Adaptive resolution in physics based virtual environments

M. Servin[*] Umeå University

C. Lacoursière[†] Umeå University

F. Nordfelth[‡] Algoryx Simulation

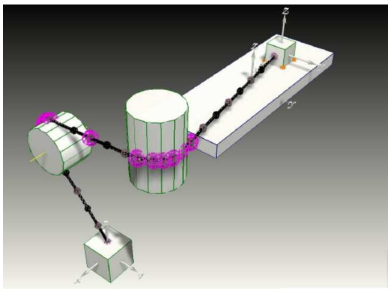

Figure 1: *Example from application including a wire with adaptive resolution. The system has two boxes connected by a wire routed over two cylindrical wheels that are free to rotate. Black dots shows lumped wire masses and magenta dots show contact nodes. Numerical robustness is secured by recognizing unstable states and simplifying the system to lower wire resolution for which the system is stable.*

Abstract

We propose a systematic approach to adaptive resolution in physics based virtual environments (VEs) that combines the conventional requirements of realtime performance, visual appearance with important requirements on the physical simulation, such as accuracy and numerical robustness. In particular, we argue that adaptive resolution is a key element to achieve robustness in fixed time-step VEs. The idea is to adaptively substitute unstable subsystems with more simplified and robust models. The method is demonstrated on systems including stiff wires. The algorithm brings stability, realtime performance and preservation of the important physical invariants to the system. The application to general systems is discussed.

Keywords: adaptive resolution, virtual environment, physics based animation, fixed time-step, numerical stability

1 Introduction

In physics based virtual environments (VEs), the state space of possible configurations is necessarily large and the trajectories depend critically on both user interaction and the dynamics of simulated objects. This is particularly true for training system where the state space cannot be restricted, i.e., catastrophic situations resulting from operating errors must be reachable in the simulation to avoid false training. This makes it difficult to construct VEs that are guaranteed to be robust, numerically stable over the entire state space, run in realtime, and offer optimal visual appearance and accuracy for the purposed use. The problem increases in complexity when taking account of different hardware set-ups with different performance, and view-dependent level-of-detail.

We propose a systematic approach to adaptive resolution in physics based VEs that combines the conventional requirements of realtime performance, visual appearance with important requirements on the physical simulation, such as accuracy and numerical robustness.

1.1 Related work

There is a vast litterature on *adaptive resolution* techniques, or *level-of-detail* (LOD) algorithms, in the context of 3D computer graphics and virtual environment since the early work by [Clark 1976]. The difficulties associated with applying LOD and culling algorithms in *physics based* virtual environments has attracted little scientific attention, but was adressed at least partially by [Carlson and Hodgins 1997] and [Chenney and Forsyth 1997]. Previous approaches have focused on either adapting the *spatial* resolution for the sake of visual quality or accuracy, e.g., [Spillmann and Teschner 2008] and [Redon et al. 2005], or on adapting the *time step-size* to avoid numerical instabilities in stiff systems [Debunne et al. 2001]. The latter approach conflicts with the realtime performance requirement and fixed time-step designs common in VEs. The former technique based on spatial refinement can introduce instabilities. We have found no example in litterature that combines the requirements of realtime performance, visual appearance with the important requirements on the physical simulation, such as accuracy and numerical robustness.

1.2 Our contribution

Our approach uses the fact that *the tendency for instabilities decreases as a system is made more coarse-grained.* For many systems these instability thresholds can be predicted in advance and incorporated in an adaptive resolution scheme. In particular, we apply this idea successfully in interactive simulations including inextensible but otherwise flexible wires—one of many stiff systems that plague VEs with numerical instabilities [Servin and Lacoursière 2008]. We present an algorithm for finding the optimal spatial resolution for the discretized wire objects at each time-step. The optimality is based on a line quality measure that includes weighted requirements for numerical stability, limited computational time and system granularity (linked to both visual appearance and solution accuracy). The algorithm brings stability, realtime performance and preservation of the important physical invariants to the system. Available computational resources can also be used to increase the resolution of specific parts of the VE, as prioritized by the user. Adaptive resolution has also proved to be a key element in finding robust contact methods for wires. We discuss how this works and show application examples of the method in the context of a maritime training simulator. Finally, we discuss the use of this method

[*] e-mail: martin.servin@physics.umu.se

[†] e-mail:claude@hpc2n.umu.se

[‡] e-mail:fredrik@algoryx.se

to more general systems than wires.

2 Adaptive resolution

The basis of physics based VEs is now outlined and a general algorithm for adaptive resolution is constructed.

2.1 Multibody system dynamics

Physics based VEs are examples of multibody system simulations (MBSS) which computes the motion of the system at discrete time-steps from a set of equations of motion. The computed motion depends on the initial conditions, the external and internal forces and various types of constraints, e.g., for modeling of joints, motors and dry frictional contacts. The computed motion also includes a numerical error that depends on the time step size and on the numerical solver being used. For practical reasons many VEs are run with fixed time-step, say, $h = 1/60\ s$. Dynamics on time scales shorter than h cannot be perceived visually by the user and is considered to be unimportant. Computational time should thus not be spent on resolving the high frequency dynamics unless it is necessary for the overall behavior. Using variable time-step or different time-steps for the different subsystems is possible but generally not practical in the realtime context, and thus finds little support in existing code libraries for rendering, collisions detection, or physics engines. Only fixed-step simulation is considered in what follows.

We use the descriptor form of multibody system dynamics. The variables for the system are generalized position q, velocity $\dot{q}$ and Lagrange multiplier λ for the kinematic constraint vector $g(q, \dot{q}, t) = 0$. For simplicity we assume the multibody system to be a particle system. The extension of the theoretical framework to include also rigid and deformable solid bodies is fairly straight forward. Given that the system has N_p particles and N_c scalar constraints the system variables have dimension $\dim(q) = \dim(\dot{q}) = 3 \times N_p$ and $\dim(\lambda) = \dim(g) = N_c$.

The system Lagrangian is

$$\mathcal{L}(q, \dot{q}, \lambda, t) = \tfrac{1}{2}\dot{q}^T M \dot{q} - V(q) - \lambda^T g \tag{1}$$

where M is the system mass matrix of dimension $3N_p \times 3N_p$, which is symmetric and positive definite and V is the potential energy of the system. The Euler-Lagrange equations derived from the least action principle are then

$$M\ddot{q} = -\nabla_q V + G^T \lambda \tag{2}$$
$$g(q, \dot{q}, t) = 0, \tag{3}$$

where $G \equiv \nabla_q g$ is the constraint Jacobian. We notate the system state at time t by $x(t) = (q, \dot{q}, \lambda)$.

The equations of motion (2) can be integrated using a variety of techniques. In particular, there is considerable choice for constraint satisfaction and stabilization, ranging from direct or iterative linear algebra techniques, penalty formulations, and constraint projection strategies. Our choice is the combination of a discrete-time variational technique described in [Kharevych et al. 2006], and constraint stabilization and regularization [Lacoursière 2007]. This yields a computational procedure to produce $x(t + h)$ given $x(t)$. The discrete-time variational integrators are derived from the least action principle rather than from discretization of the equations of motion. These steppers preserve many of the important physical invariants of the system by construction. This makes them generally more robust. At fixed time-step for instance, linear and angular momentum are preserved to machine precision, globally over the entire simulation. Examples of variational integrators are the Verlet stepper and symplectic Euler.

2.2 A general algorithm for adaptive resolution

Besides the time evolution of positions and velocities we now also consider the evolution of a variable number of bodies in the MBSS. We assume a resolution variable r that represents the state of resolution for the systems. This can be an integer, integer vector or more complicated set of integers depending on how each subsystem may be described at different levels. For each r the system has a specific number of particles and connectivity. We further assume a set of quality measures $Q(r)_\alpha > 0$, $\alpha = 1, 2, \ldots$, that each measures a specific quality of a specific subsystem. Good quality corresponds to values close to unity and poor quality corresponds to values close to zero. Next we list a number of qualities that can be included. *i) User perceived quality*, $Q_{\text{user}}(r, \texttt{view}, \texttt{inter})$. The subsystem which the user is focusing on should have high geometric level of detail and high functionality. User focus is determined through the camera view (`view`) and level of interaction with the subsystem (`inter`). *ii) Accuracy*, $Q_{\text{acc}}(r, x)$. This measures the accuracy of the numerical solution for a subsystem. Flexible systems, such as deformable solids or fluids, may require sub-division to maintain a specified accuracy when deformed. *iii) Computational time*, $Q_{\text{time}}(r, t_{\text{est}} t_{\text{lim}})$. During each time-step there is a fraction $t_{\text{lim}} < h$ of the step-size that may be spent on computing the dynamics. The actual time for computation for each subsystem for a given resolution r can be estimated to some number t_{est}. High quality is when $\sum t_{\text{est}} < t_{\text{lim}}$. *iv) Robustness*, $Q_{\text{rob}}(r, x, M, f)$. The risk of numerical instabilities—diverging or erratic changes in velocities or positions—can be estimated from the state x of the subsystem, possibly in combination with a model for the numerical stability of the subsystem at various levels of resolution. These models may typically also include system mass M and the forces in the system f. In section 3 we give examples of robustness quality measures for a specific class of system and we elaborate on the problem of estimating the numerical robustness in general systems further in section 4. As a cost function for keeping the application quality high we introduce $\frac{1}{2}\sum_\alpha w_\alpha Q(r)_\alpha^{-2}$ with weight coefficients $w_\alpha > 0$ for each quality.

The problem of evolving a VE with optimal quality can be treated as a problem of evolving the extended system with variables (x, r) and Lagrangian

$$\tilde{\mathcal{L}}(x, r, t) = \tilde{\mathcal{L}}(x, t) - \tfrac{1}{2}\sum_\alpha w_\alpha Q(r)_\alpha^{-2} \tag{4}$$

The resolution variable r can be treated either as a continuous variable and rounded to integer values or as an integer variable. The extended time stepping algorithm involves solving a complicated nonlinear equation, e.g., using Newton-iterations and taking into account the dependence in $Q(r)$ on $(q, \dot{q})$ and any implicit dependency in $(q, \dot{q})$ on r. This self-consistent formulation has the advantage that it has a variational formulation and may thus be approached with variational integrators like the rest of the system, i.e., treat the resolution parameters as any other dynamic variable and thus preserve the physical invariants of the system even at the events of change in resolution. It may however be too time consuming to perform Newton-iterations and solve these equations exactly. Instead, as a proof of concept, we assume a weak coupling between the variables $x = (q, \dot{q}, \lambda)$ and r and solve for them separately. An algorithm for MBSS including adaptive resolution for optimal quality in this form is given in Algorithm 1. Our approximation is to use the locally optimal solution for r', i.e., $r' = \arg\min_r(\frac{1}{2}\sum_\alpha w_\alpha Q(r)_\alpha^{-2})$. It is critical that the system reconfiguration is constructed to preserve the important physical invariants of the system, most importantly the total momentum. Step 9 in the algorithm involves also reconfiguration of the system connectivity, e.g., given by the constraints $g(x)$.

Algorithm 1 Adapt resolution for optimal quality

```
system initialization $x$ and $r$
while VE running do
    user interaction $\rightarrow$ $(\texttt{view}, \texttt{inter})$
    accumulate explicit forces $-\nabla_q V(x)$
    update contact data and constraint data $g(x)$
    step particles $x_{i-1} \rightarrow x_i$
    compute the quality function
    $Q(r) = (Q_{\text{user}}, Q_{\text{acc}}, Q_{\text{time}}, Q_{\text{rob}})$
    compute optimal system resolution $r'$
    reconfigure the system $(q_i, \dot{q}_i, r) \rightarrow (q'_i, \dot{q}'_i, r')$
end while
```

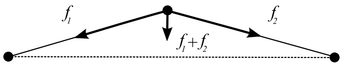

Figure 2: *The transversal force, $f_1 + f_2$, on the particles increases with the wire tension. Numerical instabilities develops when the wire tension and thus the oscillation frequencies become larger than the frequency of the numerical integration.*

3 Application to wires

In this section we describe physics based VEs containing wire systems with adaptive resolution for optimal quality. We model a wire by a set of constraints $0 = g^{\text{wire}} = (g_1^{\text{wire}} g_2^{\text{wire}}, \ldots, g_{N-1}^{\text{wire}})$ on a collection of $N \leq N_p$ particles, such that the particles are connected in line topology with pairwise distance constraints $0 = g_i^{wire} = |q_a - q_b| - l_i$ for maintained segment length i. The extension of this model to include stretch and bend elasticity and contact nodes is delayed to Sec. 3.3

Inextensible wires is an example of stiff systems that are well-known to be numerically unstable when the strain becomes too high. For instance, a wire with segment length of the order of $1\ m$, particle mass $1\ kg$ integrated with the symplectic Euler algorithm at step-size $h = 1/60s$ cannot support loads much larger than $100\ kg$ in normal gravity. This is a severe limitation in simulations where the wires and cables should be used for heavy hoisting or anchoring. The instability may also develop in the absence of heavy loads by the wire inertia of its own, e.g., in a whip effect. The cause of the numerical instability is that when the wire has normal modes of transversal vibrations with frequencies $\omega \geq h^{-1}$. These modes are excited by the numerical noise in the system. The normal mode frequencies are proportional to the wave velocity of transversal vibrations $c \equiv [fL/m_{\text{tot}}]^{1/2}$, where f is the wire tension, $m_{\text{tot}} = Nm$ is the total wire mass and wire length L. With increasing wire tension the transversal force component increases, see Fig. 2, and the normal mode oscillation frequencies increases along with it. The spectrum of normal modes for a wire of homogeneous distribution of N particles is discrete and the normal mode frequency ω_n ranges [Fetter and Walecka 1980]

$$\omega_n = \frac{2(N+1)c}{L} \sin\left[\frac{n\pi}{2(N+1)}\right] \quad , \; n = 1, 2, \ldots, N \tag{5}$$

The maximum frequency is $\max(\omega_n)$ can be approximated to $2(N+1)c/L$. A necessary condition on the number of point masses for the wire to be numerically stable can thus be formulated as

$$N < N_{\text{crit}} \equiv \frac{1}{2h}\sqrt{\frac{Lm_{\text{tot}}}{f}} \tag{6}$$

Observe that the extreme case with $N = 0$ is unconditionally stable. This is the *massless cable* in [Servin and Lacoursière 2007] and has no modes of transversal oscillations. If the wire would have been simulated with a time-explicit integrator and an explicit force model such as a spring forces the more restrictive *Courant condition* for numerical stability should have been included as well.

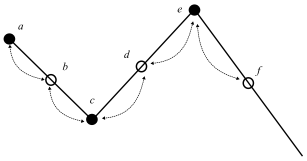

Figure 3: *Illustration of the* COARSE/REFINE*-transition for adaptive resolution of wires. In order to preserve total mass and momentum these are redistributed over the neighboring particles.*

3.1 Adaptive wire resolution

In order to realize an implementation of the adaptive resolution algorithm, Algorithm 1, we must decide on quality measures for the wire systems, define a wire resolution number and construct transition rules for wire refinement and coarsening.

Just for the clarity of the presentation here we assume spatially homogeneous distribution of wire masses and let the number of masses depend on a *resolution number* $r = 0, 1, 2, 3, \ldots$, as $N = 2^r + 1$. Each *refinement*, $r \rightarrow r_+ = r + 1$, and coarsening, $r \rightarrow r_- = r - 1$, means the number of wire segments are doubled or halved, respectively, as depicted in Figure. 3. The transition to another level of resolution should preserve total wire mass, local center of mass, local wire rest length and the linear momentum. The following transition rules accomplishes that. The particle mass depends on the resolution number as $m_{\text{tot}}/(2^r + 1)$. New particles b are added at half distance along the wire segments linking existing particles a and c. After eliminating a particle b the new wire segment is the straight wire connecting the two neighboring particles a and c. A REFINE-transition transforms the velocities as $(\dot{q}_a, \dot{q}_c, \dot{q}_e, \ldots) \rightarrow (\dot{q}_{a+}, \dot{q}_{b+}, \dot{q}_{c+}, \dot{q}_{e+}, \dot{q}_{f+}, \ldots)$, where

$$\dot{q}_{a+} = \dot{q}_a \tag{7}$$

$$\dot{q}_{b+} = \left(\frac{2^{r+1}+1}{2^r+1} - 1\right)\left[\frac{1}{\eta_a}\dot{q}_a + \frac{1}{\eta_c}\dot{q}_c\right] \tag{8}$$

$$\dot{q}_{c+} = \dot{q}_c \tag{9}$$

where η_a and η_c are the number of neighbors (1 or 2) of particle a and b, respectively. A COARSE-transition transforms the velocities as $(\dot{q}_a, \dot{q}_b, \dot{q}_c, \dot{q}_d, \dot{q}_e, \dot{q}_f, \ldots \rightarrow (\dot{q}_{a-}, \dot{q}_{c-}, \dot{q}_{e-}, \ldots)$, where

$$\dot{q}_{c-} = \frac{2^{r-1}+1}{2^r+1}\left[\dot{q}_c + \tfrac{1}{2}\left(\dot{q}_b + \dot{q}_d\right)\right] \tag{10}$$

with the exception for particles at the end points of the wires, in which case the contribution from neighboring particles—the b-term or the d-term in Eq. (10)—vanishes. Coarsening of a curved wire may produce significant compression of wire segments and violation of preservation of the total wire length. Depending on how these potentially large constraint violations are treated this may cause large energy injections and give rise to jittery and instabilities. We avoid this by allowing wire compression through modification of the constraint to an inequality constraint $g^{wire} \geq 0$. It can be shown that these transitions preserve the total momentum of the wire.

We define the wire system quality measure to be $Q =$

$(Q_{\text{acc}}, Q_{\text{time}}, Q_{\text{rob}})$ with

$$Q_{\text{acc}} \equiv \left(\frac{N}{N_p}\right)^{\gamma_{\text{acc}}} \tag{11}$$

$$Q_{\text{time}} \equiv 1 - \frac{1}{1 + e^{\gamma_{\text{time}}(t_{\text{lim}} - t_{\text{est}})}} \tag{12}$$

$$Q_{\text{rob}} \equiv 1 - \frac{1}{1 + e^{\gamma_{\text{rob}}(N_{\text{crit}} - N)}} \tag{13}$$

We do not claim that this measure of quality is unique nor canonical, but it has the functional dependency required for adaptive resolution for optimal quality in fixed time-step realtime simulations. The γ-exponents control the sensitivity to variations in N. For simplicity we have left out Q_{user} from the measure. The effect of user view and interaction are instead incorporated by modifying the weight factor w_{acc}. In practice, you may need to regularize the quality measure Q by adding a small positive number to avoid division by zero when the cost function is computed.

The COARSE-REFINE-algorithm for adaptive resolution for optimal wire quality we use is given by Algorithm 2. The computation of the wire tension required for line 7 in the algorithm is computed from the wire constraint force $G^T\lambda$.

Algorithm 2 Adaptive wire resolution

```
 1: system initialization $(q, \dot{q}, g, r)$
 2: while VE running do
 3:     user interaction $\longrightarrow (w_{\text{acc}}, t_{\text{lim}})$
 4:     accumulate explicit forces $-\nabla_q V$
 5:     update contact data and constraint data $g$
 6:     step particles
 7:     compute quality measure $(Q_{\text{acc}}, Q_{\text{time}}, Q_{\text{rob}})$
 8:     compute optimal wire resolution
        $r' = \arg\min_r \frac{1}{2} \sum_\alpha w_\alpha Q(r)_\alpha^{-2}$
 9:     if $r' < r$ then
10:         while $r' < r$ do
11:             COARSE$\longrightarrow (q_-, \dot{q}_-, g_-, r - 1)$
12:         end while
13:     end if
14:     if $r' > r$ then
15:         while $r' > r$ do
16:             REFINE$\longrightarrow (q_+, \dot{q}_+, g_+, r + 1)$
17:         end while
18:     end if
19:     Set new particle mass $m = m_{\text{tot}}/(2^n + 1)$
20: end while
```

3.2 Numerical experiments

The implementation of the adaptive wire model is made using the SPOOK-stepper introduced in [Lacoursière 2007] implemented in MATLAB. We describe the implementation in professional software in Sec. 3.3

3.2.1 Accuracy, stability and realtime performance

In order to test and demonstrate the presented method we set up a system consisting of a box of mass 100 kg supported by a wire of total mass 1 kg and length 10 m. Gravity is set to 9.8 m/s^2 and the time step $h = 1/60$ s. This system is unstable at mass ratios $1/1000$ and above, i.e., for wire resolution at $\gtrsim$ 10 particles under the weight of the box. The system is unstable for even finer resolution when the wire tension peaks, say, after a dropping the box from above. We let the resolution number and number of wire masses range between $(r, N) \in (0, 2), (1, 3), (2, 5), (3, 9), (4, 17), (5, 33)$. The parameters of the quality functions are $w_{\text{acc}} = w_{\text{time}} = w_{\text{rob}} = 1$, and $\gamma_{\text{acc}} = 2$, $w_{\text{time}} = 100/h$, $\gamma_{\text{time}} = 10$. The simulation is run for 8 $seconds$ and the result is displayed in Fig. 4, showing a series of snapshots and the time evolution of the quality measures and the number of wire particles. It can be seen that the wire avoids instability by decreasing the resolution at the high tension phase at times $t = 0 - 0.5$ s, $t = 3.5 - 4.5$ s and $t = 7 - 8$ s, and maximizes the resolution at the turn points at times $t = 2$ s and $t = 6$ s. Between time $t =$ 5–7 s we have inserted a dip in the available computational time t_{lim} that makes the resolution algorithm to decrease the resolution to maintain realtime performance.

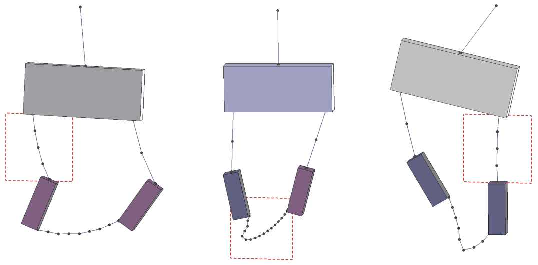

Figure 5: *Snapshots from simulation of several boxes and wires with view-dependent wire resolution. The view area is marked with red dashed wire and focuses at different wires at different times and thereby redistributes the available computational time to keep this wire at higher resolution if this is permitted by the stability requirement.*

The gain of the method can be understood from the following data from the numerical experiments. To have stable simulation at full wire resolution of the system and scenario described here would require a time step 20 times smaller than $h = 1/60$ s and consume increased computational power of *at least* the same factor. The alternative would be to keep the resolution at a level guaranteed to be stable at all times – in this case this would be at zero resolution ($r = 0$) – which would give poor accuracy and visual appearance. The computational overhead of the adaptive resolution algorithm for wires is small, roughly 1% of the total computational time spent on computing the dynamics

3.2.2 View-dependent adaptivity

We now extend the system of Sec. 3.2.1 to a system with four wires each of total mass 1 kg connecting three rigid boxes with masses $m_1 = 10$ kg and $m_2 = m_3 = 1$ kg. The estimated computational time is $t_{\text{est}} = \sum_{i=1,2,3,4} t_{\text{est},i}$. The view-dependency is introduced by increasing the accuracy weight factor $w_{\text{acc},i}$ of the wire "in view" by a factor 10^3. The adaptive wire resolution algorithm responds by giving the system "in view" higher resolution and more computational time, as long as this does not threaten the stability or realtime performance requirements. Snapshots from a simulation with view-dependent resolution are displayed in Fig. 5. The wire "in view" is the one marked by the red dashed box.

3.3 Implementation of adaptive wires in simulator software

Next we describe the implementation of adaptive wire resolution in a 3D physics simulation library called AgX [AgX] and list some important conclusions from this development. The AgX library was

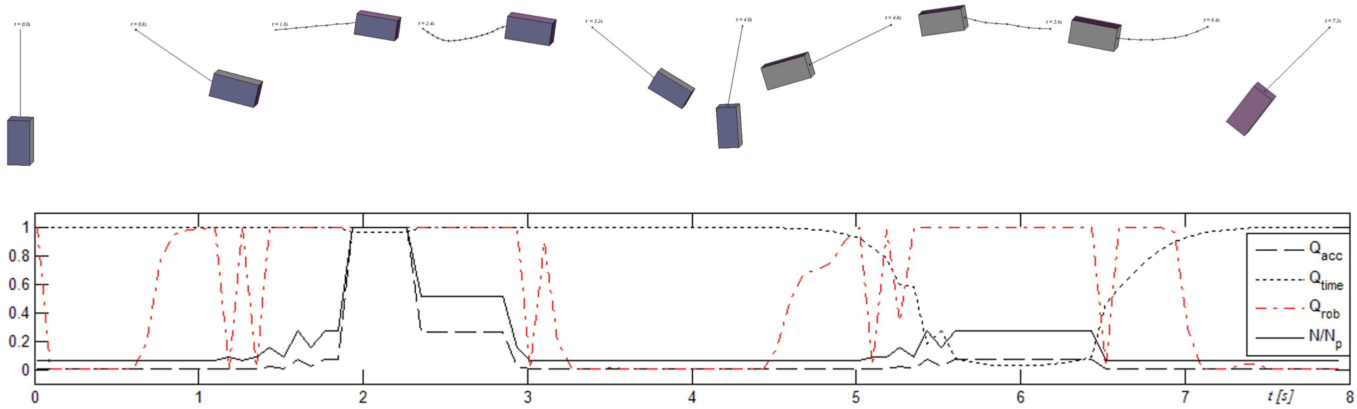

Figure 4: *The time evolution of a rigid box supported by a wire with adaptive resolution. The upper sequence of figures shows the system at different times – with the wire resolution varying with the stability criterion at high wire tension. The lower figure shows the wire quality measures and the number of wire masses (normalized by N_p) as functions of time. About the time $t = 6$ s there is a dip in the available computational time which forces the system to be coarsened.*

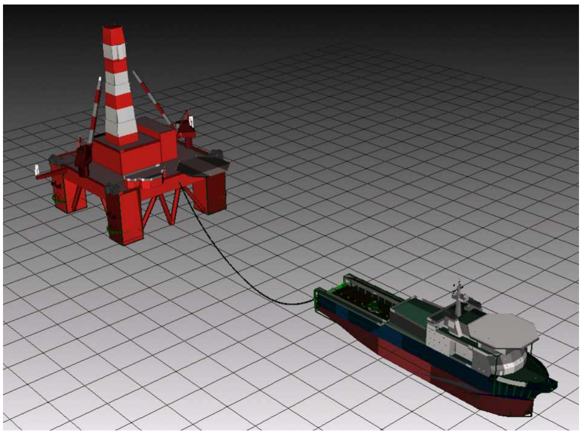

Figure 6: *Snapshot from a test-scene of the implementation of wire with adaptive resolution in a physics library used for training simulators. Wires occurs in ship handling oil rig anchoring. The wire routing can be complicated, connecting ship parts with oil rig, anchors and seabed. The wire length may vary up to several kilometers and sometimes be at very large tension.*

crafted for the realisation of professional simulators, e.g., training simulators for operation of heavy vehicles and ships. The implementation differs in several ways from the method presented in the present paper. Most importantly, it shares the strategy of securing numerical robustness by recognizing unstable states and simplifying the system to higher resolution for which the system is stable. One of the differences is that that it uses local resolution and inhomogeneous mass distribution instead of homogeneous and global wire resolution. The features of this wire model, which will be covered in more detail in a separate publication (in preparation), includes: *i) Slide nodes.* Besides attachment nodes at the ends of the wires, we include *massless slide nodes* as described in [Servin and Lacoursière 2007]. A rigid body attached to a wire with a slide node can slide along the wire like a “bead on a wire”. *ii) Frictional contact nodes.* Contact nodes are a natural extension of slide nodes. Instead of having body fixed position the contact nodes are continuously updated to be at the position on the body surface that minimizes the wire segment length. If the force at the contact node is directed “outwards” from the body the node is eliminated and the wire contact is thus detached. Stick friction in the wire direction is modeled by treating the contact node as an attachment node if the node force is within the friction cone. Sliding friction over the body surface is introduced with a friction parameter between zero and unity that diminishes the movement of the contact node by this factor. *iii) Wire elasticity.* Elasticity with respect to stretching and bending deformations of the wire is simulated following the method presented in [Servin and Lacoursière 2008] whereby established material models, e.g., for steel wire ropes and chains, can be used within the framework of constrained multibody systems.

In the development and implementation of this wire model we have found that adaptive resolution is a key element to make the wires numerically robust. The passage of wire masses over slide nodes and contact nodes is automatically handled by the resolution algorithm which eliminates the particles as they approach the node. Without this feature the local oscillation mode frequency would peak as the node and particle comes close together and the wire would become unstable. As a bonus, no special method to handle the passage of particles over nodes without risk of reflections needs to be implemented. Furthermore, handling the contacts by massless contact nodes rather than by light wire masses makes it possible to change resolution without introducing fluctuating contact forces and jittery particle behavior. The title figure, Fig. 1, shows an image from one of the test scenes. This system includes two boxes connected by a wire running over two rigid cylindrical wheels. Black dots shows lumped wire masses and magenta dots show massless contact nodes and their neigbouring wire mass nodes also involved in the friction model. Pulling a box makes the wheel turn by the wire-wheel friction force. Running the wheels with a motor makes them pull the wire and the boxes along with them. If the friction force is too small the wire will slide over the surface of the wheels. If the tension becomes large the system is simplified and with high enough tension no wire masses remains, only contact nodes and end attachment nodes. In Fig. 6 we show an image from a test-scene in the development of training simulators for anchor handling ship. These training scenarios involve multiple wires of length ranging from a few meters to several kilometers, complicated wire routings connecting various ship parts, other ships, oil rig and heavy anchors. The wire is connected to a drum and motor and has frictional contacts with the ship, in particular with guide pins and the stern roller at back of the ship.

4 Summary and discussion

We have presented a systematic approach to adaptive resolution in physics based virtual environments. The resolution adapts to maintain optimal quality with respect to a number of quality requirements imposed on the application, e.g., realtime performance, visual appearance, accuracy and numerical robustness. The latter two requirements are specific for physics based applications and are not included in conventional level-of-detail algorithms. We have presented a specific realization of this idea to multibody simulation involving wire systems and demonstrated how this improves the robustness, visual quality and time efficiency. Numerical experiments (Sec. 3.2.1) confirm that the overall computational performance can be improved by several orders in magnitude, as compared to resolving the instabilities by smaller integration time steps. We briefly described the implementation of this model in a software library used in commercial off-shore training simulators. An important conclusion from this development is adaptive resolution is a key element also to obtain robust simulation of wires with frictional contact nodes and slide nodes.

The computational power of personal computers is steadily increasing and seems to continue to do so with the emerging *massive multicore* CPU architecture. Adaptive resolution techniques is one important ingredient in order to gain the full potential of the increased computing power in VEs. Adaptive resolution of *physical systems* is particularly difficult. The most critical part is adaption into more coarse grained models in order to improve numerical robustness and diminish computational time. The general algorithm requires that the sub-systems of the VE are provided with graphical and physical models in a spectrum of granularity ranging from high to low. At lowest granularity the models should be simple model primitives, such as single rigid bodies or particles, that can be simulated extremely stable and fast. Complex objects important to the application, like a vehicle, can then be given a hierarchy of models ranging from a single primitive, to multiple connected rigid bodies and up to highly fine-grained models including deformable parts (tires, antenna, steel body, etc) and mechanical details (doors, suspension system, mechanical parts of the drive line, etc). This hierarchy of models should be assigned with unique resolution numbers, transition rules for refinement or coarse graining during simulation and quality measures with respect to numerical stability, computational time, accuracy and visual appearance. The adaptivity resolution algorithm will then automatically adapt the system to optimal resolution. If a subsystem approaches numerical instability or has its computational time decreased it will be substituted by a simplified system and more robust system. Much work remains to be done in this area, e.g., supply methods for predicting the numerical stability of complex systems at different resolution. Even though the more complex models are not as easily analyzed as wire systems they have normal modes of oscillations with some frequency spectrum depending on the resolution and occurrence of light elements and high tension. More coarse grained system has lower frequency spectrum and is more stable. Stability predictors can be constructed from known instability mechanisms. An alternative is to do precomputation of systems at different resolution levels and build prediction models in form of tables.

References

AGX. Agx multiphysics toolkit. `http://www.algoryx.se/`.

CARLSON, D. A., AND HODGINS, J. K. 1997. Simulation levels of detail for real-time animation. In *Proceedings of the conference on Graphics interface ’97*, Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 1–8.

CHENNEY, S., AND FORSYTH, D. 1997. View-dependent culling of dynamic systems in virtual environments. In *SI3D ’97: Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM, New York, NY, USA, 55–58.

CLARK, J. H. 1976. Hierarchical geometric models for visible surface algorithms. *Commun. ACM 19*, 10, 547–554.

DEBUNNE, G., DESBRUN, M., CANI, M.-P., AND BARR, A. H. 2001. Dynamic real-time deformations using space & time adaptive sampling. In *SIGGRAPH ’01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, E. Fiume, Ed., ACM SIGGRAPH, 31–36.

FETTER, A., AND WALECKA, J. D. 1980. *Theoretical Mechanics of Particles and Continua*. McGraw-Hill, 108–119.

GILLILAN, R. E., AND WILSON, K. R. 1992. Shadowing, rare events, and rubber bands - a variational Verlet algorithm for molecular-dynamics. *J. Chem. Phys. 97*, 3, 1757–1772.

KHAREVYCH, L., YANG, W., TONG, Y., KANSO, E., MARSDEN, J. E., SCHRÖDER, P., AND DESBRUN, M. 2006. Geometric, variational integrators for computer animation. In *SCA ’06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, ACM SIGGRAPH/Eurographics, 43–51.

LACOURSIÈRE, C. 2007. *Ghosts and Machines: Regularized Variational Methods for Interactive Simulations of Multibodies with Dry Frictional Contacts*. PhD thesis, Department of Computing Science, Umeå University, Sweden, SE-901 87, Umeå, Sweden.

REDON, S., GALOPPO, N., AND LIN, M. C. 2005. Adaptive dynamics of articulated bodies. ACM, New York, NY, USA, vol. 24, 936–945.

SERVIN, M., AND LACOURSIÈRE, C. 2007. Massless cable for real-time simulation. *Computer Graphics Forum 26*, 2, 172–184.

SERVIN, M., AND LACOURSIÈRE, C. 2008. Rigid body cable for virtual environments. *IEEE Transactions on Visualization 14*, 4, 783–796.

SPILLMANN, J., AND TESCHNER, M. 2008. An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum 27*, 2, 497–506.