An overview of Geometry plus Simulation Modules

Angelos Mantzaflaris

An overview of Geometry plus Simulation Modules

Angelos Mantzaflaris

Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur, France
angelos.mantzaflaris@inria.fr

Abstract. We give an overview of the open-source library "G+Smo". G+Smo is a 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. G+Smo is an object-oriented, cross-platform, fully templated library and follows the generic programming principle, with a focus on both efficiency and ease of use. The library aims at providing access to high quality, open-source software to the community of numerical simulation and beyond.

Keywords: C++ · B-splines · NURBS · isogeometric analysis · geometric design.

1 Introducing isogeometric analysis

Isogeometric analysis suggests the use of common spline basis for geometric modeling and finite element analysis of a given model. Engineering process begins with designers encapsulating their concepts on the computer using Computer Aided Design (CAD) software. With the finite element analysis (FEM) technology, the CAD model must be first meshed before any numerical simulation can be performed on it. The mesh generation results in a piecewise linear approximation of the original model, that both increases the data volume and introduces unwanted approximation errors in the geometric description. Moreover, the mesh generation phase involves tedious manual work for producing finite element meshes that are suitable for analysis. Last but not least, the results of the analysis should be projected back to the original CAD model, for allowing the designer to update it accordingly. The data transfer between models suitable for design (CAD) and analysis (FEM) poses a severe problem in industry today.

To address these issues Hughes and co-workers introduced in [4] an analysis framework which is based on NURBS (Non-Uniform Rational B-Splines), which is standard technology employed in CAD systems. They propose to keep the exact CAD geometry by NURBS surfaces, and apply numerical simulation using the same NURBS as basis functions for the analysis [2].

2 General concept of the library

The design and software development philosophy of G+Smo was triggered by the observation that simply augmenting existing libraries for geometric design or numerical analysis will not succeed, since the implementation of isogeometric methods requires a fundamentally new approach to unlock the full set of benefits of the isogeometric paradigm.

The goal of G+Smo is to realize the seamless integration of Finite Element Analysis (FEA) and Computer-aided design (CAD) with open-source code from and to the isogeometric analysis community.

Three general guidelines have been set for the development process. Firstly, we promote both efficiency and ease of use; secondly, we ensure code quality and cross-platform compatibility and, thirdly, we always explore new strategies better suited for isogeometric analysis before adopting FEA practices.

G+Smo is an object-oriented, templated C++ library, that implements a generic concept for IgA, based on abstract classes for geometry map, discretization basis, assemblers, solvers etc. It makes use of object polymorphism and inheritance techniques in order to support a variety of different discretization bases, namely B-spline, Bernstein, NURBS bases, hierarchical and truncated hierarchical B-spline bases of arbitrary polynomial order. The library was first announced in the extended abstract [5], a more detailed description is contained in [6].

3 Open-source license

The library is licensed under the Mozilla Public License v2.0 (MPL). The MPL is a simple copy-left license. The MPL's "file-level" copy-left is designed to encourage contributors to share their modifications with the library, while still allowing them to combine the library with code under other licenses (open or proprietary) with minimal restrictions. The latest revision can be downloaded from GitHub: https://github.com/gismo.

4 Modules

The library is partitioned into smaller entities, called modules. Examples of available modules include the dimension-independent NURBS module, the data fitting and solid segmentation module, the PDE discretization module and the adaptive spline module, based on hierarchical splines of arbitrary dimension and polynomial degree.

At the present point, the B-spline, Bernstein, NURBS bases, and also the truncated and non-truncated hierarchical B-spline bases (cf. [3]) are in a stable and functional state, for general dimension. From the simulation side, isogeometric simulation algorithms with the Galerkin approach, notably over non-conforming multi-patch physical domains are implemented and in a stable state. A module treating a large class of linear and non-linear elasticity problems was also released recently [7]. A module dedicated to isogeometric multigrid solvers on multipatch domains is also available [8].

Another interesting module is the assembler module based on *expression templates*, which are a powerful C++ technique for the development of user-friendly and versatile code. They have the advantage of bringing the implementation of a numerical method much closer to the mathematical notation. This allows the easy formation of isogeometric Galerkin matrices using a high-level language, without sacrificing efficiency.

The library takes advantage of modern, efficient move semantics, which are available in recent C++ versions, for more efficiency and cleaner code.

5 Code management and documentation

In order to deliver a successful library, quality of the code needs to be assured. To this end, the CMAKE cross-platform compilation system is used in the library, which allows for seamless installation of the library on different platforms. Moreover Jenkins and a CDASH testing servers are in place for executing regular nightly builds, memory checks and unit-tests on different platforms (Windows, Linux and MacOSX). Documentation is done mostly in the form of in-source text, using Doxygen as well as in the Wiki pages `http://gs.jku.at/gismo`. The documentation is currently available online in the form of HTML pages to all the users, and is updated regularly, as we actively continue adding new material, see `https://gismo.github.io`.

6 Third-party dependencies

Apart from the C++ Standard Library, we use open-source third party software for common tasks, for example the Eigen (`http://eigen.tuxfamily.org`) linear algebra library for (sparse) matrices and linear algebra computations, as well as tools for argument parsing, input and output of XML formatted files, the openNurbs library for reading Rhino's 3DM commercial CAD format (`https://www.rhino3d.com/opennurbs`), Trilinos (`https://trilinos.github.io`) for high performance solvers etc.

7 Plugins and extensions

Concerning plugins, visualization and user interfaces, we developed a plug-in of G+Smo for the Axl geometric modeling platform ([1], `http://axl.inria.fr`. This provides an interactive GUI (graphical user interface) that allows the user to manipulate multi-patch geometries by their control points and work with complicated CAD models. Nevertheless, our primary approach to visualization is based on the well-established Paraview software (`https://www.paraview.org/`)and has been enhanced, e.g., with trimmed surface visualization, gradient fields and absolute error plots, etc. Another connection has been setup with Siemens' Parasolid geometric CAD kernel (https://www.plm.automation.siemens.com), which is widely used in industry. Currently, this allows us to exchange data and to use the advanced modeling utilities available in Parasolid from within G+Smo.

Acknowledgement G+Smo is jointly developed by several contributors at the Johannes Kepler University, at the RICAM Institute of the Austrian Academy of Sciences (in the frame of the Austrian Science Fund NFN project S117) at INRIA and at the Department of Applied Mathematics of TU Delft (The Netherlands). More contributions have been made by developers from other institutions. The full list of contributors is available at https://github.com/gismo/gismo/wiki/About--G-Smo.

References

1. Christoforou, E., Mantzaflaris, A., Mourrain, B., Wintz, J.: Axl, a geometric modeler for semi-algebraic shapes. In: Davenport, J.H., Kauers, M., Labahn, G., Urban, J. (eds.) Mathematical Software – ICMS 2018. pp. 128–136. Springer (2018). https://doi.org/10.1007/978-3-319-96418-8_16
2. Cottrell, J., Hughes, T., Bazilevs, Y.: Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley, Chichester (2009)
3. Giannelli, C., Juettler, B., Kleiss, S.K., Mantzaflaris, A., Simeon, B., Speh, J.: THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. Computer Methods in Applied Mechanics and Engineering **299**, 337 – 365 (2016). http://dx.doi.org/10.1016/j.cma.2015.11.002
4. Hughes, T., Cottrell, J., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comp. Meth. Appl. Mech. Engrg. **194**(39–41), 4135–4195 (2005). 10.1016/j.cma.2004.10.008
5. Juettler, B., Langer, U., Mantzaflaris, A., Moore, S., Zulehner, W.: Geometry + simulation modules: Implementing isogeometric analysis. Proc. Appl. Math. Mech. **14**(1), 961–962 (2014). http://dx.doi.org/10.1002/pamm.201410461
6. Langer, U., Mantzaflaris, A., Moore, S., Toulopoulos, I.: Multipatch discontinuous Galerkin isogeometric analysis. In: Isogeometric Analysis and Applications, pp. 1–32. Lecture Notes in Computational Science and Engineering, Springer (2015). https://doi.org/10.1007/978-3-319-23315-4_1
7. Shamanskiy, A., Simeon, B.: Isogeometric simulation of thermal expansion for twin screw compressors. IOP Conference Series: Materials Science and Engineering **425**, 012031 (nov 2018). https://doi.org/10.1088/1757-899x/425/1/012031
8. Takacs, S.: Fast multigrid solvers for conforming and non-conforming multi-patch isogeometric analysis, arXiv preprint https://arxiv.org/abs/1902.01818 (2019).