Testing Literate Programming

TheAuthor

November 2, 2013

Abstract

This document is my first effort on writing some literate programming [Knu84, Knu92], aiming to provide the best embedded documentation for my experimental code, while developing it. The tool choosen is *nuweb*, in my view the simplest multi-language derivation from the Knuth's original WEB and CWEB tools [KL94]. My approach to literate programming is also motivated by the wish for independency of the content from a specific programming language. Actually, I have several candidate languages — including `python`, `haskell`, and `clojure` — to use for writing the reference documentation for my research work, so that *nuweb* gives me the freedom to freely experiment with them, even mixing chunks of code written in different languages in the same document, while providing an execution environment and unit testing in the same time.

Contents

1 Introduction

The intent of this document is writing the reference documentation for an experimental algorithm aiming at generating the *signed* incidence coefficients of the boundary and coboundary matrices for a complex of convex cells generated and handled using the LAR representation scheme recently developed by Vadim Shapiro, Antonio DiCarlo and me.

The main idea of this algorithmic approach is to use a geometric realisation of the cell complex as a simplicial complex, by extending the orientations of cells and the relative signed incidence coefficients from simplices to polytopes, so providing a robust, but yet simple way to compute the operator matrices in the realm of general convex cells.

1.1 Preview

In Section 2 we discuss how to generate a simplicial complex and how to compute its (co)boundary operator matrices. In Section 3 we generate a T-complex of cuboidal cells, and we extract its skeleton complexes of dimension 2 and 1. In Section 4 we finally approach the problem of generating a (co)boundary matrix with signed coefficients, and fully implement an algorithm for its generation in the general case.

2 Signed (co)boundary matrices of a simplicial complex

Importing the library First of all, a modeling application having to deal with simplicial complexes must import the $Simple_X^n$ library, denoted `smplxn` in `python`. The name of the library was firstly used by our CAD Lab at University of Rome "La Sapienza" in years 1987/88 when we started working with dimension-independent simplicial complexes [PBCF93]. This one in turn imports some functions from the `scipy` package and the geometric library `pyplasm` [].

⟨Inport the $Simple_X^n$ library 2⟩ ≡

```
import sys
sys.path.insert(0, 'lib/py/')
from smplxn import *
◇
```

Macro referenced in 3c, 4cd.

2.1 Structured grid

2.1.1 2D example

Generate a simplicial decomposition Then we generate and show a 2D decomposition of the unit square $[0,1]^2 \subset \mathbb{E}^2$ into a 3×3 grid of simplices (triangles, in this case), using the `simplexGrid` function, that returns a pair `(V,FV)`, made by the array `V` of vertices, and by the array `FV` of "faces by vertex" indices, that constitute a *reduced* simplicial LAR of the $[0,1]^2$ domain. The computed `FV` array is then dispayed "exploded", being ex, ey, ez the explosion parameters in the x, y, z coordinate directions, respectively. Notice that the `MKPOLS` pyplasm primitive requires a pair `(V,FV)`, that we call a "model", as input — i.e. a pair made by the array `V` of vertices, and by a zero-based array of array of indices of vertices. Elsewhere in this document we identified such a data structure as CSR(M_d), for some dimension d. Suc notation stands for the Compressed Sparse Row representation of a binary characteristic matrix.

⟨ Generate a simplicial decomposition ot the $[0,1]^2$ domain 3a ⟩ ≡

```
V,FV = simplexGrid([3,3])
VIEW(EXPLODE(1.5,1.5,1.5)(MKPOLS((V,FV))))
◇
```

Macro referenced in 3c.

Extract the $(d-1)$-faces Since the complex is simplicial, we can directly extract its facets (in this case the 1-faces, i.e. its edges) by invoking the `simplexFacets` function on the argument `FV`, so returning the array `EV` of "edges by vertex" indices.

⟨ Extract the edges of the 2D decomposition 3b ⟩ ≡

```
EV = simplexFacets(FV)
ex,ey,ez = 1.5,1.5,1.5
VIEW(EXPLODE(ex,ey,ez)(MKPOLS((V,EV))))
◇
```

Macro referenced in 3c.

Export the executable file We are finally able to generate and output a complete test file, including the visualization expressions. This file can be executed by the `test` target of the `make` command.

`"test/py/test01.py"` 3c ≡

⟨ Inport the $Simple^n_X$ library 2 ⟩
⟨ Generate a simplicial decomposition ot the $[0,1]^2$ domain 3a ⟩
⟨ Extract the edges of the 2D decomposition 3b ⟩
◇

2.1.2 3D example

In this case we produce a $2 \times 2 \times 2$ grid of tetrahedra. The dimension (3D) of the model to be generated is inferred by the presence of 3 parameters in the parameter list of the `simplexGrid` function.

⟨ Generate a simplicial decomposition ot the $[0,1]^3$ domain 4a ⟩ ≡
```
V,CV = simplexGrid([2,2,2])
VIEW(EXPLODE(1.5,1.5,1.5)(MKPOLS((V,CV))))
◇
```
Macro referenced in 4c.

and repeat two times the facet extraction:

⟨ Extract the faces and edges of the 3D decomposition 4b ⟩ ≡
```
FV = simplexFacets(CV)
VIEW(EXPLODE(1.5,1.5,1.5)(MKPOLS((V,FV))))
EV = simplexFacets(FV)
VIEW(EXPLODE(1.5,1.5,1.5)(MKPOLS((V,EV))))
◇
```
Macro referenced in 4c.

and finally export a new test file:

"test/py/test02.py" 4c ≡

⟨ Inport the $Simple_X^n$ library 2 ⟩
⟨ Generate a simplicial decomposition ot the $[0,1]^3$ domain 4a ⟩
⟨ Extract the faces and edges of the 3D decomposition 4b ⟩
◇

2.2 Unstructured grid

2.2.1 2D example

2.2.2 3D example

3 Generation of LAR of a cuboidal complex

4 Algorithm for signed (co)boundary of a polytopal complex

A top-down view of the program structure is as follows:

"test/py/test06.py" 4d ≡

⟨ Inport the $Simple_X^n$ library 2 ⟩
⟨ Vertex array 5a ⟩
⟨ 3D-cells CSR matrix 5b ⟩
⟨ 2D-face CSR matrix 5c ⟩
⟨ 1D-edge CSR matrix 5d ⟩
⟨ main 6 ⟩
◇

$$\alpha, A, \beta, B, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \Phi$$

⟨ Vertex array 5a ⟩ ≡

```
V = [[4,10,0.0*2], [4,10,1.0*2], [4,10,2.0*2], [8,10,0.0*2], [8,10,1.0*2], [8,10,
2.0*2], [14,10,0.0*2], [14,10,1.0*2], [14,10,2.0*2], [8,7,0.0*2], [8,7,1.0*2],
[8,7,2.0*2], [14,7,0.0*2], [14,7,1.0*2], [14,7,2.0*2], [4,4,0.0*2], [4,4,1.0*2],
[4,4,2.0*2], [8,4,0.0*2], [8,4,1.0*2], [8,4,2.0*2], [14,4,0.0*2], [14,4,1.0*2],
[14,4,2.0*2]]
◇
```

Macro referenced in 4d.

⟨ 3D-cells CSR matrix 5b ⟩ ≡

```
CV = [[0,1,3,4,9,10,15,16,18,19],[1,2,4,5,10,11,16,17,19,20],[3,4,6,7,9,10,12,13
],[4,5,7,8,10,11,13,14],[9,10,12,13,18,19,21,22],[10,11,13,14,19,20,22,23]]
◇
```

Macro referenced in 4d.

⟨ 2D-face CSR matrix 5c ⟩ ≡

```
FV = [[0,3,9,15,18],[1,4,10,16,19],[2,5,11,17,20],[3,6,9,12],[4,7,10,13],[5,8,
11,14],[9,12,18,21],[10,13,19,22],[11,14,20,23],[0,1,3,4],[0,1,15,16],[1,2,4,5],
[1,2,16,17],[3,4,6,7],[3,4,9,10],[4,5,7,8],[4,5,10,11],[6,7,12,13],[7,8,13,14],
[9,10,12,13],[9,10,18,19],[10,11,13,14],[10,11,19,20],[12,13,21,22],[13,14,22,23
],[15,16,18,19],[16,17,19,20],[18,19,21,22],[19,20,22,23]]
◇
```

Macro referenced in 4d.

⟨ 1D-edge CSR matrix 5d ⟩ ≡

```
EV = [[0,3],[0,15],[1,4],[1,16],[2,5],[2,17],[3,6],[3,9],[4,7],[4,10],[5,8],[5,
11],[6,12],[7,13],[8,14],[9,12],[9,18],[10,13],[10,19],[11,14],[11,20],[12,21],
[13,22],[14,23],[15,18],[16,19],[17,20],[18,21],[19,22],[20,23],[0,1],[1,2],[3,
4],[4,5],[6,7],[7,8],[9,10],[10,11],[12,13],[13,14],[15,16],[16,17],[18,19],[19,
20],[21,22],[22,23]]
◇
```

Macro referenced in 4d.

⟨ main 6 ⟩ ≡

```
simplices = pivotSimplices(V,CV,d=3)

VIEW(STRUCT([
        MKPOL([V,AA(AA(lambda k : k+1))(simplices),[]]),
        STRUCT(MKPOLS((V,EV)))
        ]))

print"\nsimplexOrientations(simplices) =", simplexOrientations(V,simplices),"\n"
◇
```

Macro referenced in 4d.

5 Index

Index of variables

Index of macros

References

[KL94] Donald Ervin Knuth and Silvio Levy, *The cweb system of structured documentation: Version 3.0*, 1st ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[Knu84] Donald E. Knuth, *Literate programming*, Comput. J. **27** (1984), no. 2, 97–111.

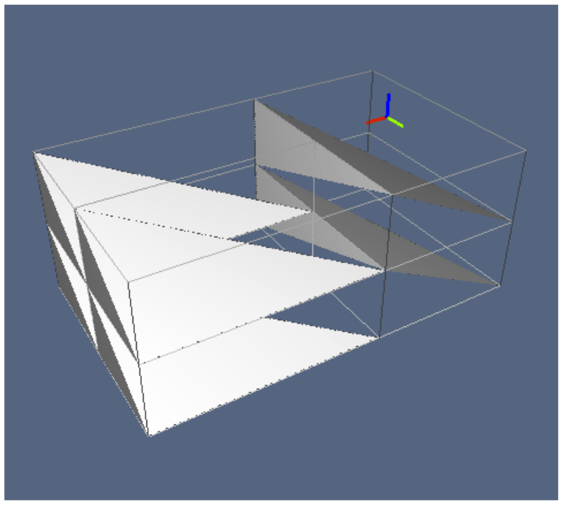

Figure 1: example caption

[Knu92] ———, *Literate programming*, Center for the Study of Language and Information, Stanford, CA, USA, 1992.

[PBCF93] A. Paoluzzi, F. Bernardini, C. Cattani, and V. Ferrucci, *Dimension-independent modeling with simplicial complexes*, ACM Trans. Graph. **12** (1993), no. 1, 56–102.