k+i)u(-i)$  for  $k \geq h$
- $w(k) = \sum_{-k \leq i \leq g} w(k+i)u(-i)$  for  $h-1 \geq k \geq 1$
- $w(0) = \sum_{0 \leq i \leq g} w(i) \sum_{i \leq j \leq g} u(-j)$ .

Using the representation  $w(k) = \alpha \beta^k$  we can derive the characteristic function as

$$p(z) = S(z^{-1}) \cdot A(z) - 1 = U(z^{-1}) - 1$$

with the generating functions  $A(z)$  and  $S(z)$  of the interarrival and service time

- $U(\beta^{-1}) = \sum_{-g \leq i \leq h} u(i)\beta^{-i} = 1, E(U) = \sum_{-g \leq i \leq h} i u(i) = -\sum_{-h \leq i \leq g} i u(-i) < 0,$
- $w(k) = \sum_{1 \leq j \leq h} \alpha_j \beta_j^k$

for simple zeros  $\beta_1, \dots, \beta_h$  of  $z^h p(z)$  inside the unit circle.

Since the degree of the characteristic system equation is  $g+h$ , we need the roots exclusively inside the unit circle. Now we use the following algorithm:

- Input: characteristic polynomial
- Determine the zeros  $\beta_j, j = 1, \dots, h$ , inside the unit circle. All roots except zero have to be simple
- Solve the Vandermonde system for  $\alpha_j$  using the  $\beta_j$
- Compute enclosures for  $w(0)$  and  $w(k), k > 0$
- Output: verified workload  $w$ .

This algorithm has the following problems: For huge values  $g$  it is difficult to localize  $h$  zeros in the unit circle and to find tight enclosures for the zeros of the large system. Only tight enclosures lead to a suitable solution of the Vandermonde system for the coefficients  $\alpha_j$  and to tight enclosures for the workload  $w(k)$ . So it seems promising to apply some theory to guarantee the existence of  $h$  zeros inside the unit circle and to compute initial values that are close to these zeros before starting a numerical solver.

In queueing theory, Rouché's theorem is very important to prove ergodicity conditions or the existence of solutions for polynomial equations building the generating functions. Generally, the theorem is used to show the existence of a certain number of zeros in the analyticity domain of a complex function. However, in certain cases it is very difficult to verify the assumptions of the theorem. Thus, it is of general interest to provide an alternative approach to obtaining the zero distributions of the system by using numerical methods or properties of the underlying stochastic process (i.e., the Markov chain).

In the GI/GI/1 case, it can be shown with the extended Rouché theorem [15] that exactly  $h$  roots are inside the unit disc.

**Klimenok's Theorem.** Let the functions  $f(z)$  and  $\varphi(z)$  be analytic in the open disk  $|z| < 1$  and continuous on the boundary  $|z| = 1$ , and the following relations hold:

$$|f(z)|_{|z|=1, z \neq 1} > |\varphi(z)|_{|z|=1, z \neq 1}, \quad f(1) = -\varphi(1) \neq 0.$$

Also, let the functions  $f(z)$  and  $\varphi(z)$  have derivatives at the point  $z = 1$  and assume that the following inequality holds:  $(f'(1) + \varphi'(1))/f(1) < 0$ . Then the numbers  $N_{f+\varphi}$  and  $N_f$  of zeros of the functions  $f(z) + \varphi(z)$  and  $f(z)$  in the domain  $|z| < 1$  are related as follows:  $N_{f+\varphi} = N_f$ .

In the present case we set  $\varphi(z) := z^h \sum_{-h \leq i \leq g} u(-i)z^i$ ,  $f(z) := -z^h$  and find  $(f'(1) + \varphi'(1))/f(1) = (h - \sum_{-h \leq i \leq g} u(-i)(h+i)) = -\sum_{-h \leq i \leq g} u(-i) < 0$ .

If we assume a uniformly distributed  $U$ , it is possible to derive the asymptotic behavior of the zeros  $\beta$ ,  $|\beta| < 1$ ,  $g \rightarrow \infty$ . Assume  $u(-i) = 1/c$ ,  $c := g+h+1$ . Then

$$cU(\beta^{-1}) = \beta^g + \dots + \beta^{-h} = c \rightarrow \beta^{g+1} - \beta^{-h} = c(\beta - 1) \Rightarrow$$

$$\beta^{-h} \approx c(1 - \beta), \quad \beta = r \exp(i\theta). \text{ Then it follows that}$$

$$cr^h = 1/(1+r^2-2r \cos \theta)^{1/2}, \quad r = c^{-1/h}(1+O(1/h)),$$

$$\arg \beta = 2\pi k/h + 1/h \arctan(r \sin \theta / (1 - r \cos \theta)).$$

Furthermore,

$$|(\beta^{h+g+1} - 1)/(\beta - 1)| \geq (r^{g+h+1} - 1)/(r + 1) := cr^h \Leftrightarrow 1 = r^h(c + cr + r^{g+1}) \Leftrightarrow$$

$$r = (c + cr + r^{g+1})^{-1/h} \Rightarrow r > (c + c^{1/h} + c^{-(g+1)/h})^{-1/h}, \text{ since } r < c^{-1/h},$$

which provides a lower bound for  $r$ . The asymptotic behavior can be used to generate suitable initial values automatically as an input for standard root finder tools.

**Theorem.** Assume that a closed expression for  $z^h S(z^{-1})A(z)$  can be obtained such that the asymptotic relation  $z^h S(z^{-1})A(z) \approx T(z, h)$  for  $g \rightarrow \infty$  and  $|z| < 1$  holds true. Then the roots  $w$  inside the unit circle fulfill  $w^h/T(w, h) \approx 1$ .

**Example 1.** In the equidistributed case we have

$$z^h S(z^{-1})A(z) = \frac{(1 - z^{h+1})(1 - z^{g+1})}{(g+1)(h+1)(1-z)^2}, \quad T(z, h) := \frac{1 - z^{h+1}}{(g+1)(h+1)(1-z)^2}.$$

Note that the degree of the numerator is substantially reduced.

Next, we want to discuss the more general SMP/GI/1 systems and a Markov chain with  $m$  states. We use  $p_m(z) = z^{mh} \det(U^T(z^{-1}) - I)$  to denote the characteristic polynomial of the SMP/GI/1 system and consider the following determinant function:

$$p_m(z) = \det \begin{pmatrix} \sum_{j=-h}^g u_{11}(-j)z^{j+h} - z^h & \sum_{j=-h}^g u_{12}(-j)z^{j+h} & \dots & \sum_{j=-h}^g u_{1m}(-j)z^{j+h} \\ \sum_{j=-h}^g u_{21}(-j)z^{j+h} & \sum_{j=-h}^g u_{22}(-j)z^{j+h} - z^h & \dots & \sum_{j=-h}^g u_{2m}(-j)z^{j+h} \\ \dots & \dots & \dots & \dots \\ \sum_{j=-h}^g u_{m1}(-j)z^{j+h} & \sum_{j=-h}^g u_{m2}(-j)z^{j+h} & \dots & \sum_{j=-h}^g u_{mm}(-j)z^{j+h} - z^h \end{pmatrix}^T$$

with  $u_{ij}(k) = \Pr(U_{n+1} = k, \sigma_{n+1} = j \mid \sigma_n = i)$  and  $\sum_{-g \leq k \leq h} \sum_{1 \leq j \leq m} u_{ij}(k) = 1$ .

As in the previous case, we apply Klimenok's theorem. This can be done successfully for  $m=2$  and 3, but for greater  $m$  it is only possible when further restrictive conditions are applied to the distribution  $\{u_{ij}(k)\}$ . To derive the asymptotic behavior of the zeros, we write

$$p_m(z) = \det \begin{pmatrix} a_{11}(z) - z^h & a_{12}(z) & \dots & a_{1m}(z) \\ a_{21}(z) & a_{22}(z) - z^h & \dots & a_{2m}(z) \\ \dots & \dots & \dots & \dots \\ a_{m1}(z) & a_{m2}(z) & \dots & a_{mm}(z) - z^h \end{pmatrix}.$$

For  $m=2$ , we infer  $|p_2(z)| \geq |1 - a_{11}| |1 - a_{22}| - a_{12}a_{21} = 0$  with  $a_{ii} = \max_{|z|=1} |a_{ii}(z)| = a_{ii}(1)$ ,  $i = 1, 2$ , and  $a_{12} = 1 - a_{11}$ ,  $a_{21} = 1 - a_{22}$ . Therefore, the assumptions in Klimenok's theorem are fulfilled since the first  $\geq$ -sign can be replaced by the  $>$ -sign if  $z \neq 1$ .

We put  $f(z) = (a_{11}(z) - z^h)(a_{22}(z) - z^h)$ ,  $\varphi(z) = -a_{12}(z)a_{21}(z)$  and find  $f(1) = (1 - a_{11})(1 - a_{22}) > 0$ ,

$$f'(1) = \left( \sum_{j=-h}^g (j+h)u_{11}(-j) - h \right) (a_{22} - 1) + \left( \sum_{j=-h}^g (j+h)u_{22}(-j) - h \right) (a_{11} - 1),$$

$$\varphi'(1) = - \left( \sum_{j=-h}^g (j+h)u_{12}(-j) \right) a_{21} - \left( \sum_{j=-h}^g (j+h)u_{21}(-j) \right) a_{12},$$

$$\begin{aligned} f'(1) + \varphi'(1) &= a_{21} \left( h - \sum_{j=-h}^g (j+h)(u_{11}(-j) + u_{12}(-j)) \right) \\ &\quad + a_{12} \left( h - \sum_{j=-h}^g (j+h)(u_{22}(-j) + u_{21}(-j)) \right) \\ &= -a_{21} \sum_{j=-h}^g j(u_{11}(-j) + u_{12}(-j)) - a_{12} \sum_{j=-h}^g j(u_{22}(-j) + u_{21}(-j)) < 0. \end{aligned}$$

The stochastic matrix  $P=(a_{ij}(1))_{m \times m}$  has  $m$  eigenvalues  $z_1, \dots, z_m$ . The greatest equals one, and all the others are located in the unit circle.

We define  $c\gamma(z) := (z^c - 1)/(z - 1)$  and evaluate the determinant in the case of equidistribution

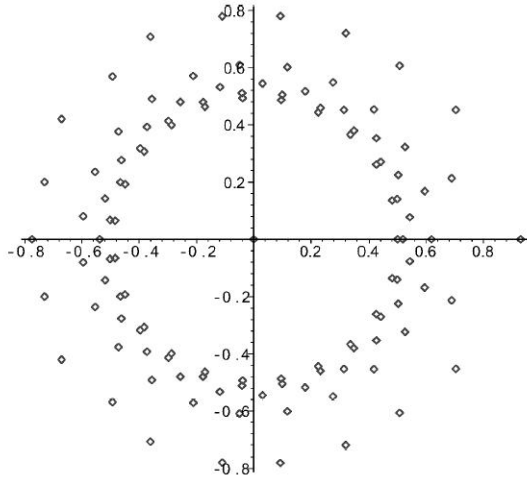
$$\begin{aligned}
p_m(z) &= \det \begin{pmatrix} a_{11}(z) - z^h & a_{12}(z) & \dots & a_{1m}(z) \\ a_{21}(z) & a_{22}(z) - z^h & \dots & a_{2m}(z) \\ \dots & \dots & \dots & \dots \\ a_{m1}(z) & a_{m2}(z) & \dots & a_{mm}(z) - z^h \end{pmatrix} \\
&= \det \begin{pmatrix} a_{11}\gamma(z) - z^h & a_{12}\gamma(z) & \dots & a_{1m}\gamma(z) \\ a_{21}\gamma(z) & a_{22}\gamma(z) - z^h & \dots & a_{2m}\gamma(z) \\ \dots & \dots & \dots & \dots \\ a_{m1}\gamma(z) & a_{m2}\gamma(z) & \dots & a_{mm}\gamma(z) - z^h \end{pmatrix} \\
&\approx (c - cz)^{-m} \det \begin{pmatrix} a_{11} - c(1-z)z^h & a_{12} & \dots & a_{1m} \\ a_{12} & a_{22} - c(1-z)z^h & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{1m} & a_{2m} & \dots & a_{mm} - c(1-z)z^h \end{pmatrix} \\
&= c^{-m}(1-z)^{-m} \prod_{i=1}^m (z_i - c(1-z)z^h), \quad c = g + h + 1, \quad g \rightarrow \infty, \quad |z| < 1.
\end{aligned}$$

Thus, the  $m$  relevant asymptotic relations as  $g$  tends to infinity are given by

$$c(1-z)z^h \approx z_i, \quad i = 1, \dots, m,$$

$z_i$  denoting the eigenvalues of the stochastic matrix. This fact can be used to generate the initial suggestion. After that, the roots of polynomials with complex interval coefficients are enclosed by an interval iteration. A verification step concludes the process.

Figure 2 shows the zero distribution for an SMP(5) adaptation in the case  $m = 5$ ,  $h = 23$ ,  $g = 64$ . We clearly discern five circles with a deformation near the point 1.



**Example 2.** Now we choose  $h = 25$ ,  $g = 500$ ,  $m = 2$ . The characteristic polynomial has degree  $2(g+h) = 1050$ . The corresponding asymptotic polynomial  $p(z)$  has 55 zeros, one zero outside, 4 zeros near 1 and 50 zeros inside the unit circle. The computed initial values have been successfully used to obtain tight enclosures.

**Fig. 2.** The 115 zeros of the characteristic polynomial with  $m = 5$ ,  $g = 64$  and  $h = 23$



As an example of how tight the enclosures computed this way are, consider the zero  $z = -0.424018874\ 2877889 + 0.501740168\ 5083331i$  and the corresponding enclosure  $[-0.4240188742877994, -0.424018874287782], [0.5017401685083224, 0.5017401685083437])$ .

With

$$H(z) = 0.04(1 - z^{26}) - 0.01(1 - z)(z^{18} + z^{17} + z^{16} + z^{15}) + 0.02(1 - z)(z^9 - z^8 + z^7 - z^6),$$

$$A(z) = 0.002(1 - z + z^2), B(z) = 0.001(1 + z)$$

the corresponding characteristic polynomial is

$$p_2(z) = \det \begin{pmatrix} 0.5H(z)A(z) - (1-z)^2 z^{25} & 0.5H(z)A(z) \\ 0.4H(z)B(z) & 0.6H(z)B(z) - (1-z)^2 z^{25} \end{pmatrix}$$

and has degree 55.

### 3.2 The Grassmann-Jain algorithm

Alternatively, we can apply the Grassmann-Jain algorithm to solve the SMP/GI/1 matrix case. The settings for the transition probabilities of the Markov process

$$u_{ij}(k) = \Pr(U_{n+1} = k, \sigma_{n+1} = j \mid \sigma_n = i); i, j \in \{1, \dots, m\}; \mathbf{u}(k) = (u_{ij}(k)), -g \leq k \leq h$$

$$U_n = A_n - S_n \text{ interarrival and service times of the } n\text{-th arrival}$$

$$\sigma_n \in \{1, \dots, m\} \text{ states of the underlying Markov chain}$$

$$l_{ij}(k) = \Pr(I = k, \sigma_E = j \mid \sigma_A = i) \text{ the probability for an idle period } I \text{ of duration } k$$

$$\sigma_A \text{ and } \sigma_E \text{ initial and final state of a given busy period}$$

imply the following relationships among the probabilities  $v_{ij}(k)$  and  $l_{ij}(k)$

$$\mathbf{v}(k) = \mathbf{u}(k) + \sum_{m=1}^{\min(h-k, g)} \mathbf{v}(k+m)\mathbf{l}(m), \quad k = 0, \dots, h; \quad (1)$$

$$\mathbf{l}(k) = \mathbf{u}(-k) + \sum_{m=0}^{\min(g-k, h)} \mathbf{v}(m)\mathbf{l}(m+k), \quad k = 1, \dots, g. \quad (2)$$

Here  $v_{ij}(k)$  denotes the probability that a phase with level  $k$  and initial state  $j$  is observed within a busy period having initial state  $\sigma_A = i$ .

Relation (1) defines a recursion for the probability interval matrices  $\mathbf{v}(k)$  with known  $\mathbf{u}$  and  $\mathbf{l}$ : ( $\mathbf{v}(h) = \mathbf{u}(h)$ ,  $\mathbf{v}(h-1) = \mathbf{u}(h-1) + \mathbf{v}(h)\mathbf{l}(1)$ , ...), the relation (2) represents an equation system for the probability interval matrices  $\mathbf{l}(k)$ ,  $k = 1, \dots, g$ .

In [11] the software environment is described that computes the Wiener-Hopf factorization numerically using the method developed originally for GI/GI/1 systems by Grassmann and Jain and now adapted to the SMP/GI/1 case. In order to solve equations (1) and (2), the iteration process introduced by Grassmann and Jain takes initial approximations for  $\mathbf{l}(k)$ . Thus, a sequence of interval matrix distributions  $\mathbf{l}^{(n)}(k)$ ,  $\mathbf{v}^{(n)}(k)$  for  $n = 0, 1, \dots$  is obtained. After this, a verification step is performed. If the condition  $[\mathbf{v}^{(n+1)}(k)] \subset [\mathbf{v}^{(n)}(k)]$  holds for all  $k = 0, \dots, h$ , Brouwer's fixed point theorem guarantees that the correct solution  $\mathbf{v}(k)$  is contained in  $[\mathbf{v}^{(n+1)}(k)]$  for all  $k = 0, \dots, h$ . With this verified result for  $\mathbf{v}(k)$  we compute a verified enclosure of  $\mathbf{l}(k)$ .

Consider an example which leads to a large system:

$$\begin{aligned} \Pr(A = 0) = 0, \Pr(A = i) = 0.00025 \text{ for } i = 1, \dots, 4000; \\ \Pr(S = 0) = 0, \Pr(S = i) = 0.0003125 \text{ for } i = 1, \dots, 3200. \end{aligned}$$

The following table shows the probabilities produced by polynomial and Wiener-Hopf factorization methods:

$w$	Polynomial Factorization ( $\cdot 10^{-4}$ )	Wiener-Hopf ( $\cdot 10^{-4}$ )
$w(0)$	3223.840[778932494,802745668]	3223.840790[807376,940053]
$w(10)$	1.684[161009480889,276354777528]	1.684218682[091688,205687]
$w(50)$	1.685[508983181590,602789470004]	1.685555886[256863,371390]
$w(99)$	1.686[696387027708,769329452040]	1.686732858[176097,291241]
$w(10^3)$	1.61790[5425768473,6315600827]	1.617905870[627717,748028]

Note that the Wiener-Hopf enclosures are tighter because the  $w(k)$  here are obtained directly.

The newly developed ‘Interval-based numerical software with result verification of the workload distribution for Internet traffic’ (INoWaTIV) (see Fig. 3) provides a unified approach via a graphic user interface to the verified computation of the workload distribution using both presented methods, the polynomial factorization and the Wiener-Hopf method for GI/GI/1 and SMP/GI/1 service systems. Based on the discussion presented in this section, the user can choose a method to determine initial values for the equation solver and – in a forthcoming version – directly influence the Grassmann-Jain iteration process. The results of the numerical analysis are visualized and parallelly stored in a text file. Thus the user has easy access to the distribution of the zeros of the characteristic polynomial as well as to the workload distribution, quantiles and mean values.

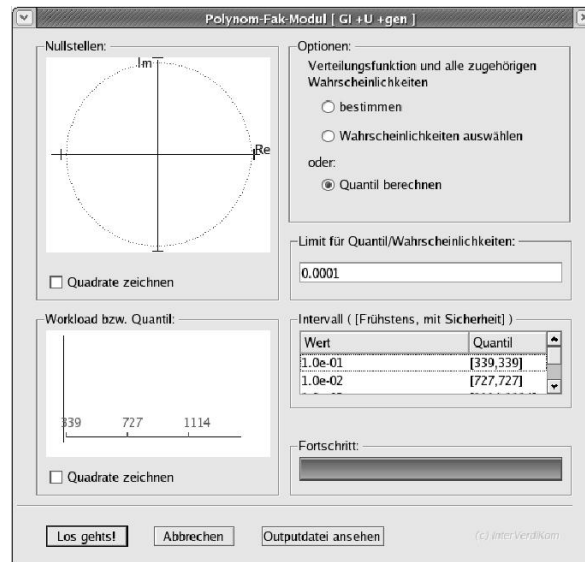


Fig. 3. Screenshot of the INoWaTIV-Tool

## 4 Multibody modeling and simulation software

Right from the beginning, the software in section 3 has been designed to work with intervals and thus provides guaranteed results. In contrast, the task of our second case study is to verify the efficacy of MOBILE, a tool that is already in use. To cover all the modeling facilities our target MSS offers, various validating techniques, such as accurate distance measuring and verified solving of systems of (non-)linear and differential equations, have to be integrated into MOBILE. The process of integration spans all stages of the scheme in section 2.

We have shown in detail in [5] how accurate distance algorithms (DA) can be imported into MOBILE. This integration corresponds mostly to the second stage of the scheme. We start with an accurate DA import by making the recently developed distance library PolyLib act as an element of MOBILE. Simultaneously, we can approach validation from a different angle by converting to intervals the existing DA from MOBILE. A naive replacement of IEEE data types with interval ones results in a considerable wrapping effect, especially if distances are intended to be used to model dynamics. An improvement can be achieved by changing the underlying data type (for example, from intervals to Taylor models), by modifying the existing modeling elements (for example, rotation error elimination strategy [20]), or by enhancing DA from MOBILE themselves. Moreover, if the dynamics of a system has to be modeled, the use of a better IVP solver might be helpful.

To validate DA from MOBILE, a complex element MoChordPointPolyeder, based on PolyLib, was introduced. This improves the modeling of the robot motion in intricate geometric environments. On the other hand, the standard DAs from MOBILE (the MoChord-based class branch) were reimplemented to produce guaranteed results and integrated into two verifying extensions of MOBILE – NiceMOBILE and SmartMOBILE.

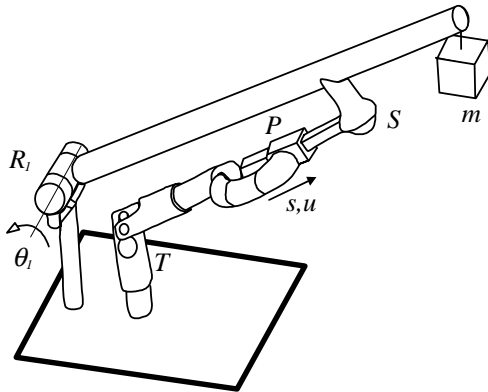
Let us consider SmartMOBILE in more detail. It was developed to model dynamic properties of multibody systems in a validated way. The programming concept of MOBILE presupposes, however, that such a tool can model kinematics as well. The objectives of the design of SmartMOBILE were, first, to develop a basic data type that would automatically compute derivatives of the solution of the motion equations and, second, to make this MOBILE version data type independent, allowing the user to apply his own solutions should the provided data type be unsuitable for the particular modeling task.

The first objective is met through implementing a composite data type TMOInterval and an interval IVP solver TMOAWAIntegrator (a transmission element based on VNODE). The data type helps to introduce algorithmic differentiation into MOBILE and therefore provides enough information for the solver to handle the initial value problems given in their algorithmic form.

The second goal is achieved by using the C++ template technique. This involves systematic replacement of all MOBILE classes with their equivalents containing *placeholders* instead of data types. Particular data types (e.g. TMOInterval) can be uniformly and easily substituted for these placeholders by the user during the last modeling stage.

At the present time SmartMOBILE is capable of modeling dynamic (and by implication kinematic) properties of various classes of multibody systems, including

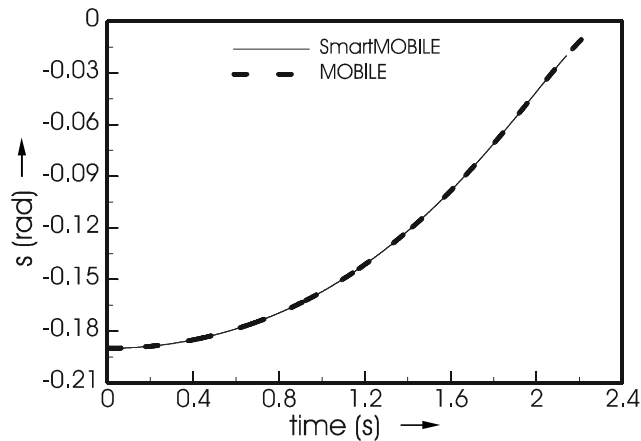
non-autonomous and explicitly solvable closed-loop systems. However, the computing time and wrapping effect in this version are not inconsiderable. Nevertheless, as the comparison with other approaches to MMSS validation and other IVP solvers shows, the tool seems to be more effective for (at least several) complicated multi-body systems, such as triple pendulum and one-axis manipulator, in regard to either computing time or global error (cf. [4]).



As an example of modeling in SmartMOBILE, consider the one-axis manipulator shown in Fig. 4. It consists of an extension arm driven by a linear actuator in combination with a four-joint chain. This chain includes a revolute ( $R_1$ ), a prismatic ( $P$ ), a universal ( $T$ ) and a spherical joint ( $S$ ) modeled with the help of separate elementary joints. A one kg mass  $m$  is attached to the arm. The system has one degree of freedom ( $s$ ) and, additionally, a control variable ( $u$ ) characterizing the linear actuator.

**Fig. 4.** One-axis manipulator

We do not have to change the given MOBILE model of this one-axis manipulator much to obtain the verified dynamics in SmartMOBILE. First, we choose a validated basic data type (e.g. the provided `TMoInterval`). Next, we use templates of the transmission elements with names similar to those used in MOBILE. Finally, after introducing the above-mentioned IVP solver `TMoAWAIntegrator`, we obtain the behavior of the variable  $s$  over the time interval (for the point initial conditions  $[-0.19;0]$  and  $u=60$ ).



**Fig. 5.** Position  $s$  of the one-axis manipulator

Fig. 5 shows the verified (solid line) and usual (dashed line) solutions. Note that the algorithm from SmartMOBILE produces the results only up to  $t=2.139$  and breaks down due to the wrapping effect after that point. However, this is not as discouraging a result as one might assume. If we obtain the expressions for the equations of motion of the one-axis manipulator with the help of an appropriate symbolic MMSS and then try to solve these equations with VNODE, it breaks down for  $t=0.582$ . Therefore, in this case SmartMOBILE can even be considered quite robust.

In Fig. 6, solutions obtained with SmartMOBILE, MOBILE and VNODE in relation to the midpoints of the intervals obtained with SmartMOBILE are shown. (The time interval  $[0,0.5]$  was chosen to represent all the results adequately.) We notice that the solution from MOBILE, which seemed to coincide with that of SmartMOBILE in Fig.5, lies considerably outside the validated bounds. In the scale of  $10^{-12}$  for the validated solutions, we can discern that the enclosures of both validated solutions intersect and are still quite tight. The overall similarity of results obtained with SmartMOBILE and VNODE shows that we cannot completely rely on the dynamic properties of this system as supplied by MOBILE.

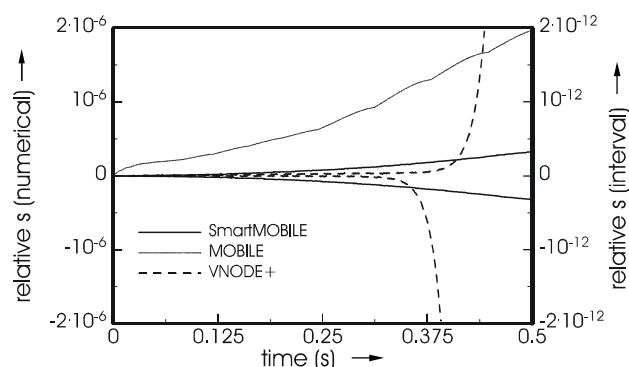


Fig. 6. Relative position of the one-axis manipulator

## 5 Reliable geometric modeling

Now we come to our last case study, which describes a way to plan reliable, collision-free paths by combining two geometric modeling and simulation systems with the distance library PolyLib.

A common technique for reconstructing a scene object is to use bounding volumes relying on an object's representation as a hierarchical model with axis-aligned bounding boxes. A typical example of a bounding volume is an octree. The main advantage of octrees is that the object representation does not depend on the nature of the real solid. This is a useful property for objects with complex structures that are difficult to describe using exact mathematical formulae. An adaptive enclosure of a real solid that depends only on the chosen maximum level of the tree is another advantage of the octree data structure. Because of their adaptive depth, octrees can be

used to describe a virtual environment constructed from camera data. For moving objects, it is interesting to employ unaligned octrees to avoid the wrapping effect during the octree decomposition in a common coordinate system after several rotations.

Quadtrees and octrees are well-known data types used to represent flat and spatial data hierarchically in a tree-based structure with four or eight childnodes. Alternatively, subpavings [13] – lists of disconnected boxes obtained by dividing the space along a plane – are employed. To construct the trees, a square or a cube including 2D- or 3D-objects is divided into four mutually disjoint subsquares or eight mutually disjoint voxels until the required closeness to the object is reached. Each node, square or cube is checked to see whether it is full (black), partially empty (gray) or empty (white) of solid material. If the nodes are empty or full, they do not need to be subdivided any further. In the case of partial emptiness, the nodes need to be subdivided to create a higher level quadtree or octree. The subdivision process is repeated until all the nodes are either full or empty or until the maximal resolution level has been reached. To obtain an outer or an inner hierarchically structured approximation of the object, partially occupied leaves are filled or emptied.

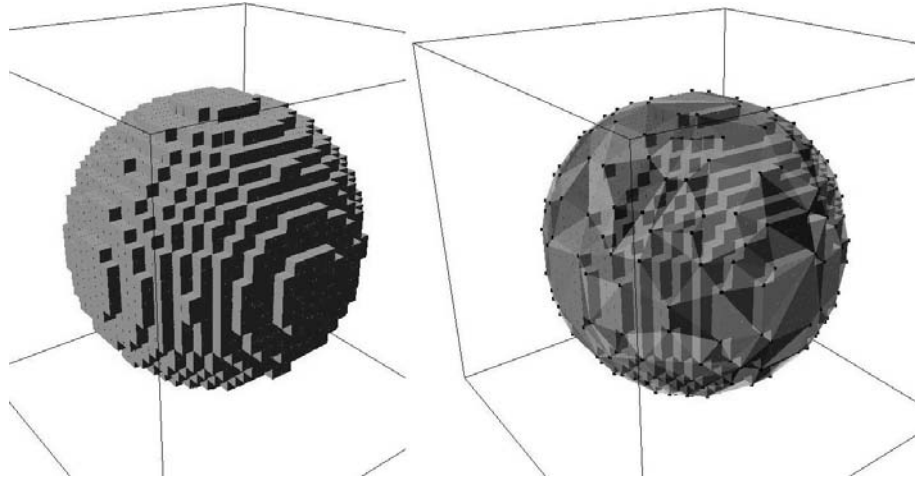
The distance calculation for octree-encoded objects is based on a simple computation of the distance between two boxes, which can be done with high accuracy. If the objects are represented in an axis-aligned octree, the vertices of the boxes are machine numbers by construction. This case is dealt with in [6]. The use of an appropriate local coordinate system reduces the case of two unaligned cubes to the case of an unaligned cube and an axis-aligned one. The distance and the corresponding couples of distance points between an axis-aligned and a rotated cube that both belong to a higher hierarchy level of the octrees can be computed by first applying mutual inclusion and intersection tests (see [8]). Then the distance computation can be reduced to a few typical cases: interval vertex to facet, interval vertex to vertex or edge, and interval edge to edge or vertex.

To cope with the unwanted rasterization effect and the multitude of corner points, a new algorithm for building the convex hull of an octree was presented in [10] (see Fig. 7). Using the convex hull of an octree allows one to apply algorithms for computing the distance between convex polyhedra and to speed up the distance calculation. Moreover, it helps to reduce the wrapping effect. To obtain lower bounds for the distance between two octrees, we can build the convex hulls of both and afterwards apply well-known distance algorithms for convex polyhedra. A feasible way of achieving the convex enclosure of an octree representing an object is to use the extreme vertices of the boxes forming the boundary. Then, the convex hull of all extreme vertices yields the enclosure of the object.

The case of a quadtree  $Q$  is treated in [10]. First, a rectangle hull of  $Q$  at level  $k$ ,  $k = 0, 1, \dots$ , is constructed, i.e. the smallest axis-aligned rectangles  $R_1, \dots, R_4$  that cover  $Q$  at level  $k$ , with the left-, right-, bottom- and topmost corners of  $Q$ .

For an octree, the preprocessing phase can be done analogously using a transformation of the octree into a bintree and updating the nodes which contain the possible extreme points of the convex hull at the level under inspection successively in  $x$ -,  $y$ - and  $z$ -direction. The preprocessing starts with the smallest axis-aligned cube covering the octree and finishes with a set of the extreme points. Instead of  $R_1, \dots, R_4$  we have eight regions, each with three extreme corners. To eliminate non-extreme

points, the well-known QuickHull algorithm can be used. Further details will appear elsewhere.



**Fig. 7.** Octree enclosure of a sphere and its convex hull

Both algorithms are implemented in a new tool ZMIN by Min Zhang [17] and described in his thesis. ZMIN also provides an interface to import the subpaving format defined by Jaulin *et al* [13] and used in SIVIA software. This allows us to build reliable geometric models by combining the geometric MSS SIVIA and ZMIN with the distance algorithms package PolyLib.

## 6 Conclusion

In this paper, the successful integration of various validating numerical methods into modeling and simulation software systems was presented. One achievement is the newly developed ‘Interval-based numerical software with result verification of the workload distribution for Internet traffic’ (INoWaTIV), which provides a unified approach to the verified computation of the workload distribution for GI/GI/1 and SMP/GI/1 service systems. The main advantage of this approach is the guaranteed accuracy of the results even when rigorous proofs are inaccessible, which is important for quality of service evaluation. Furthermore, it was shown how a multibody modeling and simulation system can be enhanced to model guaranteed kinematic and dynamic behavior, which presupposes solving linear and nonlinear equations or initial value problems with result verification. Finally, algorithms for computing the verified convex hull and a distance enclosure for objects represented by hierarchical geometric models were highlighted.

## 7 Acknowledgements

This research was carried out in two recent projects: ‘Traffic modeling in multi-service networks and analysis of resource requirements for Quality of Service support’ and ‘Accurate hierarchical structures in the construction of non-linear models with result verification’, both funded by the German Research Council (DFG).

## References

- [1] Adan, I. J. B., van Leeuwen, J. S. H., Winands, E. M. M.: On the application of Rouché’s theorem in queueing theory. Eindhoven University of technology (2005)
- [2] Alt, R., Frommer, A., Kearfott, R. B., Luther, W. (eds.): Numerical software with result verification. Dagstuhl-Seminar 19.01.-24.01.2003, LNCS 2991, Springer, Heidelberg Berlin, New York (2004)
- [3] Auer, E., Kecskeméthy, A., Tändl, M., Traczinski, H.: Interval Algorithms in Modeling of Multibody Systems. LNCS 2991, Springer, Heidelberg Berlin, New York (2004) 132-159
- [4] Auer, E.: Interval Modeling of Dynamics for Multibody Systems. SCAN 2004 (Fukuoka, Japan), to appear in Journal of Computational and Applied Mathematics
- [5] Auer, E., Dyllong, E., Luther, W., Stankovic, D., and Traczinski, H.: Integration of accurate distance algorithms into a modeling tool for multibody systems. IMACS, Paris (2005) Paper T2-I-73-0397
- [6] Bühler, K., Dyllong, E., Luther, W.: Reliable Distance and Intersection Computation Using Finite Precision Geometry. LNCS 2991, Springer, Heidelberg Berlin, New York (2004) 160-190
- [7] Cohen, J. W., Down, D. G.: On the role of Rouché’s Theorem in Queueing Analysis. CWI, Amsterdam (1995)
- [8] Dyllong, E., Luther, W.: Accurate Distance Algorithms for Octree-Encoded Objects. GAMM-Tagung, Dresden 2004, PAMM Vol 4 (1) (2004) 562-563
- [9] Dyllong, E.: Akkurate Abstands-Algorithmen mit Ergebnisverifikation, Dissertation, University of Duisburg-Essen, Reihe 20, Nr. 390, VDI-Verlag, Düsseldorf (2004)
- [10] Dyllong, E., Luther, W.: Verified Distance computation between octree-encoded objects. SCAN 2004 (Fukuoka), to appear in Journal of Computational and Applied Mathematics
- [11] Fausten, D. and Haßlinger, G.: Verified Numerical Analysis of the Performance of Switching Systems in Telecommunication. LNCS 2991, Springer, Heidelberg, Berlin, New York (2004) 206-225
- [12] Grassmann, W. K., and Jain, J. L.: Numerical solutions of the waiting time distribution and idle time distribution of the arithmetic GI/G/1 queue. Operations Research 37 (1989) 141–150
- [13] Jaulin, J., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis. Springer, London (2001)
- [14] Kecskeméthy, A.: MOBILE Version 1.3. User’s guide, University of Duisburg (1999)
- [15] Klimenok, V.: On the Modification of Rouché’s Theorem for the Queueing Theory Problems. Queueing Systems 38 (2001) 431–434
- [16] Krämer, W.: A Priori Worst Case Error Bounds for Floating-Point Computations. IEEE transactions on computers, 47(7) (1998) 750-756
- [17] Min Zhang: Konvexe Einschüsse von hierarchischen intervallbasierten Modellen. Diploma thesis, University Duisburg-Essen (2005)



- [18] Nedialkov, N. S.: The design and implementation of an object-oriented validated ODE solver. Kluwer Academic Publishers (2002)
- [19] Stankovic, D.: Implementation von hoch genauen Abstandsalgorithmen zwischen Polyedern zur Anbindung an MOBILE. Master 's thesis, University of Duisburg-Essen (2004)
- [20] Traczinski, H.: Integration von Algorithmen und Datentypen zur validierten Mehrkörpersimulation in MOBILE. Technical report, University of Duisburg-Essen (2005)
- [21] Traczinski, D., Haßlinger, G., Luther, W.: Polynomial factorization for servers with semi-markovian workload: performance and numerical aspects of a verified solution technique. *Stochastic Models*, Vol. 21, (2005) 643–668
- [22] Wolff von Gudenberg, J.: OOP and Interval Arithmetic – Language Support and Libraries. LNCS 2991, Springer, Heidelberg Berlin, New York (2004) 1–14  
<http://www-info2.informatik.uni-wuerzburg.de/staff/wvg/Public/>