TERRIFIC
Enhancing Interoperability

SEVENTH FRAMEWORK PROGRAMME

Deliverable D6.1 of Task 6.2

Initial version of the TERRIFIC Isogeometric Toolkit

Circulation:	PU[1]
Partners:	SINTEF, INRIA, UNIKL
Authors:	Vibeke Skytt (SINTEF), Bernard Mourrain (INRIA), Anh-Vu Vuong (UNIKL)
Version:	1.1
Date:	March 26, 2012

[1] PU=Public

Copyright

Document History

Vers.	Issue Date	Stage	Content and changes
1.0	8^{th} March 2012	100%	Ready for submission
1.1	26^{th} March 2012	100%	Updated submission level

Executive Summary

This document contains deliverable D6.1 of the TERRIFIC FoF STREP Project.

This deliverable describes the initial version of the TERRIFIC isogeometric toolkit.

Contents

1 Introduction

An isogeometric toolkit provides tools for isogeometric analysis. In some applications, for instance shape optimization or non-linear simulation problems, an isogeometric solver may be such a tool, but in most cases isogeometric tools can be found in the area of spline spaces, geometry construction and interrogation, and visualization.

The concept of an isogeometric toolkit was first developed in the EU project EXCITING in the FP7 transport program, see www.exciting-project.eu. There an isogeometric toolkit was assembled with contributions of background software from SINTEF (SISL and GoTools) and INRIA (Axel). These libraries have been developed, tested and used in a variety of applications during a number of years. The toolkit was extended in the project period 2008-2012, through work both within the project and in parallel (national) projects. The initial version of the TERRIFIC toolkit is thus identical with the final version of the EXCITING toolkit. The existing toolkit components are provided by three TERRIFIC partners:

SINTEF: SISL and GoTools

INRIA: Axel, parameterization tools, optimization tools and heat and flow solvers

UNIKL: Structural solvers. Note that the research group providing these tools has moved from TU Munich (for EXCITING) to TU Kaiserslautern (for TERRIFIC).

Each contribution will be described in this document.

2 Organization of tools

The tools are located on INRIA's GForge server as a project by the name TERRIFIC.

The tools are organized in separate modules with simple interdependence. GoTools depends on SISL. Axel and the parameterization tools make use of the GoTools tools and so do the structural solvers from UNIKL.

The tools are implemented in C++ except for SISL that is written in plain C. The code is organized in a makefile system using CMake. The code, except for SISL which has its own manual, is equipped with doxygen comments enabling a user to generate a manual. The doxygen documentation contains information on different levels, from general information related to a module to an explanation of the parameter list of some function within a specific class in a module. Some parts of the toolkit may have a more complete documentation than others, but all modules are documented.

SISL was developed in a Unix environment and has since then been compiled and used under a number of operating systems. GoTools has been developed on a Linux platform using gcc, but runs also on Windows with Visual Studio 2008 or newer. The INRIA software, i.e. AXEL, the solvers integrated in AXEL, the parameterization and optimization tools, support Linux Fedora/Ubuntu and Max Os Lion/Snow Leopard. The gcc compiler must be no older than 4.2, cmake 4.6 or newer and QT 4.7 or newer. The structural solvers are developed in Ubuntu Linux with gcc 4.4.3, but are expected to run on all common Unix systems. They are successfully built in a Windows environment.

Some central functionalities in the toolkit are explained through example programs. Some of these example programs explain the use of a single function like the computation of the closest point in a curve to a given point. Others illustrate how to organize different functionalities to achieve some purpose, typically to construct a model of some complexity. One example program in GoTools/trivariatemodel even constructs a model of a simplified mid ship including deck and stiffeners.

3 SISL and GoTools

SISL and GoTools constitute a set of geometry libraries. SISL has got a functional interface while GoTools is mainly accessible through class interfaces.

3.1 Modularization of SISL and GoTools

The main modules for the GoTools part of the isogeometric toolkit are gotools-core, trivariate, compositemodel and trivariatemodel, but some of these modules depend on SISL, the triangulation library ttl, and other GoTools modules. There is also an independent access to the supporting modules, but they are not equipped with example programs and may have a more limited documentation. In the following an overview of all modules is given.

SISL SINTEF's spline library. This is a mature library written in C. It handles NURBS curves and surfaces. It contains a number of methods to define geometry, but its strength lies in operations related to curves and surfaces, in particular intersection functionality. SISL does not handle trimmed surfaces and has no topology structures.

ttl A triangulation library where the representation is based on a half edge topology. Used by compositemodel.

gotools-core This GoTools module partly overlaps with SISL. It contains parametric curves and surfaces and operations upon them. The entities are splines, whether rational or not, and elementary curves and surfaces. Trimmed surfaces are included in the module.

- Independent classes for handling of points, bounding boxes, directional cones and other supporting entities
- Construction methods for spline curves and surfaces
- Operations on the various entities
- Curve and surface smoothing and point approximation to produce curves and surfaces

topology Structures for computing the topology of a surface model. Used by compositemodel.

trivariate Volumetric entities.

- Spline volumes and elementary volumes
- Operations on volumetric entities
- Some creation methods
- Spline volume fairing

parametrization Parametrization of scattered data points. Used by compositemodel.

implicitization Approximating spline curves and surfaces by an algebraic representation. Used by the intersection module.

intersection Intersections involving spline curves and surfaces, and surface self intersection. Used by compositemodel.

igeslib Read from and write to IGES format. Conversion between SISL and GoTools data structures

compositemodel View a set of curves or surfaces as one entity and perform operations on this model

- Topology structures for surface sets and boundary represented solids
- Fetch topological information from a collection of surfaces
- Operations on a set of surfaces seen as one entity
- Factory for creation of surface sets and composite curves
- Some Boolean operations on surface sets
- Regularization of surfaces and surface sets, i.e. replacing trimmed surfaces by a number of trimmed surfaces with 4 boundary curves or non-trimmed spline surfaces.

trivariatemodel Volume sets.

- Topology structures for volume models
- Topology analysis for a subset of possible collections of volumes. If the volume collection is organized in a corner-to-corner configuration, the topology analysis can be applied.
- Fetch topological information from a collection of volumes
- Some construction methods for volume models
- Interrogation functionality for volume models

isogeometric_model This module represents an interface between the GoTools functionality and an isogeometric analysis application and is intended for multi block models in 2D and 3D. The module was specified by UNIKL and SINTEF, and the 2D part of it has been implemented, although not used in an analysis setting so far. It has thus not yet proved its usefulness. An example program related to this module illustrates how to build the interface and fetch information from it.

viewlib Viewer for curves and surfaces. This module uses openGl and Qt.

3.2 SISL and GoTools as tools for isogeometric analysis

GoTools is organized into modules to divide a complex system into manageable pieces and to allow an application to use only its central parts. GoTools uses SISL functionality whenever appropriate, but a user of the isogeometric toolkit seldom needs to relate to SISL directly. However, looking at GoTools from the field of isogeometry, the functionality may be grouped as follows:

Spline space exploration, refinement, geometry interrogation Refine, degree elevate, evaluate and fetch other information from spline curves, surfaces and volumes and the associated spline spaces. This functionality can be found in gotools-core/geometry and trivariate depending on the type of entity.

Geometry construction and import Read curves, surfaces and volumes from a file or construct one entity of a specific type. It is possible to read IGES files and files in the internal GoTools format. This is done in the module igeslib. Geometry construction is performed in gotools-core/geometry, gotools-core/creators or trivariate depending on the entity type.

Manipulation of geometry entities Spline curves, surfaces and volumes may be smoothed. For volumes and 2D surfaces this operation is equivalent to a change in the parametrization. Smoothing combined with point approximation or other constraints is also possible. These methods lie in gotools-core/creators or trivariate. A trimmed surface with 4 boundary curves can be approximated with a spline surface. This functionality is placed in the compositemodel module.

Multi block data structures Topology structures to represent adjacency in a surface or volume set can be found in compositemodel and trivariatemodel. Topology analysis for surface sets is performed in the module topology, but this is hidden behind an interface in compositemodel. A data structure more directed towards isogeometry is found in isogeometric_model for the surface case. The initial data including adjacency information is fetched from a surface model as represented in the compositemodel module.

Methods for creation of multi block models A multi block model may be created one block at a time and the application is responsible for maintaining the appropriate continuity. Some functionality for a more direct construction of multi block models can be found in compositemodel and trivariatemodel, but note that this functionality is still limited and immature. Consult the doxygen documentation and the example programs for the available possibilities.

4 Parametrization tools

In the isogeometric framework, the parametrization of the computational domain, which corresponds to the mesh generation in FEA, has an impact on the analysis results and on the efficiency. Moreover, in FEA, one can perform arbitrary refinements on the computational mesh, but in isogeometric analysis (IGA) using trivariate B-splines, the refinement is not arbitrary, we can only perform refinement operations in the three parametric directions by knot insertion or degree elevation. The parametrization of a computational domain in IGA is determined by control points, knot vectors and the degrees of B-spline objects. For IGA problems of three dimensions, the knot vectors and the degree of the computational domain parametrization are determined by the given boundary surfaces. Hence, finding the proper placement of inner control points is a key issue in IGA. In the current version of the isogeometry toolbox, five methods are implemented for the 2D and 3D parametrization of the computational domain. They are available in the plugin provided by the package `parametrization` (see figure 1 and figure 2):

Discrete Coons mask method. Given boundary control points, the inner control points can be represented by linear combinations of the boundary control points. Several masks are used to define different rules to generate the inner control points.

Constraint optimization method. Firstly, two kinds of sufficient conditions for injective B-spline parametrization are derived with respect to the control points. Then the parametrization of the computational domain is computed by solving a constraint optimization problem, in which the constraint conditions are the sufficient conditions for injectivity of the planar B-spline parametrization, and the optimization term leads to the minimization of quadratic energy functions related to the

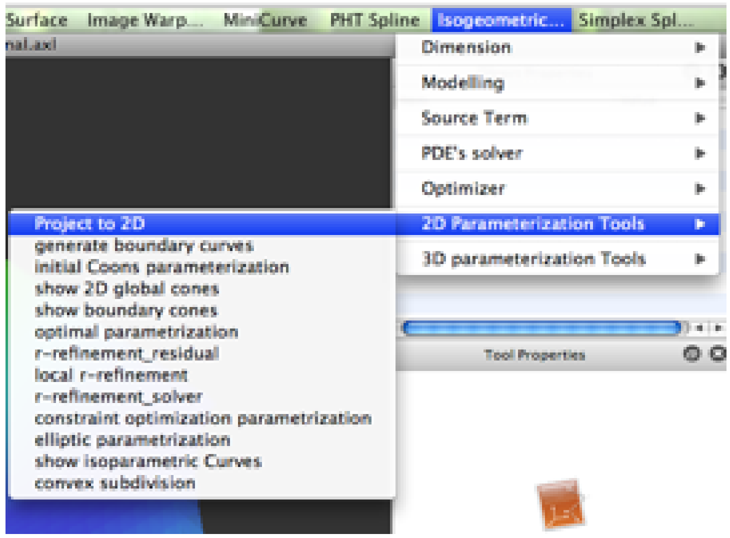

Figure 1: Interface for 2D parametrization tools

first and second derivatives of the planar B-spline parametrization. By using this method, it is guaranteed that the resulting parametrization has no self-intersections, and the isoparametric net has good uniformity and orthogonality properties.

Variational harmonic method. For mesh generation in FEA, there is a general method that determines the grid points through the solution of an elliptic partial differential equation system with Dirichlet boundary conditions on all boundaries. There are several advantages offered by elliptic mesh generation. The theory of partial differential equations guarantees that the mapping between physical and transformed regions will be one-to-one. Another important property is the inherent smoothness of the solution of elliptic systems. A disadvantage of the elliptic method is that there will be some non-uniform grid elements near convex (concave) parts of the boundary. Different from the usual elliptic mesh generation method, the method employed here focusses on the isogeometric version (the coordinates of the interior control points are unknown variables), and converts the elliptic PDE into a nonlinear constraint optimization problem. A regular term is integrated into the optimization formulation to achieve a more uniform grid.

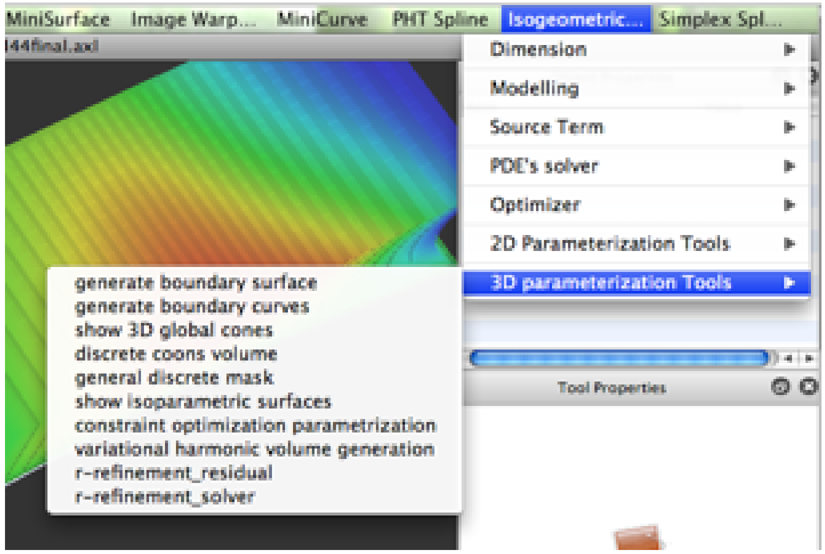

Figure 2: Interface for 3D parametrization tools

r-refinement for optimization of the computational domain. r-refinement in FEA is a re-meshing operation on the computational domain to minimize the cost function while keeping the number of elements constant. For isogeometric problems with unknown exact solutions, a residual-based a-posteriori error estimation is used for isogeometric analysis. The a-posteriori error estimation is obtained by inverse mapping from the solution field to the computational domain, which uses the basic idea of isogeometric analysis. Using this error estimation, r-refinement can reposition the inner control points to minimize the error estimator. The difference between r-refinement and h-refinement, p-refinement or k-refinement is that the number of control points and the degree of basis functions are unaltered, and the positions of control points on the computational domain are subjected to change with the required degree of accuracy.

Local r-refinement. The local error indicator is calculated patch by patch, then the patches on which the local error indicator is bigger than the marking threshold will be marked. Finally the control points corresponding to the marked patches are repositioned by minimizing the error indicator.

The proposed methods produce a suitable and optimal parametrization for isogeometric analysis. 2D and 3D parametrization problems are both implemented in this part. Besides, there are also some other functionalities in this part, such as the generation of iso-parametric curves and surfaces, the generation of boundary surfaces and curves from a spline volume or surface, and the illustration of injectivity cones to show the quality of the resulting parametrization of the computational domain. A first multi-block volume parametrization method of the computational domain is also provided.

5 Optimization tools

A C++ library for isogeometric shape optimization has been developed in the package `optimization` of the toolbox. The optimization tools are designed according to an "ask and tell" architecture. The following algorithms have been implemented:

- A Steepest Descent method including an adaptive line search technique and a centered finite-difference approximation of the gradients
- An Evolution Strategy based on a Gaussian sampling of the search space and population recombination
- An Evolution Strategy based on covariance matrix adaptation
- Quasi-Newton algorithm
- Quasi-Newton BFGS algorithm
- Hierarchical multi-level optimization method based on knot insertion

A plugin in the modeler Axel can be used to set all the parameters required to define the design optimization problem (see figure 3), in particular, the choice of the design parameters (control points of a starting B-Spline curve for 2D problems) and their degrees of freedom (x or y), the choice of the objective function and the possible constraints, as well as the choice of the optimizer and its parameters. Then a function can be called to solve the optimization problem, yielding two new B-Spline curves: one for the optimized B-Spline curve and one for the corresponding physical solution.

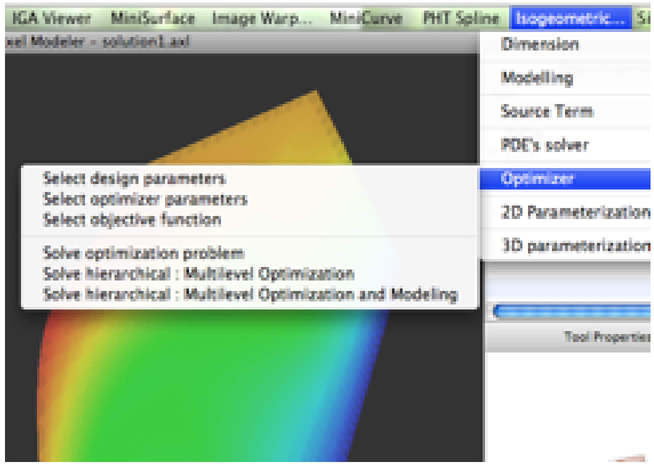

Figure 3: Interface for the isogeometric optimizer

6 Axel as an integration framework

The isogeometric paradigm aims at connecting the representation of the geometry and the function basis used in numerical solvers for simulation problems on this geometry. We implement this concept by developing the geometric modeler AXEL as an integration platform for isogeometric analysis. It provides tools to visualize, manipulate and edit various parametric objects including rational curves/surfaces, B-spline curves/surfaces/volumes, NURBS curves/surfaces/volumes; to select and run different isogeometric solvers on these models and to render graphically the computed solutions.

In 2D isogeometric analysis, the solution field is represented by a B-spline surface with 3D control points. Similarly, for 3D isogeometric analysis, a B-spline volume with k-D control points is used to represent the solution field, where k is the number of kinds of unknown solution variables. This unified mathematical representation of geometric model and analysis results is beneficial for visualization in isogeometric analysis.

A plugin named `QIGAVisualizationPlugin` has been developed in Axel for isogeometric analysis. As for post-processing in finite element analysis, color-map and iso-value objects are used for the visualization and analysis of simulation results. The color-map tools (also known as pseudo-coloring) involve assigning a different color to each value in the function's range, such that color visually conveys numeric information. For points in the domain other than the input grid points, the function can be interpolated, and the resulting scalar can be used with the color map. A C++ class named "QHSV-colormap" is added to convert the solution value into HSV or RGB color information. The color-map bar is also added into the visualization plugin to show the numerical values with respect to the color information. Clipping plane, which is a kind of OpenGL tool, is used for the visualization of the interior color map in isogeometric solution volumes.

Iso-value object has an implicit form to represent the solution part which has the same solution values. It is an important tool for field visualization. The iso-surface in the 3D case for a given value v is the set of all points in the domain for which the function has the value v. Under a fairly common set of conditions, this point set is a surface or set of surfaces. For the 2D case, this concept reduces to an iso-curve or isocline. The iso-curve is the same thing as a contour line in topographical maps. The iso-surface is the 3D analog of an iso-curve. The classical marching square/cube method is employed in our implementation for rendering iso-value objects. See figure 4 for the interface.

7 Solvers for heat conduction and flow problems

We developed a plugin in Axel providing different types of solvers for heat conduction and compressible flow. It consists of three parts:

- Isogeometric solver for the 2D heat conduction problem;
- Isogeometric solver for the 3D heat conduction problem;
- Isogeometric solver for the 2D compressible flow problem.

The isogeometric solver includes a C++ tutorial for solving isogeometric problems. The whole algorithm for solving the corresponding isogeometric problem can be described as follows:

- Computation of the matrix coefficients.
- For each knot span:
  - Computation of the B-spline function and gradient values at the Gauss points;

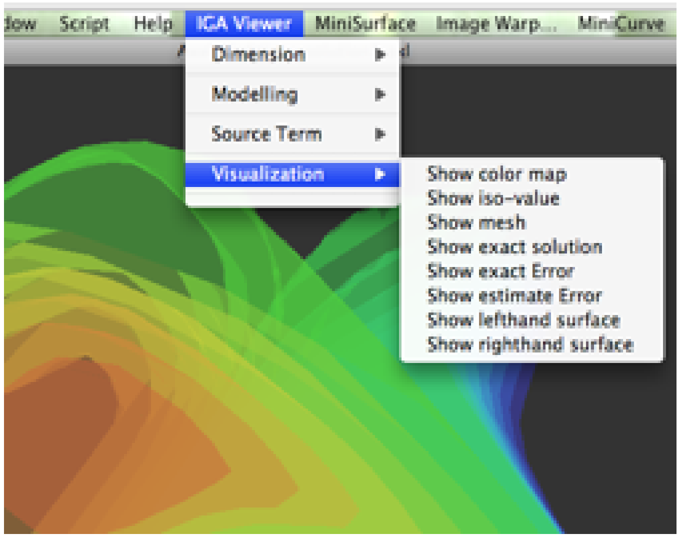

Figure 4: Interface for visualization

  - Computation of the transformation matrix and Jacobian at the Gauss points;
  - Calculation of the stiffness matrix coefficients;
  - Assembly of the matrix;

- Computation of the right-hand side terms (source terms and Neumann boundary contributions);
- Imposition of values for boundaries with Dirichlet conditions;
- Solving of the linear system;
- Possible refinement process.

In this plugin, the user can input the boundary conditions and select the solver parameters interactively. GoTools is used to evaluate the basis functions and their derivatives. The visualization of analysis results is performed by the visualization plugin of the toolbox.

8 Structural solvers

The isogeometric structural solvers were developed in the EXCITING project.

The application domain of this software is the numerical solution of linear elasticity problems. These can be posed in two and three dimensions based on the geometric descriptions offered in GoTools via spline surfaces and volumes. In this way it is possible to directly apply the simulation on a given geometric spline description, which is one of the core ideas of isogeometric analysis.

The basis function evaluation in isogeometric analysis is a little bit more involved than in standard FEM. Nevertheless, in computer aided geometric design different advanced algorithms for evaluation and refinement are known. The implementation makes use of these possibilities provided by the GoTools library, which range from the data structures that represent a parametrization to evaluation techniques. A main achievement regarding these solvers is the interaction between the geometry oriented GoTools library and the features of the numerical routines added within EXCITING.

A short overview of the modules in this part of the isogeometric toolbox is shown in Tab. 1.

Name	Description	Toolbox dependencies
`IA_shared`	Common support routines for the solvers. Include, for example, the parsing instruction, an interface for equation solvers and post processing.	gotools-core, trivariate
`LinElast2D`	Solver for plane stress.	gotools-core
`LinElast3D`	Solver for linear elasticity in three dimensions.	gotools-core, trivariate

Table 1: Module list for the structural solvers

An external parsing library (libconfig++) for reading the input file and suitable routines for numerical linear algebra were included. The data format of the open source FEM solver getfem++ is used and an interface to the sparse direct solver UMFPACK has been established. For post processing and visualization the results of the simulation are formatted into the vtk format, so that they can be utilized by post processing tools like Paraview.

9 Summary

The initial version of the TERRIFIC toolkit contains tools in the following areas:

- Basic spline functionality for the 1–, 2– and 3–variate case
- Construction methods for single geometry entities
- Topology structures for 2–variate and 3–variate geometry entities
- An initial set of construction methods for models of surfaces or volumes
- An interface to IGES
- Parametrization tools
- Visualization tools
- Structural solvers and solvers for heat conduction and compressible flow. Note that the solvers relate to one patch.
- Optimization tools

The basic geometry functionality is quite complete, although new needs may arise when the toolkit is used in new contexts. The toolkit does not emphasize non-geometrical tools to be used in isogeometric solvers, and there is a lack of multi-patch solvers. The construction methods for complex isogeometric models are not sufficient. The current toolkit misses a STEP interface. Most of the mentioned shortcomings of the toolkit will be addressed during the project period. Moreover, new types of tools will be added through the application work packages, WP1 to WP4. These topics are addressed in the deliverable D6.2 of WP6.