Generation and Refinement of Spherical Volumetric Shell Domains for FEA and IGA applications

Sotiris Gkoulimaris

Europa

- Radius: 1561 km
- Surface Area: 306’000 square km

- Water layers: 120 km
  - Brittle Ice
  - Convecting Ice
  - Liquid Water
- Ice Shell: 0.1 -> 30 km

- Surface Features: 0.1 -> 10x km in width

Representing a Spherical Shell in CAD

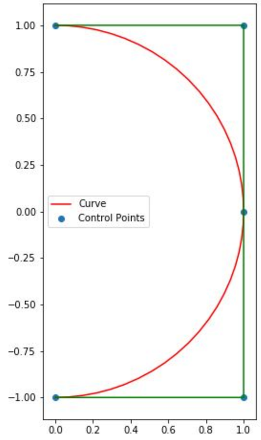
B-Splines
A generalization of the Bezier Curve.
A Spline function of given degree can be expressed as a linear combination of B-splines of that degree.

- NURBS Entities
  - Constructed from B-Splines:
    - Define a *knot vector* (set of parametric space coordinate values)
    - Set a polynomial degree *p*
    - With a given *knot vector* and *p* we can define a set of B-Spline *basis functions*
      - Computed recursively using the Cox-De Boor algorithm.
    - Together with a set of *control points* **P** we construct a piecewise-polynomial B-Spline curve.
  - We can also construct higher dimensional entities like surfaces and volumes
    - We need a number of *knot vectors* equal to the dimensions of the entity.
      - 2 for surfaces, 3 for volumes
  - NURBS expands B-Splines by creating a *rational basis function* by assigning a weight *w* to each *control point.*

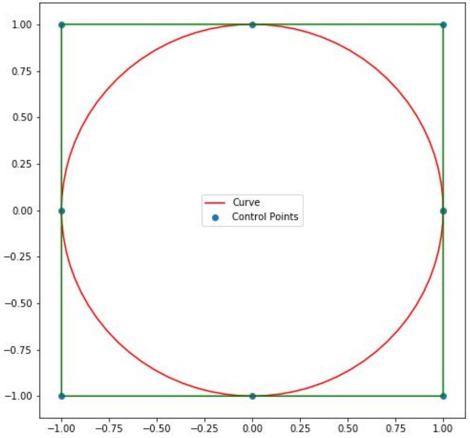
- To define a spherical volumetric shell, we need 3 NURBS curves, 2 to create a sphere and the last to make it a shell.

Representing a Spherical Shell in CAD

- Required parametric curves:
  - Half-Circle (degree 2)
  - Full-Circle (degree 2)
  - Shell Radius (degree 1)

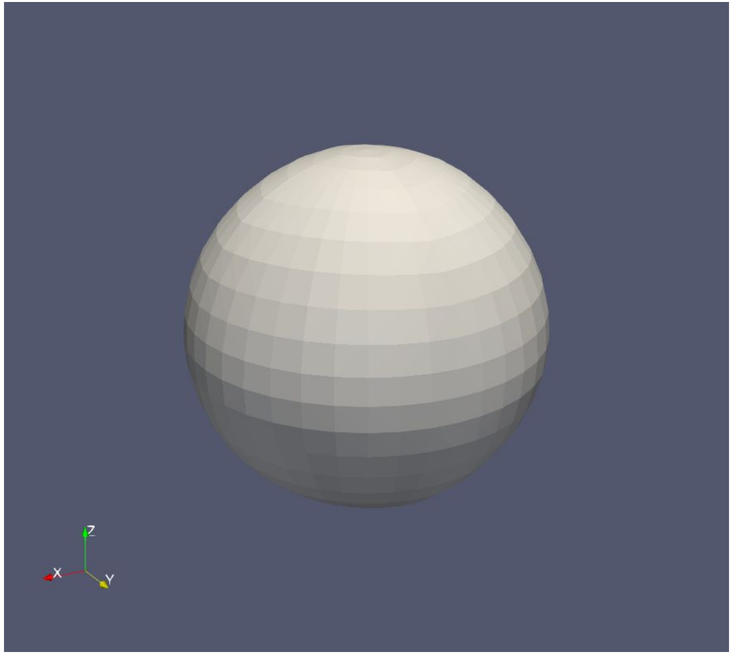


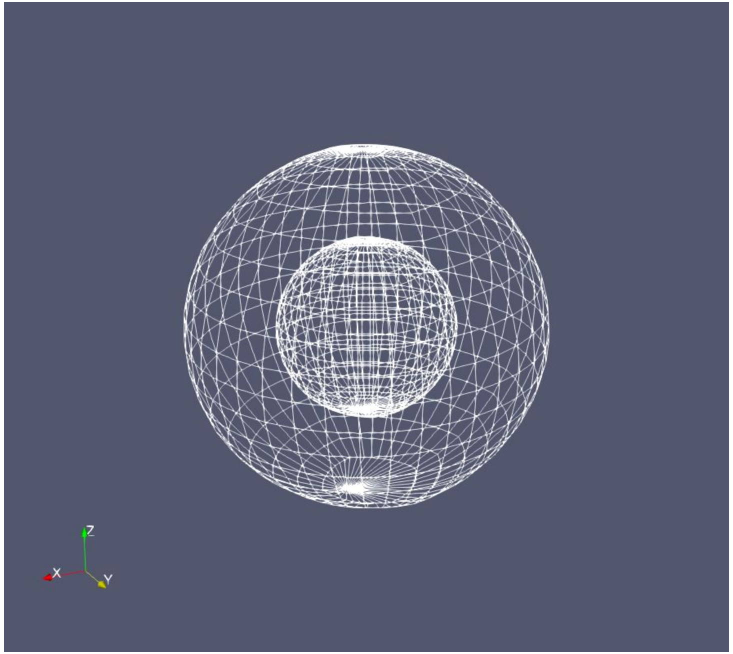
a)



b)

Representing a Spherical Shell in CAD





Limitations of Existing Implementations

- Most common limitations observed:
  - No direct support for NURBS volumes
- Limitations of Blender:
  - Does not expose its Spline primitives
- Limitations of Rhino:
  - Limited implementation of local refinement technologies like T-Splines

A custom tool would be desirable.

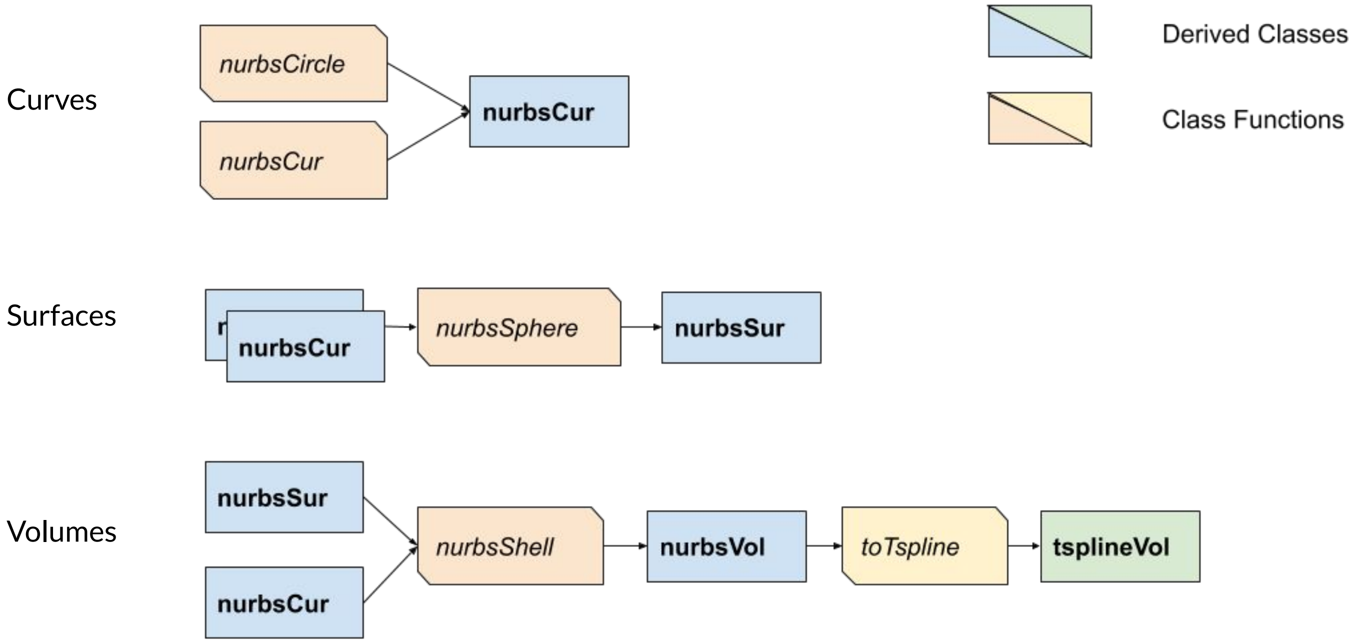
Implementation

- It is written in C++
  - Following the principles of Abstract Class Design (Object Oriented Factory)
  - Added functionality for Derived Classes:
    - Generators (constructors),
    - Operators (e.g. Refinement)
    - Samplers
- The tool constructs NURBS volumes
  - By expanding on existing literature (*NURBS Book*).

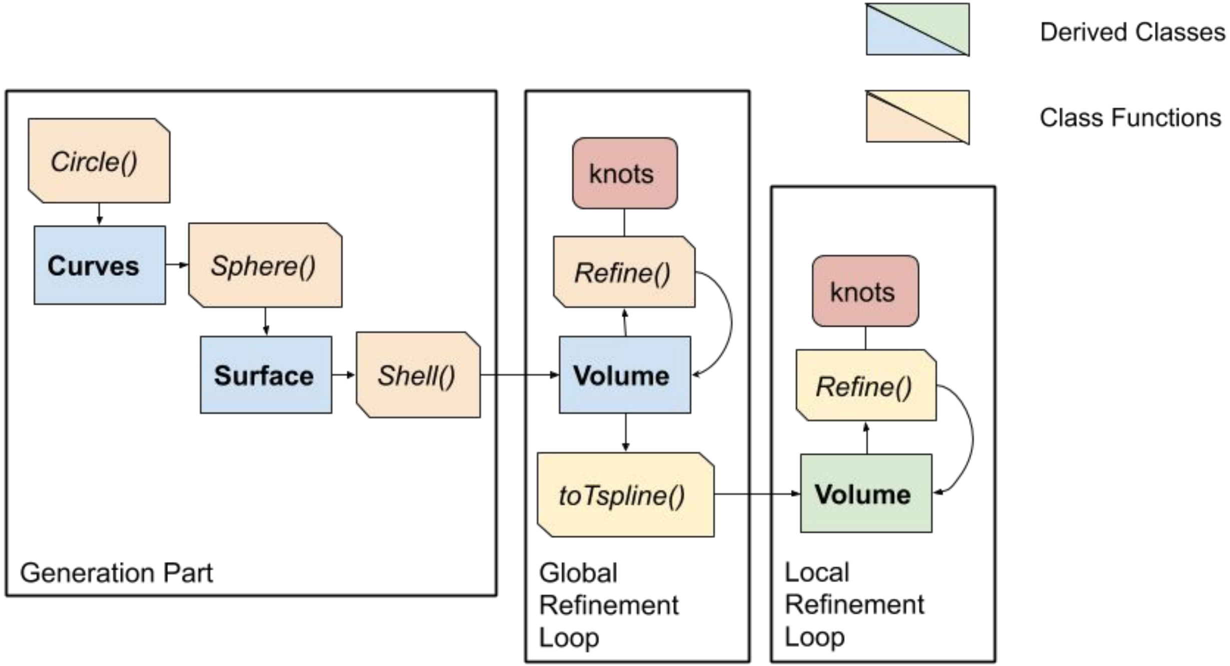
- Implements Global Refinement with Knot Insertion
- Implements Local Refinement with S-Splines

<u>Not a library. Only basic NURBS functionality is implemented</u>

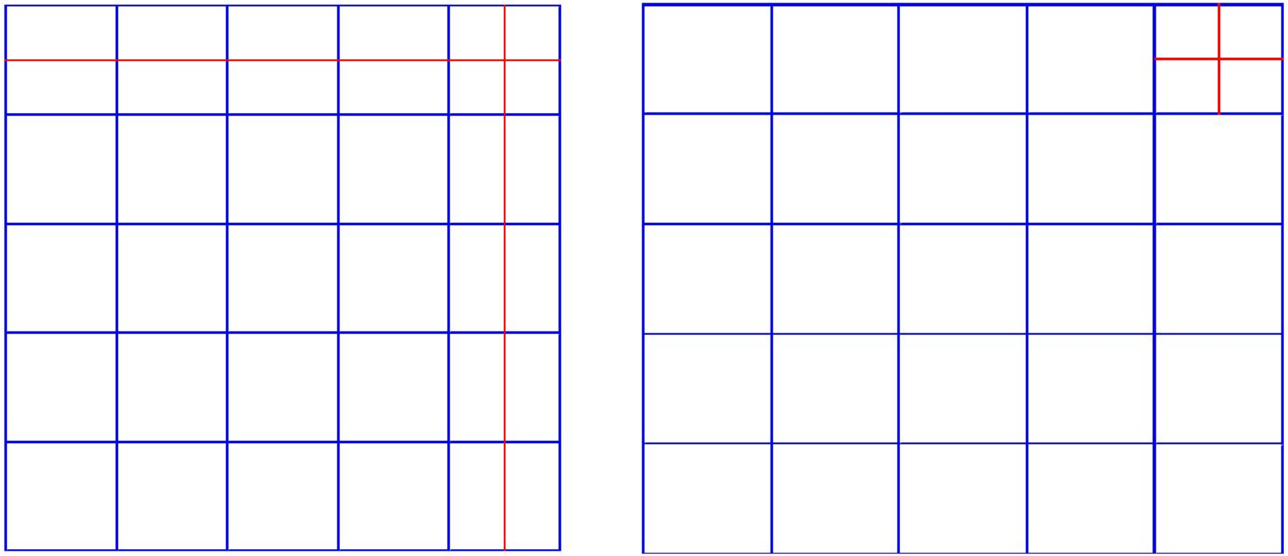
Implementation



Implementation



Global and Local Refinement



a) Global Refinement

b) Local Refinement

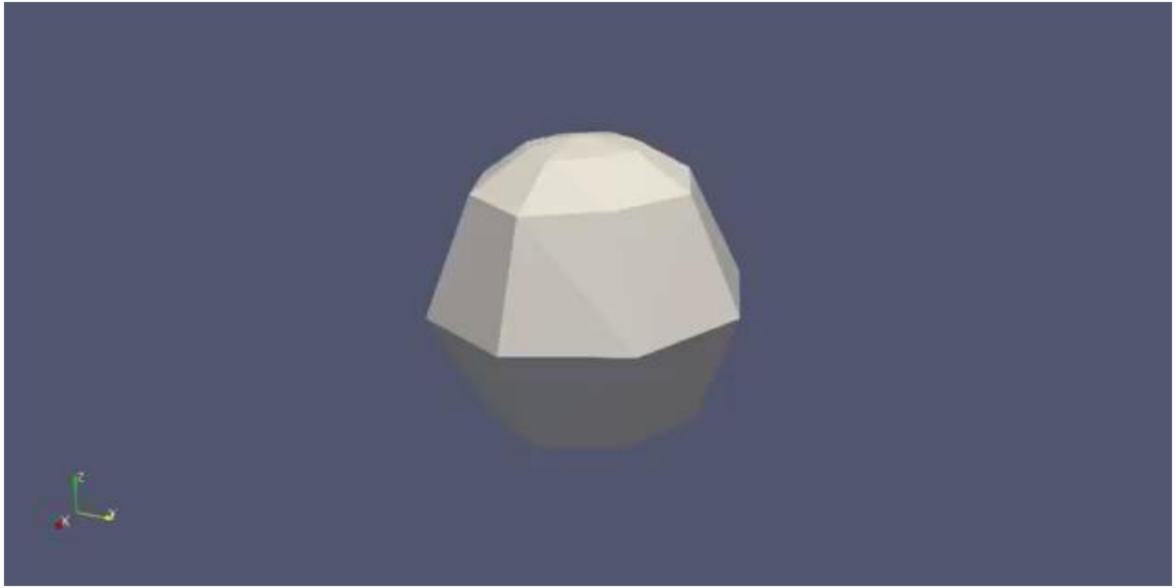
Global and Local Refinement

- Using knot insertion for Global Refinement.
  - It is the analog of h-refinement in FEA. Given new *knots* it constructs a new knot vector that contains the new *knots* and adds new *control points* accordingly.
  - The geometry is left intact!
- S-Splines for Local Refinement.
  - A flavour of T-Splines
  - Requires a shift from global to local perspective:
    - Define a *local knot vector* for every *basis function*
    - Use the *local knot vectors* corresponding to each point in the control to form a *blending function.*
  - S-Splines add a scalar in the definition of the *blending function:*
    - It ensures partition of unity for the rational case.
    - Is computed by the refinement algorithm. (Every *control point* has an initial value of 1)

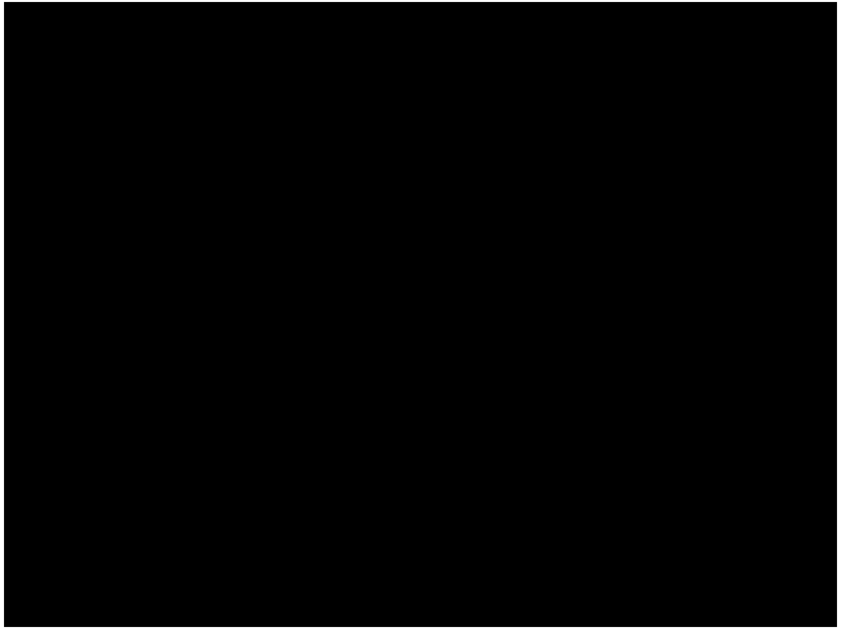
- Implemented for volumes

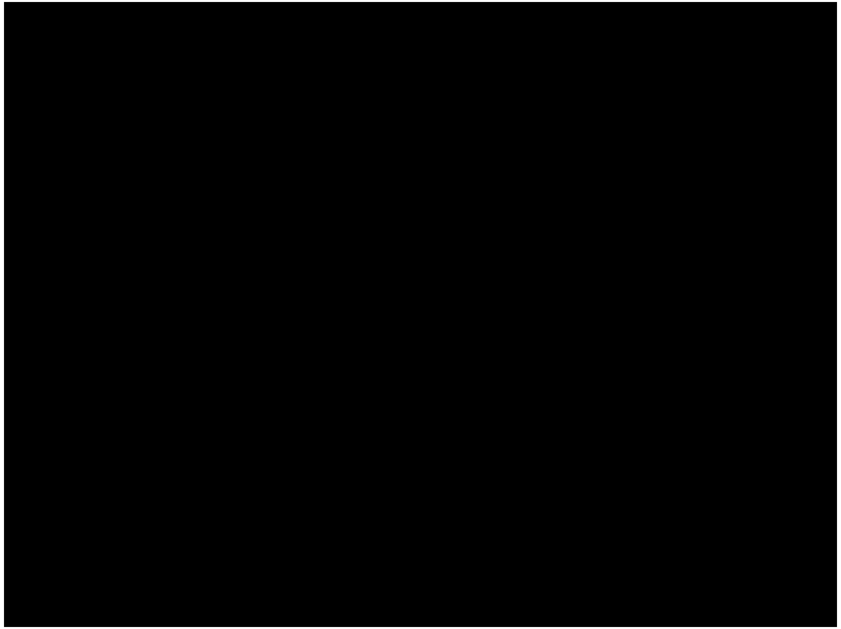
T Spline
Non-uniform (rational) B-splines with T-junctions. Locally Refinable. Control points are added without propagating to other parametric directions.

Refinement Examples: Uniform Domain Refinement

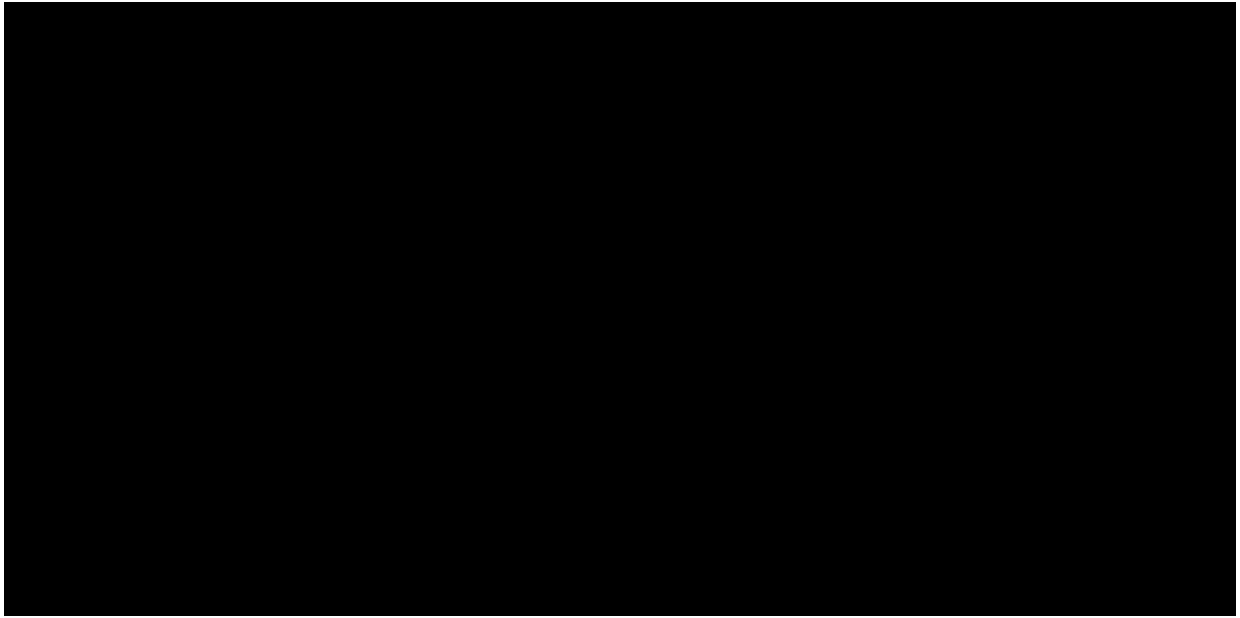


Refinement Examples:
Selective Refinement





Refinement Examples:
Local Refinement



Discussion and Future Work

- The tool successfully constructs and refines volumetric spherical shell domains.
- It can be used for both Finite Element Analysis and Isogeometric Analysis.

- Future work:
  - There are opportunities to parallelise:
    - Evaluating points of a spline is computationally independent.
  - Functionality can be extended for lower dimensional entities:
    - Include global and local refinement for curves and surfaces.
  - S-Spline refinement is “meshy”:
    - Wrapper functions can be implemented to streamline the process.
  - S-Spline local refinement can be more robust:
    - Only Class 1 refinement was implemented.
  - More algorithms for T-Spline refinement can be implemented.

Bibliography

Vance, S. D. et al. (2018) ‘Geophysical Investigations of Habitability in Ice-Covered Ocean Worlds’, Journal of Geophysical Research: Planets, 123(1), pp. 180–205. doi: 10.1002/2017JE005341.

Sederberg, T. W. and Zheng, J. (2001) ‘T-splines and T-NURCCs’.

Li, X. and Sederberg, T. W. (2019) ‘S-splines: A simple surface solution for IGA and CAD’, Computer Methods in Applied Mechanics and Engineering. Elsevier B.V., 350, pp. 664–678. doi: 10.1016/j.cma.2019.03.035.

Hughes, T. J. R., Cottrell, J. A. and Bazilevs, Y. (2009) Isogeomtric Anlysis: Toward Integration of CAD and FEA.

Piegl, L. and Tiller, W. (1997) Monographs in Visual Communication, The NURBS Book. Springer Series.

Thank you!
:)