Havok's ***Dynamical Systems***
A Non-Technical Overview

Jad Nohra

I Introduction

I.1 History

I initiated the Dynamical Systems 'team' within Havok after we suffered a number of failed MilSim projects. I realized that there was finally some alignment between the mathematical studies that I had been doing for several years independently of Havok, and the specialist skills necessary for successfully tackling the reasons of failure. A more detailed description of this is in the appendix. So far, only Dave Gargan and the Munich office are aware of this 'team', and it consists of only myself. I nevertheless call it a team because I could use some programming help, so it might become one when the time is right.

I have been and still am keeping it under wraps because it is quite a risky and long term research project. I want to protect it from product and sales related pressure until it is ready for it, which is not yet the case. Luckily, KMW has been a very good and patient partner in this context. They are nice to work with on a technical level, but more importantly, they are very relaxed about risks and timelines because they have experienced first-hand the failure of tackling this kind of simulation first by themselves and later with our 'game physics'.

I.2 Purpose

I.2.1 Robotics Roots

The birth of 'game physics' was possible due to two main factors. The performance improvements of PC hardware along with a number of influential papers by scientists from the Robotics field. Their goal was not to launch our business, but simply to adapt and fine tune certain kinds of simulations needed in Robotics to take advantage of the mentioned improvements. It was a lucky coincidence that some of the techniques involved were mathematically at a level reachable by non-specialists, while at the same time not so trivial to be programmable by almost anyone.

I.2.2 1996

The key academic papers were published around 1996, and it is therefore not a surprise that the spurt of real-time rigid body dynamics middleware followed: Ipion/Havok in 1998, Meqon in 1999, Open Dynamics Engine in 2000, Novodex in 2002, Tokamak in 2003, Karama/Vortex/MC Labs in 2001, OpenTissue in 2001, Bullet in 2003, along with others that came later. The scientists behind the driving academic papers pursued their academic interests, which were in general not directly related, and when they were, not as accessible as the 1996 ones. In short, the 'low hanging fruits' were gone.

As a result, most of the middleware companies that did and do not have scientists with the necessary mathematical background, a background not at all common amongst programmers, were stuck with the intuitive 1996 technology, along with variations and optimizations. This technology can be summarized by four tenants:

- Intuitive formulation of velocity constraints without knowledge of multi-variate calculus. This is done by simply calculating the velocity differences at certain points and dividing by mass to obtain corrective impulses.
- Iterative solving of the mixed LCP problem using the intuitive technique of sequentially applying the mentioned error impulses, clamping them when necessary. This is not identical, but almost equivalent to what is called 'Projected Gauss-Seidel'.
- Not updating positions from the updated velocities when integrating for stability. This is also called semi-implicit integration.
- An intuitive geometrically oriented implementation of the GJK algorithm. There are algebraic variants but these are less accessible.

I.2.3 The Gap

No improvements have been made, and none are possible without specialization. It is not true that this is due to lack of demand: everyone knows how limiting the stability and softness problems are. What is in fact true is that there is a mathematical gap. Companies that do have the scientists have long passed that gap, and this is usually directly attributable to key scientists (e.g Claude Lacoursière from CM Labs and now Algoryx).

What makes bridging the gap non-trivial is a certain conundrum associated with it: the people who have the knowledge (e.g the academics of 1996) could not care less and have completely different goals and ambitions and the people who do care are programmers with less than adequate background, a background that is not at all easily acquired.

The purpose of the Dynamical Systems team is to bridge this gap within Havok.

II Technological Deliverables

In the long term, the internal technological deliverables are solvers, integrators, pre-conditioners and tuning automation tools that will allow for as much fidelity as possible given the real-time constraints. We know this is possible simply because our competitors have achieved it.

- A rigid body dynamics solver, capable of handling multiple high fidelity industrial machines in real-time, including their contact interactions with static objects and with each other.
- A constrained dynamics massless cable solver, specialized for non-slacking cables under very heavy loads.
- A continuum dynamics cable solver, for general cables.
- A granular matter solver, specialized for the simulation of interaction between soil and vehicles.
- Specialized constraints when necessary.
- Components that can be used in special cases within 'game physics'.

The reason for these not being unified is that although the mathematical knowledge and some of the core technology is shared, the real-time simulation requirements dictate specialized solutions (e.g solvers) that take into account the specifics of the dynamical system to be simulated. Per example, a general cable has a much simpler geometrical structure than an industrial machine while on the other hand a great length of it has to be simulated. Similarly, a massless cable is simpler in structure than a general one but is subjected to vibrations of much higher frequencies. Specifics such as these dictate very different mathematical models and hence algorithms.

A kind of ***superdemo*** that I have in mind, highlighting a large part of the technology, is beautifully captured by the following image.

Figure 1: A complex industrial machine, stabilized by cables, working on a very step hill of slippery granular matter.

Finally, given the obvious effort and research involved, it is difficult to give useful timelines at this early stage, but this is alleviated by having KMW as a sort of pilot project partner and hopefully others as the right opportunities arise.

III Appendix

III.1 Mathematical Prerequisites

The mathematical prerequisites for performing well within the field of numerical dynamical systems simulation vary based on ambition. On the lowest level, any technical non-mathematical background is enough to read and implement the kinds of accessible results of 1996 but no more. Specifically, a familiarity with low dimensional geometry, high-school calculus and physics upgraded with rudimentary 3D vector algebra and willingness to delve into the specifics of rigid body dynamics, especially rotation. With these, one is able to use the tools provided by these papers but not understand them.

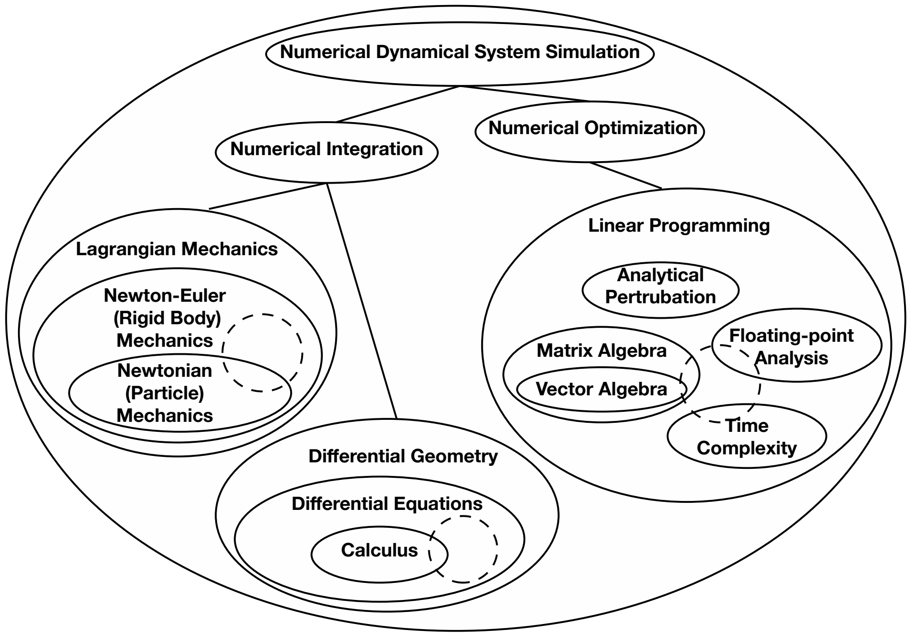

Figure 2: A very crude illustrative figure that highlights the ***gap***. If knowledge is measured by area, the area usually accessible to non-specialists is the one surrounded by the dotted circles, the total area being what is required.

At this level, one does not have the ability to follow the advances by reading the related literature and latest papers, understand the mathematical proofs that show that these techniques work, decipher the theories about less accessible competing techniques, understand the papers that explore the quantitative differences between them, let alone be inspired by the known techniques to develop more targeted ones. All of this is only possible by ascending to the higher, more refined languages of Differential Geometry, Lagrangian Mechanics and Numerical Analysis along with their long list of prerequisites.

In my experience, sheer will power is of no use in covering the gap. It took humanity millennia to develop these languages. Languages that open the door to the results that make the whole difference in our context. Languages that even the standard curriculum of applied mathematics does not do justice or give enough time. Only with love, patience, perseverance and pain can they be attained.