Geometry & Computation for Interactive Simulation

Jorg Peters (CISE University of Florida, USA),
Dinesh Pai (University of British Columbia),
Ulrich Reif (Technische Universitaet Darmstadt)

Sep 24 – Sep 29, 2017

1 Overview

The workshop advanced the state of the art in geometry and computation for interactive simulation by introducing to each other researchers from different branches of academia, research labs and industry. These researchers share the common goal of improving the interface between geometry and computation for physical simulation – but approach it with differing emphasis, techniques and toolkits. A key issue for all participants is to shorten process times and to improve the outcomes of the design-analysis cycle. That is, to more quickly optimize shape, structure and properties to achieve one or multiple design goals. Correspondingly, the challenges laid out covered a wide spectrum from hierarchical design and prediction of novel 3D printed materials, to multi-objective optimization minimizing fuel consumption of commercial airplanes, to creating training scenarios for minimally invasive surgery, to multi-point interactive force feedback for virtually placing an engine into a restricted cavity. These challenges map to challenges in the underlying areas of geometry processing, computational geometry, geometric design, formulation of simulation models, isogeometric and higher-order isoparametric design with splines and meshingless approaches, to real-time computation for interactive surgical force-feedback simulation. The workshop was highly succesful in presenting and contrasting this rich set of techniques. And it generated recommendations for educating future generation of researchers in geometry and computation for interactive simulation (see outcomes). The lower than usual number of participants (due to a second series of earth quakes just before the meeting) allowed for increased length of individual presentations, so as to discuss topics and ideas at length, and to address basics theory. Especially in the second half of the week, this encouraged speakers to present ideas on controversial topics, such as illuminating to the downside of interactivity in simulation for some applications, and to present unpublished results.

2 Recent Developments and Open Problems

Performance prediction, robust computation and realistic simulation rely on subtle *interaction of geometric models* and *physical models* (partial differential equations). Creating geometry and computing with geometry are each rich topics by themselves giving rise to multiple communities that each focus on separate, often isolated aspects of geometry and computation. One challenge is therefore to foster new interaction between these different communities.

2.1 Model reduction to enable user interaction with complex simulations

Advances in model reduction, i.e. fast pre-computation of dominant eigen-spaces, have recently enabled interactive simulation of very large, highly-tessellated objects. Fast approximation algorithms are essential for applications where users interact in real-time with a complex simulation. To allow for an effective interaction, many software tools provide a user with a set of predefined controls. Model reduction provides techniques for the design of fast approximation algorithms for scenarios in which the set of user interactions is low-dimensional compared to the degrees of freedom of the system to be simulated. Model reduction approaches follow an offline-online paradigm. In an offline phase, the system to be simulated is observed and analyzed and a low-dimensional system is constructed. This low-dimensional system is fast to evaluate and approximates the complex system in the space of configurations reachable by the user controls. In the online phase, only the low-dimensional system is evaluated. Though methods for real-time simulation of elastic bodies have been introduced, see [6, 7, 8] and references therein, there remain many challenges and open problems. Examples are the construction and efficient handling of non-linear subspaces, error control (a priory and online) for real-time simulation of non-linear systems, methods for fast updating the reduced systems in the online phase and application of reduction to micromaterials of 3D printing and haptic force-feedback techniques.

2.2 Simulation Environments

Interactive simulation has to rely on the rich substrate of opensource software libraries to harness the complexity of the of interactive physics solving not only sophisticated PDE models but also real-time collision detection and response. For example, interactive surgical simulation includes challenges of soft-tissue simulation, of tool organ collision detection, collision response in the form of real-time force-feedback, cutting and tearing and providing a convincing visual appearance. Such challenges go far beyond the usually well-fenced application of numerical algorithms and require the *integration* of several toolkits. Fortunately, many new research results and tools have recently become available through open-source software libraries such as sofa, blender, bullet, CGAL, G+smo, etc. Current challenges lie in the integration and federation of the software packages. For example, one recent such integration, Blender2SOFA, links the geometric modeling interface and capabilities of Blender to the softbody physics simulation capabilities of the Simulation Open Framework Architecture (SOFA) [5]. A second category of challenges includes setting up incentive systems to maintain a pool of contributors and an innovative ecosystem.

2.3 Meshingless spline techniques for design and analayis

In academia, geometric models and interfaces to modeling are typically developed in mechanical engineering or computer science, possibly using tools of approximation theory, while solving geometrically-inspired equations is considered to fall into the realm of numerical analysts in applied mathematics. In industry, the separation between model creation and model analysis has led to a time-consuming cycle between separate geometric design and FEM analysis departments.

Underlying this divide is the difference in mathematical models for representing geometric objects and physical data. While tensor product spline functions are a widely-used standard for the description of high quality surfaces in Computer-Aided Design systems, simulation is mostly based on piecewise polynomial functions on specific tesselations of the domain. Repeated conversion from on representation to another is necessary to the design geometric objects with specific physical properties, and solving this task is considered to be the bottleneck in physics-based geometry design. Conversion is time-consuming and, perhaps even worse, typically entails a loss of accuracy when the exquisit detail of high-quality geometric models cannot be preserved or, conversely, the numerically sophisticated analysis of geometrically simplified models yields suboptimal results on the original.

Several models for unifying the representations of geometric and physical objects have been proposed to overcome this problem. As there seems to be no alternative to the use of splines in surface modeling, these attempts aim at using splines also for the approximation of PDE solutions. One example are *Weighted Extended B-Splines* [2] represent a systematic approach in that direction. Here, a weight function which vanishes at the boundary of the domain is used as a multiplier for standard splines to satisfy essential boundary conditions, and the stability problem is solved by a specific coupling process for B-splines near the boundary. Another approach is called *Isogeometric Analysis (IGA)* by [9]. IGA is based on a piecewise parametrization of the domain in terms of B-splines. Using the same basis functions for the representation of physical data yields a highly consistent framework. IGA has attracted much interest in recent years, but still faces many challenges. (For progress near irregularities in the underlying quad mesh see the last paragraph). Isogeometric Analysis requires a tesselation of the simulation domain into images of boxes. This so-called *hex meshing problem* is known to be extremely difficult and existing algorithms are far from being satisfactory. Hence, alternative spline-based approaches are sought which are meshingless in the sense that the knot grid is regular and completely independent of the shape of the considered geometry.

In the context of PDEs on open subsets of $\mathbb{R}^d$, such methods are facing two major problems: A potential loss of stability and an appropriate incorporation of essential boundary conditions. The first problem can be solved conveniently using extension techniques, as suggested by Höllig, Reif and Wipper [2]. However, it is still not clear how to deal best with boundary conditions. Know approaches leverage weight functions, Lagrange multiplier techniques, and different variants of penalty methods. When applied in combination with variational formulations of elliptic PDEs, a further problem concerns quadrature of functions on trimmed grid cells. While being almost trivial from a theoretical point of view, the technical complexity of this task is considerable. By contrast, collocation methods suggest simple implementation and efficient assembly of system matrices. However, our understanding of convergence properties of such approaches is still limited.

Spline methods can also be applied to PDEs defined on submanifolds. Here, the basic idea is to extend the sought solution to ambient space, approximate the resulting PDE there, and restrict the so-found solution to the submanifold. First promising results of this method are available for parabolic PDEs, while the much harder case of elliptic PDEs is currently under investigation. The results of Dziuk and Elliot [1] are encouraging, but suffer from the fact that the extended PDE is degenerate. Hence, improved extension techniques are sought to obtain uniformly elliptic equations in ambient space.

Another important recent development is the the ability to not only model but compute on surfaces consisting of tensor-product (i.e. 4-sided) pieces with an irregular layout. That is, three or more than four such patches can join together at an iregular point. The main theorem of [3] assures that any surface generated via 'geometric continuity' spawns a corresponding analysis space that must have the same 'geometric continuous' structure and [4] gives a conrete example. At present, the full space of such geometrically continuous constructions is subject of investigation.

3 Presentation Highlights

Based on their emphasis in design and simulation, the presentations can be understood as falling into four categories. This is reflected by the four subheadings below.

3.1 Reducing the complexity to enable interactive simulations for large models.

In his talk *Of Coarse its Fine* , David Levin introduced his approach to simulating fine-scale structure using agregate properties. He observed that traditionally, numerical algorithms for simulating continua have been based on the idea that high resolution and high accuracy go hand-in-hand. And while engineering and computational mechanics have pushed resolution limits in pursuit of simulation fidelity, computer graphics has often gone the other way, sacrificing resolution and accuracy for performance. David posed the question, 'can we obtain both high-performance and predictive accuracy'? The approach presented avoids this seeming fundamental trade-off using measurement-based and data-driven simulation techniques that allow reflecting aggregate properties of the finer resolution in the coarse. Examples of deformable models show that these methods apply both to forward and to inverse problems in graphics and 3D printing.

In *Numerical Analysis in Visual Computing: not too little, not too much*, Uri Ascher introduced the 'eye-norm' of Visual Computing and illustrated it by case studies involving numerical methods and analysis. The first example is motion simulation and calibration of soft objects such as cloth, plants and skin. While the governing elastodynamics PDE system, discretized in space at the variational level via the finite element method, leads to a large, expensive to assemble, dynamical system in time; and damped motion is required to mask highly oscillatory stiffness. The alternative is geometric integration yielding more quantitative computations. The second study is an image processing problem using local approaches that do not necessarily use underlying PDEs.

Klaus Hildebrandt's *Model Reduction for Elasticity-Based Shape Processing* reviewed model reduction techniques for constructing fast approximation algorithms for shape optimization. The goal is to obtain approximate solutions at run times that are independent of the resolution (triangulation density) of the discrete shapes to be optimized. His applications showed real-time elasticity-based shape interpolation and the processing of curves in shape spaces in which a shape (ie the geometric model triangulation) is a single point. Klaus also outlined the concept of compressed vibration modes of elastic bodies, in contrast to the natural vibration modes. This yields more efficient localized ("sparse") deformations.

Alec Jacobson presented 3D printed hierarchically fitting plastic models to illustrate his talk *From Reconfigurables to Matryoshka, Optimizing Shapes and Motions over Space-Time* A reconfigurable is an object or collection of objects whose transformation between various states defines its functionality or aesthetic appeal. For example, a mechanical assembly can be composed of interlocking pieces, a folding bicycle transforms, or consider a space-saving arrangement of apartment furniture. Alec maintained that traditional computer-aided design tools are intended for static objects and reported on new tools for optimizating computational design of reconfigurable shapes. A particularly appealing application is the interactive computational design of generalized Matryoshka, a.k.a. Russian Nesting Dolls.

3.2 Meshingless analysis via higher-order isoparametric or isogeometric design.

A second series of four talks surveyed the state of the art and introduced new open-source software for isoparameteric esign with higher-order smoothly connected computational elements

Angelos Mantzaflaris presentation of a *Versatile software for isogeometric simulations* introduced a new general purpose software implementation G+smo for isogeometric analysis. He discussed how, more than a decade after the introduction of the isogeometric approach, industrialization of the technology remains a major challenge. Angelos traced this back to the need for a new way of thinking (a paradigm shift) with respect to existing computer-aided design and finite element analysis software. He pointed to the educational component since isogeometric simulation, while a big step forward in theory, requires a re-engineering of current design and simulation practices.

Jorg Peters' *Smooth surfaces and Volumes for Analysis* surveyed the state of the art in isogeometric design in the presence of irregularties in the surface; that is when more or fewer than four quadrilaterals (or not six triangular facets) meet and are to be smoothly joined to support analysis functions on the resulting manifold. After a brief overview of different surface representations, Jorg focused on a theorem that guarantees that all G^k constructions yield C^k isogeometric finite elements, and covered alternative polar and singular parameterizations in detail, showing their use for direct, meshingless, isogeometric analysis in each case.

In *G1 multi-patch parametrizations for isogeometric analysis* Thomas Takacs elaborated on the properties of C^1-smooth isogeometric function spaces over multi-patch domains (across irregularities). Thomas characterized isogeometric functions on a B-spline domain as those whose graph surface also has a B-spline representation and the domain of interest is composed of multiple B-spline patches. The presentation focused on a special class of C1-smooth functions for 'analysis-suitable' G^1 multi-patch domains and constructed a basis for a suitable subspace with optimal approximation properties. Thomas advertised the approaches flexibility in that it can be implemented easily and incorporated into an IGA framework.

Denis Zorin's talk introduced *Similarity maps and splines on surfaces* as a new technique to model smooth surfaces of arbitrary topology based on tensor product splines (e.g. subdivision surfaces, free-form splines, T-splines). Denis pointed out that the conversion of an input surface into a representation is commonly achieved by constructing a global seamless parametrization, possibly aligned to a guiding cross-field and using this parametrization as a domain to construct the spline-based surface. (A seamless parametrizations can be thought of as parameterizations of surfaces cut to disks, with isoparametric line directions and spacing on the surface matching perfectly across the cuts). Denis pointed to the fundamental difficulty in designing robust algorithms for subdivision surfaces (requiring a conforming domain mesh) or T-spline surfaces is to reliably obtain a suitable parameterization that has the same topological structure (matching singularities and more generally rotations of parametric line directions along loops matching that of the cross-field) as the guiding field poses a major challenge. To address the problem, Denis introduced the concepts of seamless similarity maps – allowing scale jumps across cuts and splines with half-edge knots, that relax the global knot interval consistency requirements on surfaces with nontrivial genus. He showed that for any given guiding field structure, a compatible parametrization of this kind exists and can be computed. This leads to fully automatic construction of high-order approximations of arbitrary surfaces with highly complex topology. Potentially this enables, automatic conversion of surfaces to isogeometric form.

3.3 Immersed and ambient spline finite elements

An complementary approach to that above, based on parameteric spline surfaces, is taken by immersed and ambient spline spaces for simulation. Here there is no need to a priori conform the elements to the shape.

Fehmi Cirak motivated using *Immersed B-Spline Finite Elements* by advances in manufacturing, most prominently in additive manufacturing or 3d printing. These techniques enable the production of high-performance products with ever increasing functional and geometric complexity from the product-scale in the order of tens of centimetres down to submillimetre scale. To represent and explore the corresponding vast design space, Fehmi proposed robust and scalable immersed/embedded boundary finite elements. In contrast to conventional finite elements these elements do not require generating and maintaining a boundary-fitted mesh. Instead a non-boundary-fitted hexahedral voxel grid associated with tensor-product B-splines can be combined with auxiliary techniques for enforcing boundary conditions. The regular voxel grid enables multiresolution surface and volume techniques already available in computer graphics and computer-aided design.

Based on the same set of reasoning, that goes back to the web-spline apporach of Hollig and Reif, Florian Martin gave a very thoughtful presentation of *WEB-Collocation for Singular and Time-Dependent Problems* . The collocation method with weighted extended B-splines (WEB-splines) uses spline approximation for the solution of stationary partial differential equations. WEB-collocation requires no mesh generation and numerical integration. Martin pointed to considerably faster computation times and an easier implementation, following a thorough review of the basics of WEB-collocation for general boundary value problems with mixed boundary conditions. The advantages over finite element methods were illustrated for Poisson's equation as typical model problem for singular and time-dependent ppartial differential equations. The underlying uniform spline spaces permit a straightforward generalization of the basic concept to hierarchical bases and the development of intuitive refinement strategies. Simulating a tsunami, the combination of the WEB-collocation concept and a time-step iteration was presented to demonstrate a novel approximation scheme for time-dependent equations.

Ulrich Reif reported on *Approximation with Ambient B-Splines and Intrinsic PDEs on Manifolds*. The basic idea is to extend given data on a manifold to the ambient space, for instance by requesting constant values along lines perpendicular to the manifold. In this way, a new function is defined, which can be approximated by standard tensor product splines in the ambient space. Restricting these splines to the manifold yields the desired approximation. It was shown that this method has optimal approximation power, and this result was illustrated by practical examples. The method can also be used to define parametrizations of free-form surfaces of arbitrary topology. In the second part of the talk, the potential of the method for the solution of intrinsic partial differential equation on manifolds was discussed. Here, not a given function but functionals like the Laplace-Beltrami operator must be extended to the ambient space of the manifold. Once this is done, the resulting higher-dimensional PDE can be approximated by tensor product B-splines, and again, the actual solution is found by restricting the solution to the manifold. Theorems guarantee existence and uniqueness of solutions.

Francesca Pitolli showed how to solve a fractional-time diffusion equation by a collocation-Galerkin method that uses the refinable spaces generated by the fractional B-splines as approximating spaces. The main advantage in using the fractional B-splines is in that their derivatives of both integer and fractional order can be expressed in a closed form that involves just the fractional difference operator. We analyze the performance of the method by solving some test problems.

Mahsa Mirzargar gave an overview of *Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering and Its Application* , its connection with well-established concepts from approximation theory, and the recent advances in SIAC filtering.SIAC filters recover some of the hidden convergence of discontinuous Galerkin (DG) methods at the same time as they smooth out the DG output. Since the introduction of Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering for DG approximation for univariate hyperbolic equations by Cockburn et al., many generalizations of SIAC filtering have been proposed. Mahsa focused on new results connecting spline theory and SIAC filtering. She explained how they have paved the way for a more geometric view of this filtering technique. Examples include the introduction of SIAC line integral with applications for streamlining and flow visualization, hexagonal SIAC using nonseparable splines, and position dependent SIAC with nonuniform knot sequences.

3.4 Advances in Toolkits for Geometry and Interactive Simulation

Due to the complexity of the overall challenge of interactive simulation, researchers typically build on

Pierre Alliez presented *The CGAL C++ library: survey and review of recent advances for geometric modeling.* CGAL stands for Computational Geometry and Algorithms library. Pierre outlined the objectives and scope of the CGAL open source project and then focused on recently added algorithms that are relevant for geometric modeling and simulation: point set processing, polygon mesh processing and mesh generation. For each software component he discussed the underlying design principles, showed a live demo and explained how users can adapt and extend CGAL to their specific needs.

Upon popular demand, in a second presentation *Low Distortion Inter-surface Mapping via Optimal Mass Transport* Pierre Alliez presented a novel approach for computing a homeomorphic map between two discrete surfaces – based on CGAL. The work optimizes a mapping by computing a mass transport plan between two surfaces. This non-linear problem, which amounts to minimizing the Dirichlet energy of both the map and its inverse, is solved using two alternating convex optimization problems in a coarse-to-fine fashion. Computational efficiency is achieved through the use of Sinkhorn iterations, modified to handle minimal regularization and unbalanced transport plans.

Hugo Talbot gave an overview of *SOFA, an open-source framework for physics simulation : a tool for research, collaboration and innovation* . SOFA is an open-source framework for multi-physics simulation that specifically aims at interactive and real-time applications, with an emphasis on medical simulation. SOFA benefits today from large, active and international community, including international universities, startups and companies. For more flexibility, SOFA is made up of a stable open-source core and many optional plugins (¿100 plugins), providing innovative numerical methods and state-of-the-art algorithms. The SOFA core has a LGPL license (permissive and non-contaminating) fostering development of prototypes and products under any commercial license. Hugo's talk was followed by Jorg Peters' presentation of the *Blender2SOFA software* . Blender2SOFA is a software bridge that semi-automates the scene-generation cycle, a key bottleneck in authoring, modeling and developing VR units for surgery simulation.

3.5 User interaction or not?

The last category of presentations focus on the human in the loop aspect of interactive computing.

Jernej Barbic presented *Challenges in 6-DoF haptic rendering of complex geometry and virtual assembly.* Six-DoF haptic rendering (at 1000 frames per second) is useful for training, especially for interactive applications in virtual assembly and maintenance of complex machinery, such as, for example, car engines and landing gears. Jernej pointed to the technological challenges for simulations involving complex distributed contact. One aspect is that there are typically many simultaneous individual contacts. This poses stability issues due to accumulated stiffness. In order for simulations to be useful, paths that violate contact due to errors in the contact resolution algorithm must be efficiently eliminated. Friction is non-trivial, due to the large number of contacts and stringent time requirements. Even preparing the signed distance fields and point clouds can take an unreasonably amount of time with models of realistic complexity. Jernej decribed new algorithms for signed distance field generation, continuous collision detection, adaptive stiffness and friction.

Jose Luis Licon Salaiz focused on the visualization of *Computational topology for pattern analysis in turbulent flow* The techniques from computational topology are used to quantify spatio-temporal complexity of environmental data in different levels of the earth's athmosphere, and for detecting minimal flow structures in direct numerical simulations.

Karan Singh's talk *Sketching and Sculpting Simulations* , explored the coupling of interactive sketching and sculpting with physical simulation. Sketch and sculpt interfaces have often been touted as natural approaches to interactive conceptual modeling. Karan explained how use symmetry and montoring of the drafting speed to produce 3D designs that are structurally stable, and to design interactively aided by physical simulation.

Quite to the contrary, Tom Grandine argued that while in his oart of the aerospace industry, interactivity is appreciated by the users, the results for complex multi-objective optimization of shape and materials are actually worse. In *The Case Against Interactive Design* Tom pointed out the inability of design methodology researchers to find positive impact and he presented several studies showing a negative impact as interactivity invited trial and error in high-dimensional spaces.

Jos Stam made a different but similar case for *Modeling through self-assembly* . Usually shapes are modeled through a top down approach for example smooth surfaces defined by a few control points. In his talk, Jos presented the case for an opposite paradigm. This approach, based on a robust framework (nucleus) of physics and collision simulation, creates shapes from a bottom up approach. Shapes emerge from small scale interactions through local interactions. This approach is inspired by micro-biology. Key to this approach is to use a dynamics solver. More details can be found at https://www.autodeskresearch.com/publications/jmi2012.

4 Scientific Progress Made

The meeting brought together leading researchers from mathematics, engineering, and computer science. The researchers were both from academics, research centers and industry. The shared interest in geometry and simulation resulted in a rich palette approaches. The intense group discussions clarified the different emphasis towards combining geometry and simulation: depending on whether the application area is aircraft manufacture or interactive simulation different points on the spectrum

We identified the following list of issues that span the space interest that make concrete the wide range of viewpoints. Participants learned about the varying focus and differing yardsticks of success in each other's areas of specialization.

- Eye ball metric vs convergence order: in computer graphics the emphasis is on the fast generation of visually plausible simulations. In engineering applications the emphasis is on verification and convergence to ground truth. Mathematicians focus on theorems guaranteeing O(h^p) convergence.

- Interactive vs Knowledge-driven: Across the disciplines, optimal design of shape and function is an important issue. Interactive environments encourage fast exploration – but also encourage trial and error approaches that impede the knowledge-driven approach of experienced engineers.

- Wall clock vs CPU clock: thoughtful integration of tools into the human creation process is more often important to engineers then higher speed of individual tasks. If the output of A does not fit into B, and it takes a lot of data adjustment, then a slower, but compatible computation of A may be preferable.

- Interactive vs accurate: the obvious trade-off, turns not to be as obvious when precomputation is permissible. For example in "of coarse it's fine", D. Levin illustrates this point by obtaining apparently accurate results with much coarser models. On the other hand, Tom Grandine's talk pointed out that interactively computed results may lead to local minima.

- Design (styling) and analysis: Smooth graded parameterizations of the computational domain manifold are indispensable for the stability and convergence of solvers for higher-order differential equations. However stylists, say in the car industry, focus exclusively on the visible shape of surfaces disregarding relative orientation, tessellation, even mathematical guarantees of continuity or suitability for later volumetric tessellation.

- Meshing vs Meshingless: Traditional finite elements require a meshing of the computational domain that often does not fully conform to the designed shape. meshingless approaches also use a mesh, but either the mesh is a trivial regular grid partition of the surrounding space, or it is the control net that defines the shape.

5 Outcomes of the Meeting

5.1 Interdisciplinary exchange and opportunities for collaboration

Major outcome was an improved understanding of the issues facing, and tools used, by researchers across disciplines and professional environment: from geometry processing, computational geometry, geometric design, formulation of simulation models, real-time computation for interactive simulation, isogeometric and higher-order isoparametric design. The workshop is part of reversing the tendency towards specialization that risks neglecting feedback to the overarching set of challenges. Below we list pointers and introductions to available tools, code or theory, e.g. opensource software packages and educational resources.

A second series of earth quakes hit Oaxaca state just prior to workshop. This caused a number of last-minute cancellations over the weekend as family and employers weighed in concerned about more damage. Fortunately, the schedule could be restructured to not compromise the overall goal of presenting and bridging the current gap between the approaches and expectations from different fields and from different professions. Participants included industry (Boeing, Autodesk, etc.), research labs (CGAL, SOFA consortium, etc.) and academics from different continents including, in addition to mathematicians, mathematically-oriented faculty in engineering and computer science from different continents. An initial five-minute one-on-one 'mathematical speed dating' event allowed participants to quickly fill each other in on their main interests and lowered the barrier for junior participants. As a result, several participants noted that the discussions were more intense than what they had experienced in venues such as Dagstuhl or Oberwolfach, and certainly than in large conference venues. Specific focal points of the discussion was the availability, accuracy and interactivity of design tools and opensource or proprietary modeling and simulation environments to support the interactive design and analysis cycle and interactive surgical simulation, where model deformation and modification tightly link analysis and design.

5.2 Synopsis of Theory and Tools for geometry and interactive simulation

We framed the discussions of the last day around the questions

- What software foundations are needed?
- What should students be taught?
- What did you learn from areas outside your own?

As a result, we compiled the following two sections on pointers to the literature and the world of useful toolkits.

Books recommended for preparation in the area of geometry & simulation

Finite Elements and Mechanics

C. Lanczos: *The Variational Principles of Mechanics.* Introduction to the variational underpinnings of classical mechanics including rigid and deformable bodies and constrained systems. The first two chapters are recommended to all. Later chapters are for specialization in physics simulation.

J. Bonet and R.D. Wood: *Nonlinear Continuum Mechanics for Finite Element Analysis.* A walk through of the math and implementation techniques behind non-linear FEM for elastica. Notation can be a bit odd as it relies on differentials rather than indicial notation but there are some nice mathematical tricks to be learned.

T. Belytschko et al.: *Nonlinear Finite Elements for Continua and Structures, 2nd ed.* A comprehensive text concerning application of the finite element to solid, fluid, and contact mechanics. Provides a readable mathematical introduction to the topic and has a nice appendix containing shape function formulas and implementation details.

T. Belytschko and J. Fish: *A First Course in Finite Elements.* Focusing on the formulation and application of the finite element method through the integration of finite element theory, code development, and software application, the book is both introductory and self-contained, as well as being a hands-on experience for any student.

J.N. Reddy: *An Introduction to the Finite Element Method.* Explains the mathematical underpinnings, but is also well suited for engineers.

G.Th. Mase and G.E. Mase: *Continuum Mechanics.* Contains a great introduction to indicial notation along with a comprehensive set of exercises to practice.

G. Strang and G. Fix: *An Analysis of the Finite Element Method.* More suitable for math students: more formal mathematically vs typical engineering FEM book, but more clear notation and concise presentation.

D. Braess: *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics.* Standard reference with examples, counter-examples, and exercises.

Geometry Processing and Mesh Generation

J.A. Baerentzen et al.: *Guide to Computational Geometry Processing: Foundations, Algorithms, and Methods.* This book reviews the algorithms for processing geometric data, with a practical focus on important techniques not covered by traditional courses on computer vision and computer graphics.

M. Botsch et al.: *Polygon Mesh Processing.* Introduction to the geometry processing pipeline – originally a SIGGRAPH course turned into a book. Not really a text book, but appropriate for graduate students.

S.W. Cheng, T.K. Dey, and J. Shewchuk: *Delaunay Mesh Generation.* Recent (2012) and most complete guide to Delaunay-based mesh generation.

H. Edelsbrunner: *Geometry and Topology for Mesh Generation.* An excellent text for courses on mesh generation.

Splines and Subdivision

K. Höllig and J. Hörner: *Approximation and Modeling with B-Splines.* Introduction to (rational) Bézier-Curves, (B-)Splines, spline approximation, and multivariate splines. See also http://www.siam.org/books/ot132/ and http://www.siam.org/books/ot132/) for problem collection and interactive spline programs (Matlab).

J. Peters and U. Reif: *Subdivision Surfaces.* Comprehensive description of the analysis of subdivision surfaces with extraordinary vertices.

M. Sabin: *Analysis and Design of Univariate Subdivision Schemes.* Very intuituve description. Less rigorous, but much better accessible than standard journal references.

Exterior Calculus

B. O'Neill: *Elementary Differential Geometry.* Undergrad introduction to differential geometry with emphasis on surfaces in $\mathbb{R}^3$ and differential forms.

K. Jänich: *Vector Analysis.* Undergrad introduction to differential manifolds and differential forms with emphasis on intuition and examples.

D. Bachman: *A Geometric Approach to Differential Forms.* An introduction to forms and exterior calculus taught from a geometric perspective. Provides a relatively painless introduction to forms and exterior calculus for both math and CS students.

Online Materials

D. Baraff and A. Witkin: *Physically-Based Modeling.* Excellent walk through of how physics-based animation is done in graphics.
graphics.stanford.edu/courses/cs448b-00-winter/papers/phys_model.pdf

J. Barbic and E.D. Sifakis: *FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction.* Tutorial for those wanting a computer science introduction to finite element methods as they are used in computer graphics.
femdefo.org

University of Colorado at Boulder: *Introduction to Finite Element Methods.* Introductory material. Notes with practical FEM details (stiffness matrix entries, mathematica code for computation).
colorado.edu/engineering/CAS/courses.d/IFEM.d/

University of Colorado at Boulder: *Nonlinear Finite Element Methods.* Advanced material.
colorado.edu/engineering/cas/courses.d/NFEM.d/Home.html

Software and Toolkits for work in the area of geometry & simulation

Machine Learning

PyTorch and **TensorFlow** are the two most popular machine learning toolkits (developed by Facebook and Google respectively). Both rely on a python API to implement machine learning algorithms such as deep neural nets and come equipped with automatic differentiation and a number of gradient-based optimization schemes for training.

Geometry Processing

CGAL: An open-source and cross-platform C++ library. Some basic computational geometry algorithms and data structures, as well as algorithms for point set and mesh generation and processing are provided.

- Components: A list of components can be found here.
- Demo: Since a few years the library comprises the so-called ”polyhedron demo”: a cross-platform 3D demo with GUI based on Qt, OpenGL and QGlViewer that enables experimenting with most of the 3D algorithms of CGAL: point set processing (denoising, simplification, surface reconstruction), mesh processing (simplification, remeshing, modeling, skeletonization, parameterization, etc), mesh generation (surface triangle and volume tetrahedron). There are also basic editing tools for point clouds and meshes, and for making high-res screen snapshots. If you are on windows there is no need to compile CGAL to try out the demo: a precompiled executable can be downloaded from the list of packages.
- Support: If you get stuck remember that you are not alone. You can subscribe to the user mailing list and then ask for help, ask even naive questions such as ”can I do this?”, or ”I cannot compile”, or even complaints such as ”I hate templates”. You can also suggest changes or addition – as in all open source projects! There is a forum.
- Contribute: You can contribute in many ways: acknowledge, send nice pictures of what you do with it, bug reports, feature requests, pull requests, suggestions, submission of new components, etc.
- Courses: Pierre Alliez has lots of material for teaching CGAL, together with hands-on exercises, just contact him by email. Support is also available through Pierre!

Meshlab: An open-source solution with visualization and basic mesh processing algorithms. Many people use it also to generate figures for their paper.

LibIGL: A simple C++ geometry processing library. There is a wide functionality including construction of sparse discrete differential geometry operators and finite-elements matrices such as the cotangent Laplacian and diagonalized mass matrix, simple facet and edge-based topology data structures, mesh-viewing utilities for OpenGL and GLSL, and many core functions for matrix manipulation which make Eigen feel a lot more like MATLAB.

GPToolbox: A toolbox of useful matlab functions for geometry processing. There are also tools related to constrained optimization and image processing. Typically, these are utility functions that are not stand-alone applications. Includes matlab scripts for creating SIGGRAPH quality visual output.

BezierView: BezierView (short bview) is a lightweight viewer that renders Bézier patches, rational Bézier patches and polygonal meshes. It provides a simple tool to analyse surfaces based on curvature plots, curvature needle plots, and highlight line plots.

Simulation

SOFA: An open-source solution for interactive physics simulation (mesh-based & meshless) based on C++, e.g., dedicated to continuum mechanics, basic thermodynamics. Free plugin creation (self-licensing) and free support available with Hugo. Plugins available for Unity, soon Unreal.

G+Smo: (Geometry + Simulation Modules, pronounced ”gismo”) is a new open-source C++ library that brings together mathematical tools for geometric design and numerical simulation. It implements the relatively new paradigm of isogeometric analysis, which suggests the use of a unified framework in the design and analysis pipeline. Currently in Beta release.

Project Chrono: A FEM package from Dan Negrut's group. Project Chrono represents a community effort aimed at producing a physics-based modelling and simulation infrastructure based on a platform-independent, open-source design. The name of this software infrastructure is Chrono. Some of its features are listed below. More information is available at the project website. The applications areas in which Chrono is most often used are vehicle dynamics, robotics, and machine design. In vehicle dynamics, Chrono has mature support for tire/terrain interaction modeling and simulation.

Visualization & Rendering

Unity: A C # based game engine which includes rendering, simulation, audio and geometry tools. Usually used for just the rendering component. Note you can call C++ code from C #.

Unreal: A C++ based open-source game engine which includes rendering, simulation, audio and geometry tools. Usually used for just the rendering component.

Paraview: An efficient scientific visualization with standard vtk (and others formats).

Alembic: An open-source file format for portable saving and loading of simulation data into VFX tools such as Maya and Blender.

Houdini: A Software for VFX physics simulation. Plugin architecture allows anyone to import new algorithms into the software. Used as a rendered for fluid simulation in computer graphics.

QT: An open source UI toolkit.

QGLViewer: An open source 3D viewer.

OpenCascade: An open source CAD kernel library, comparable, e.g., to ACIS from Dassault.

FreeCAD: An open source parametric 3D CAD modeller with a GUI and Python interface.

Programming Tools

MATLAB: A convenient programming language for scientific computing, providing a huge set of standard routines from all areas of mathematics. Cheap for students, relatively cheap for academic use, not cheap otherwise.

SciLab: An open source Matlab-like software with large associated documentation.

Octave: The GNU open source alternative to Matlab.

Python: An object-oriented, interpreted, and interactive programming language.

Julia: A high-level dynamic programming language for numerical computing.

CUDA: A toolkit for nvidia GPU implementation, powerful using the associated SDK tools (cusparse, thrust, etc.). However, multithreading power is increasing and could soon overtake on GPU computing.

References

[1] G. Dziuk and Ch.M. Elliot, Finite element methods for surface PDEs, *Acta Numerica* **22** (2013), 289–396.

[2] K. Höllig, U. Reif, and J. Wipper, Weighted extended B-spline approximation of Dirichlet problems *SIAM Journal on Numerical Analysis* **39/2** (2001), 442–462.

[3] David Groisser and Jörg Peters. Matched G^k-constructions always yield C^k-continuous isogeometric elements. *Computer Aided Geometric Design*, 34:67–72, 2015.

[4] Kęstutis Karčiauskas, Thien Nguyen, and Jörg Peters. Generalizing bicubic splines for modelling and IGA with irregular layout. *Computer Aided Design*, 70:23–35, Jan 2016.

[5] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. Sofa an open source framework for medical simulation. In *Medicine Meets Virtual Reality (MMVR'15)*, Long Beach, USA, February 2007.

[6] Jernej Barbic and Doug L. James, Real-Time subspace integration for St. Venant-Kirchhoff deformable models, *ACM Trans. Graph*, 2005,3,24, 982–990.

[7] Optimizing cubature for efficient integration of subspace deformations, An, Steven S. and Kim, Theodore and James, Doug L. *ACM Trans. Graph.* 2008,5,1–10, 27.

[8] An Efficient Construction of Reduced Deformable Objects, von Tycowicz, Christoph and Schulz, Christian and Seidel, Hans-Peter and Hildebrandt, Klaus*ACM Trans. Graph.*, 2013, 6, 213:1–213:10, 32.

[9] Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y., Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering 2005, 194, 4135–4195.