

# libigl: Prototyping Geometry Processing Research in C++

Daniele Panozzo<sup>1</sup> and Alec Jacobson<sup>2</sup>

<sup>1</sup>New York University

<sup>2</sup>University of Toronto

## Abstract

*Modern geometry processing algorithms depend on an ever-growing toolbox of fundamental sub-routines and data structures. Prototyping from scratch requires much time building basic tools rather than focusing on the novel research idea. Many existing code libraries have unsatisfactory APIs and the time spent implementing sub-routines is often replaced with time spent learning complex, templated object hierarchies or memory layouts. Libigl is a C++ library of geometry processing algorithms designed for and by researchers. Its wide functionality includes construction of common sparse discrete differential geometry operators (such as the cotangent Laplacian), 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. Libigl places extreme importance on ease of use and experimentation. To this end, algorithms are directly exposed as functions taking simple matrix types as inputs and outputs. Libigl is a "header only" library and compiles on Windows, Mac, and Linux. In this course, we will walk through an introduction of libigl via readymade examples spanning the gamut of geometry processing applications and tasks. Attendees will be able to follow along on their laptops. We will explain the core functionality of libigl, how to piece together complex algorithms from library functions, and how to interface to libigl from Python and MATLAB. We will highlight some of libigl's most powerful features: including mesh booleans, quad remeshing, parameterization, and shape deformation. We will conclude with live coding sessions demonstrating libigl's effectiveness and ease-of-use.*

## 1. Introduction

We introduce libigl with a series of self-contained examples. The purpose of each example is to showcase a feature of libigl while applying it to a practical problem in geometry processing. We will present the basic concepts of libigl and introduce a simple mesh viewer that allows to visualize a surface mesh and its attributes. All the tutorial examples are cross-platform and can be compiled on MacOSX, Linux and Windows.

## 2. libigl design principles

libigl is based on 4 main design principles:

1. No complex data types. We mostly use matrices and vectors. This greatly favors code reusability and forces the function authors to expose all the parameters used by the algorithm.
2. Minimal dependencies. We use external libraries only when necessary and we wrap them in a small set of functions.
3. Header-only. It is straight forward to use our library since it is only one additional include directory in your project. (if you are worried about compilation speed, it is also possible to build the library as a static library).
4. Function encapsulation. Every function (including its full implementation) is contained in a pair of .h/.cpp files with the same name of the function.

## 3. Downloading libigl

libigl can be downloaded from our github repository or cloned with git: `git clone https://github.com/libigl/libigl.git`. The core libigl functionality only depends on the C++ Standard Library and Eigen.

## 4. Course Notes and Documentation

The course notes are attached in the additional material in html format, and can be viewed online at the url <https://libigl.github.io/tutorial/>.

## 5. Geometry Processing Courses

The course continues beyond the lecture via two open source graduate-level courses on geometry processing based on libigl (<https://github.com/alecjacobson/geometry-processing>) and (<https://github.com/danielepapano/gp>).