Anders Backman, Kenneth Bodin, Claude Lacoursière, Martin Servin

Democratizing CAE with Interactive Multiphysics Simulation and Simulators

NAFEMS NORDIC Conference: Engineering Simulation: Best Practices, New Developments, Future Trends, 22 - 23 May 2012, Gothenburg, Sweden

Democratizing CAE with Interactive Multiphysics Simulation and Simulators

Anders Backman, Kenneth Bodin, Claude Lacoursière[*], Martin Servin[*]
Algoryx Simulation AB, Umeå, Sweden. Contact: contact@algoryx.se
[*] UMIT Research Lab, Umeå University, Sweden. Contact: claude@hpc2n.umu.se

Figure 1. Pictures illustrating the universality of the physics engine approach for e.g. robotics simulation; vehicle operator training simulators (Oryx Simulations); ship handling simulator (Kongsberg Maritime); optimization for granular systems and machines; and fluid simulation for visual effects.

1 Introduction to physics engines

1.1 History of physics engines

The concept of a „physics engine“ emerged in the mid to late 90's with initiatives from e.g. Havok, Ipion and Mathengine[1], addressing the computer games market. Initially, focus was on rigid multibody dynamics with contacts and constraints, but this has since extended into e.g. cloth, ropes, elasticity and fluids. The technologies were picked up from the multibody and robotics simulation domains, but physics this has also become an important field in computer graphics research. Physics engines are also relevant for the design of new parallel processors and compute languages. This is why e.g. Havok and PhysX now are owned and developed by Intel and Nvidia, respectively, and why the open source engine, Bullet, is developed in affiliation with AMD.

1.2 Physics engine technology

A physics engine is a software library for physics simulation with an application programming interface (API) that enables integration of the library with e.g. computer games, CAD programs, and virtual reality simulators. Simulations are built with declarative statements in the API for e.g. rigid bodies, constraints, motors and parameters. Contacts between bodies need not be specified explicitly, but are detected on the fly using geometric interference algorithms, and the system as a whole can often be fully reconfigured at runtime.
Most physics engines focus on real-time interactive performance. This requires relatively large timesteps so most engines use constraint based modeling and implicit integration methods. Implicity typically comes with the cost of solving a sparse linear complimentarity system (LCP) at each timestep. Often this still gives a performance advantage of 10-1000x for macroscopic systems and is well worth the trouble for stiff systems, and when using sparse parallel solvers. Physics engines that address the entertainment market generally favour performance over precision, and are not suitable for strongly coupled systems or force measurements, and so they have limited use in science and engineering. However, these limitations are eliminated in modern and more CAE oriented physics engines. Below, some of the most important recent advances are reviewed briefly.

1.2.1 Discrete variational integrators

Tradtionally, dynamic simulation is about time discretization of differential equations. A more modern approach is to model the system with a discrete Lagrangian and to derive stepping equations are from a discrete variational principle on the discrete action yielding discrete Euler-Lagrange equations[2]. This principle is not only beautiful, it also results in preserved symmetries and energy conservation. Recently, this approach has also been extended to deal with dissipation and mechanical constraints, where these constraints can be modeled using realistic material parameters[3].

1.2.2 Consistent constraint based multiphysics modeling

Constraints also apply to multiphysics modeling, e.g. of wires[4], elasticity[5], incompressible fluids[6], granular materials[7], electro mechanics and hydraulics (in progress), with great benefits in model consistency and time integration efficiency, compared to e.g. coupled co-simulation approaches.

1.2.3 High performance and heterogeneous sparse linear solvers

Non-stiff systems with small mass ratios can be solved using e.g. linear relaxation methods, at least for conceptual simulations. Stiff systems with large mass ratios benefit from recent advances in high performance sparse LCP solvers (see e.g. [3]). Heterogeneous systems benefit from splitting and a combination of solvers[8].

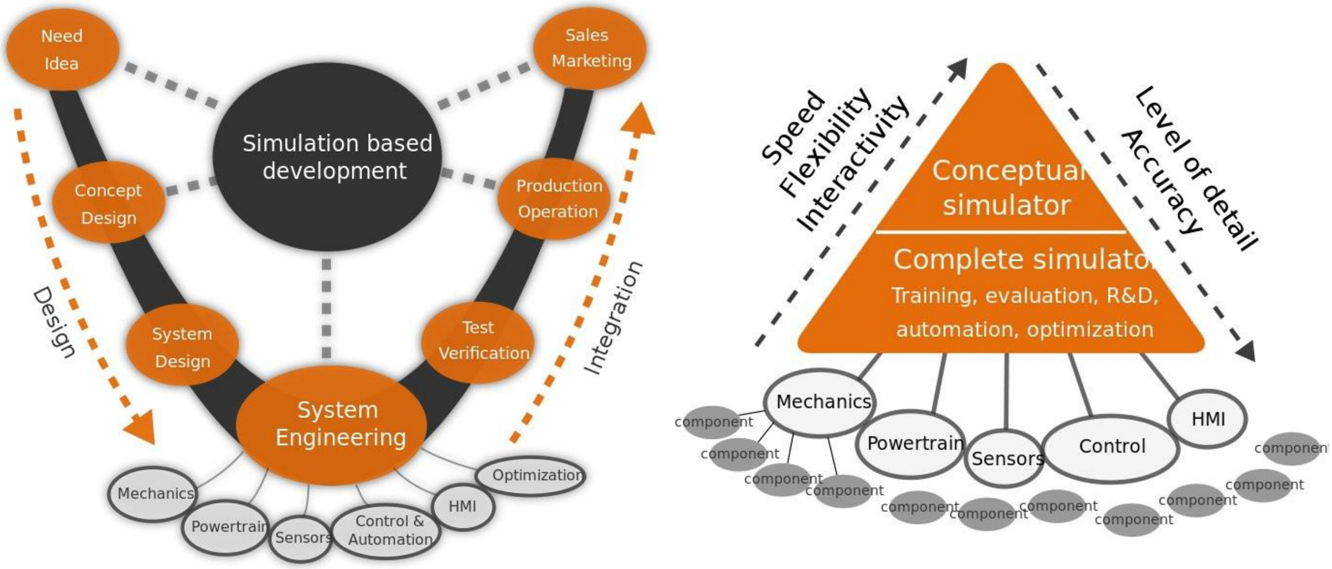

Figure 2. Conceptual illustration of convergence in simulation and simulator based product development.

2 Conclusion: democratization and convergence

Physics engines separate simulation modeling into: system declarations; physics models; numerical algorithms; and hardware, and therefore these can evolve independently but in synergy. *Democratization* occurs because of: formats for the different data types and models can be standardized openly for interoperability; and the declarative nature allows integration with easy-to-use editors accessible by non-simulation specialists. *Convergence* occurs because: a physics engine can be adapted to match wide range of requirements from design and conceptual simulation; engineering analysis and optimization; animation and marketing; to full system virtual reality simulators for virtual prototyping, marketing or training, allowing specialist to collaborate in new ways. Several such real-world examples will be given in the seminar, as well as a more thourough review of the physics engine anatomy and underlying theory. Fig. 1 shows some simulation examples, and Fig. 2 depicts the role of a physics engine as an enabling technology in the product development cycles.

3 References

[1] For an early review, see e.g. J. Lander and C. Hecker, *Physics Engines*, Game Developer Magazine, UBM TechWeb, USA, September (2000).

[2] J. Marsden and M. West. *Discrete mechanics and variational integrators*. Acta Numerica, pp. 357–514, (2001).

[3] C. Lacoursière, *Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts*. PhD thesis, Umeå University, (2007). http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-1143

[4] M. Servin, C. Lacoursière, K. Bodin, *Hybrid, multi-resolution wires with massless frictional contacts*, IEEE Transactions on Visualization and Computer Graphics, Vol:17 No:7, pp: 970 - 982, (2011)

[5] M. Servin, C. Lacoursière and N. Melin, *Interactive simulation of elastic deformable materials*, In Proceedings of SIGRAD Conference 2006 in Skövde, Sweden, Linköping University Electronic Press, Linköping, 22-32 (2006)

[6] K. Bodin, C. Lacoursière, M. Servin, *Constraint fluids*, IEEE Transactions on Visualization and Computer Graphics, Vol pp, Vol:18, No:3, (2012)

[7] M. Servin, C. Lacoursère, D. Wang, K. Bodin, *Examining the smooth and nonsmooth discrete element approaches to granular matter*, Particles 2011 – ECCOMAS.

[8] K. Bodin, C. Lacoursière, M. Nilsson and M. Servin, *Constraint Based Particle Fluids on GPGPU*, Particles 2011 – ECCOMAS.