Geometry + Simulation in Python using `pygismo`

SHS Colloquium - 02/03/2022

Hugo Verhelst
h.m.verhelst@tudelft.nl

>1 Year ago...

SHS lecture about open science

The Open-Minded Engineer

On the Relevance of Open-Source Software for Engineering

H.M. (Hugo) Verhelst [1,2,*]

[1] Numerical Analysis, Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
[2] Maritime and Transport Technology, Faculty of Mechanical, Maritime and Materials Engineering (3mE)
Delft University of Technology, The Netherlands

[*] h.m.verhelst@tudelft.nl

11-12-2020

Outline

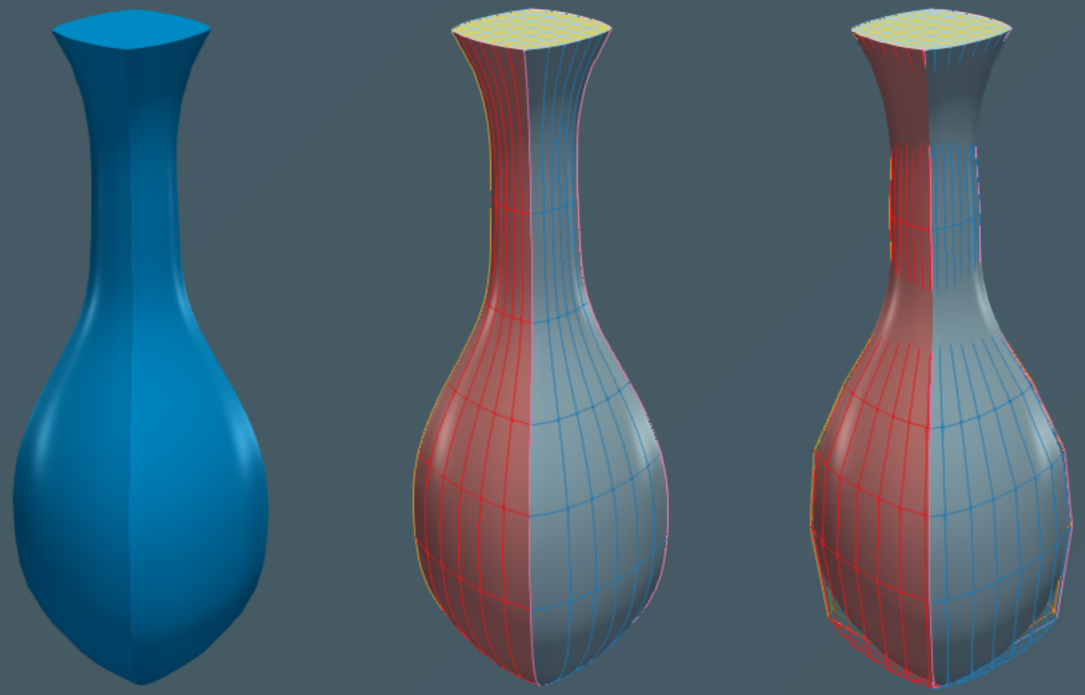
1. `pygismo`: Isogeometric Analysis
2. Geometry + Simulation Modules (G+Smo)
   - Geometry modules
   - Simulation modules
   - External modules
3. `pygismo`: geometric computations
4. `pygismo`: shell analysis
5. What's next?

1. `pygismo` Isogeometric Analysis

Computer-aided Design

1. `pygismo` Isogeometric Analysis

Computer-aided Design

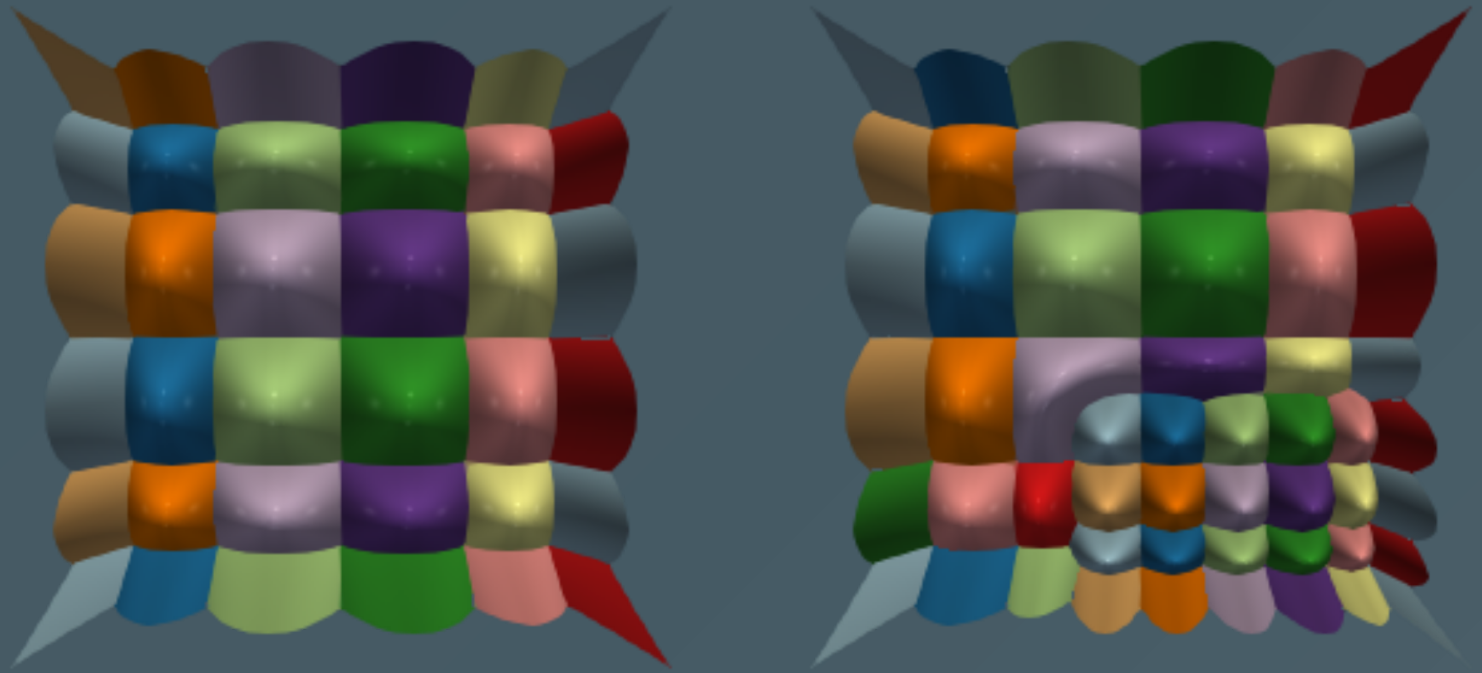


2. G+Smo

*G+Smo (Geometry + Simulation Modules, pronounced "gismo") is an **open-source C++ library** that brings together mathematical tools for geometric design and numerical simulation. (...) The library aims at providing access to **high quality**, **open-source software** to the [forming] isogeometric numerical simulation community and **beyond**.* - https://gismo.github.io

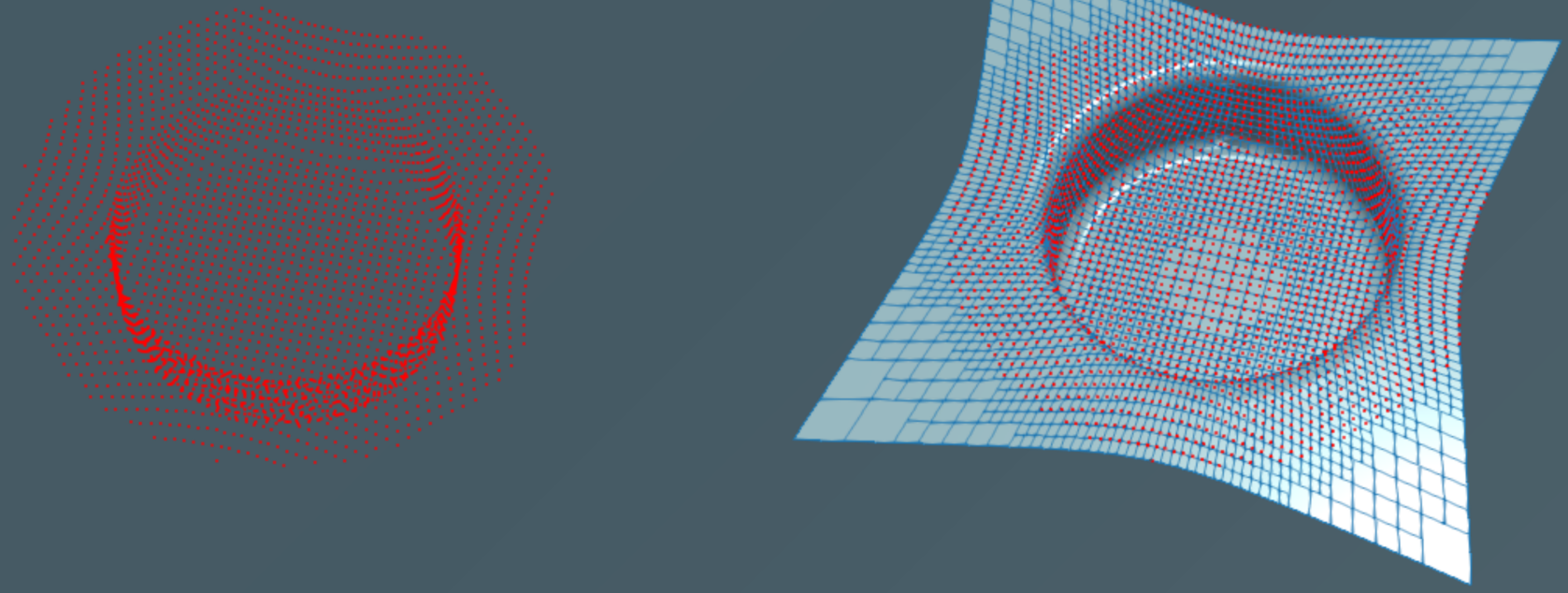
2. G+Smo - Geometry modules

- Supports several spline constructions (e.g. B-Splines, NURBS, (T)HB-Splines, MappedSplines)



2. G+Smo - Geometry modules

- State-of-the-art fitting methods



2. G+Smo - Simulation modules

- Easy-to-use assembly routines

```
gsExprAssembler<> A;
geometryMap G = A.getMap(mp);   // mp is a multi-patch geometry
space       u = A.getSpace(mb); // mb is a multi-patch basis
// assemble the Poisson equation
A.assemble(igrad(u, G) * igrad(u, G).tr() * meas(G))
                                // n-dimensional,
                                // independent of the basis
```

2. G+Smo - Simulation modules

- Easy-to-use assembly routines
- Fast solvers

```
gsSparseSolver<>::PardisoLU solver;
solver.compute(A.matrix());          // Compute LU factorization of matrix
solution = solver.solve(A.rhs());    // Compute solution
```

2. G+Smo - Simulation modules

- Easy-to-use assembly routines
- Fast solvers
- In-house physics modules
  - `gsElasticity` for solid mechanics and FSI @github
  - `gsKLShell` for thin shell mechanics @github
  - `gsStructuralAnalysis` for thin shell mechanics @github

2. G+Smo - External modules

- `Spectra` for sparse eigenvalue computations (e.g. **buckling**)
- `preCICE` multi-physics coupling library `under development`, e.g. for **FSI simulations using G+Smo and OpenFOAM**
- `openNURBS, ...` seamless integration with Rhinoceros' `3dm` file format **for read/write of CAD geometries**
- `IpOpt` for nonlinear optimization, e.g. **for shape optimization**
- `libtorch` C++ implementation of `pytorch` for **Machine Learning**
- `XBraid` multi-grid **parallel-in-time** solvers

  ----And more----

3. `pygismo`: geometric computations

4. `pygismo`: shell analysis

5. What's next? - G+Smo & `pygismo`

- `in progress` Multi-physics with preCICE
  M. Möller (EWI), A. Mantzaflaris (Inria)
- `in progress` Benchmarking unstructured {Exact/Almost/Approximate}-C^1 spline constructions for the biharmonic and Kirchhoff-Love shell equations
  P. Weinmüller (JKU Linz), T. Takacs (JKU Linz), D. Toshniwal (EWI)

5. What's next? - Shell & Solids

- `planned` Binding solids to Python
- `in preparation` goal-oriented mesh adaptivity for shells
  J.H. Den Besten (3mE), M. Möller (EWI)
- `in preparation` Approximate C^1 coupling for isogeometric Kirchhoff-Love shells
  A. Farahat (JKU Linz), M. Kapl (FU Kärnten), J. Kiendl (UBW Munich)
- `in progress` Hyperelastic Tension-Field theory constitutive law
  J.H. Den Besten (3mE), M. Möller (EWI), J. Kiendl (UBW Munich)

5. What's next? - Applications

- `in progress` Application of shell library with `PyNCT` for exploration
  J. Thies (EWI), S. Baars (RUG)
- `in progress` Multi-patch shell analysis with application to fatigue of SOS
  J.H. Den Besten (3mE), M. Möller (EWI)
- `planned` Shape and variable-angle composite optimization of shell structures
  J.H. Den Besten (3mE), M. Möller (EWI), A. Mantzaflaris (Inria), Y. Ji (Dalian UT)

Thanks!

Slides (source: hverhelst.github.io/files/2022_SHS.pdf)