Parallel & Distibuted Computing: Lecture 8

Alberto Paoluzzi

October 16, 2019

Analysis of Course Project Structure

1. Browsing the project

2. Browsing the package

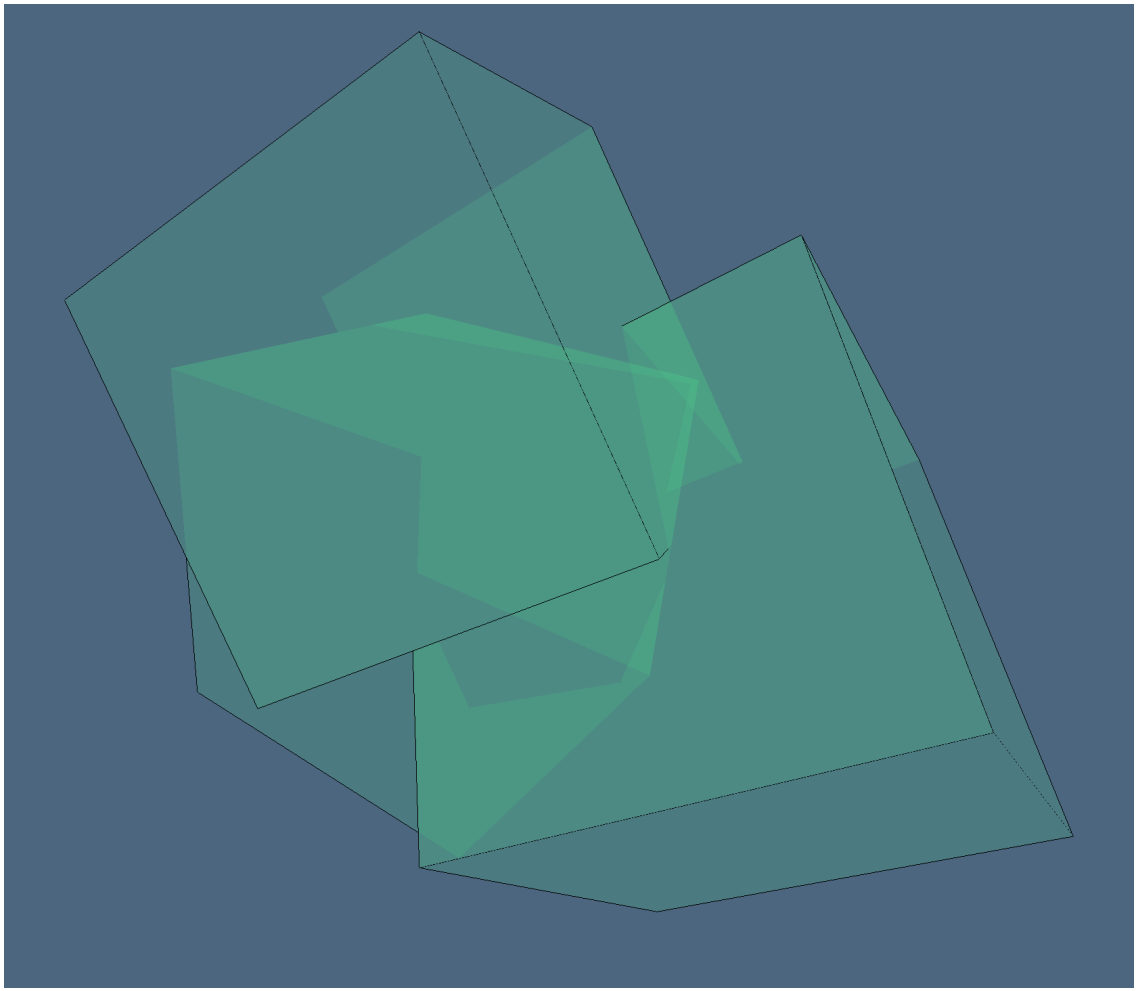
Browsing the project

Start from an example

https://github.com/cvdlab/LinearAlgebraicRepresentation.jl/blob/master/examples/3d/bool3d.jl

```
using LinearAlgebraicRepresentation, ViewerGL, SparseArrays
Lar = LinearAlgebraicRepresentation; GL = ViewerGL
import Base.union

# 3D Boolean example generation
#------------------------------------------------------------------
n,m,p = 1,1,1
V,(VV,EV,FV,CV) = Lar.cuboidGrid([n,m,p],true)
cube = V,FV,EV

# three cubes in "assembly"
assembly = Lar.Struct([ cube,
    Lar.t(.3,.4,.25), Lar.r(pi/5,0,0), Lar.r(0,0,pi/12), cube,
    Lar.t(-.2,.4,-.2), Lar.r(0,pi/5,0), Lar.r(0,pi/12,0), cube
])

V,FV,EV = Lar.struct2lar(assembly)
GL.VIEW([ GL.GLGrid(V,FV), GL.GLFrame ]);

W, (copEV, copFE, copCF), boolmatrix = Lar.bool3d(assembly)
Matrix(boolmatrix)
```

Decompose the example (1/5)



- Start with a `Struct` (note the upper-case)
- `Assembly` (useful to place the solids in world coords)

```
V,FV,EV = Lar.struct2lar(assembly)
GL.VIEW([ GL.GLGrid(V,FV), GL.GLFrame ]);
```

Decompose the example (2/5)

Chain Complex computation

```
julia> W, (copEV, copFE, copCF), boolmatrix = Lar.bool3d(assembly)
```

$$W : C_0 \to \mathbb{R}^3$$

$$\texttt{copEV, copFE, copCF} \mapsto \delta_0, \delta_1, \delta_2$$

$$\texttt{copEV', copFE', copCF'} \mapsto \partial_1, \partial_2, \partial_3$$

```
julia> Matrix(boolmatrix)
8×4 Array{Bool,2}:
  true  false  false  false
 false  false  false   true
 false   true   true  false
 false   true   true   true
 false   true  false  false
 false  false   true  false
 false   true  false   true
 false  false   true   true
```

space atoms on rows, solid terms of `assembly` on columns (first col = outer space)

$\mu : C_0 \to \mathbb{E}^3, (\delta_0, \delta_1, \delta_2)$ ≡ `(geometry, topology) = (W, (EV, FE, CF))`

A GCC allows to transform the (possibly non connected) boundary 2-cycle of a Boolean result (see the example below) into a complete B-rep of the solid result. Note that ordered pairs of letters from `V,E,F,C`, correspond to *Vertices*→*Edges*→*Faces*→*Cells* into the *Column*→*Row* order of matrix maps of operators.

Decompose the example (3/5)

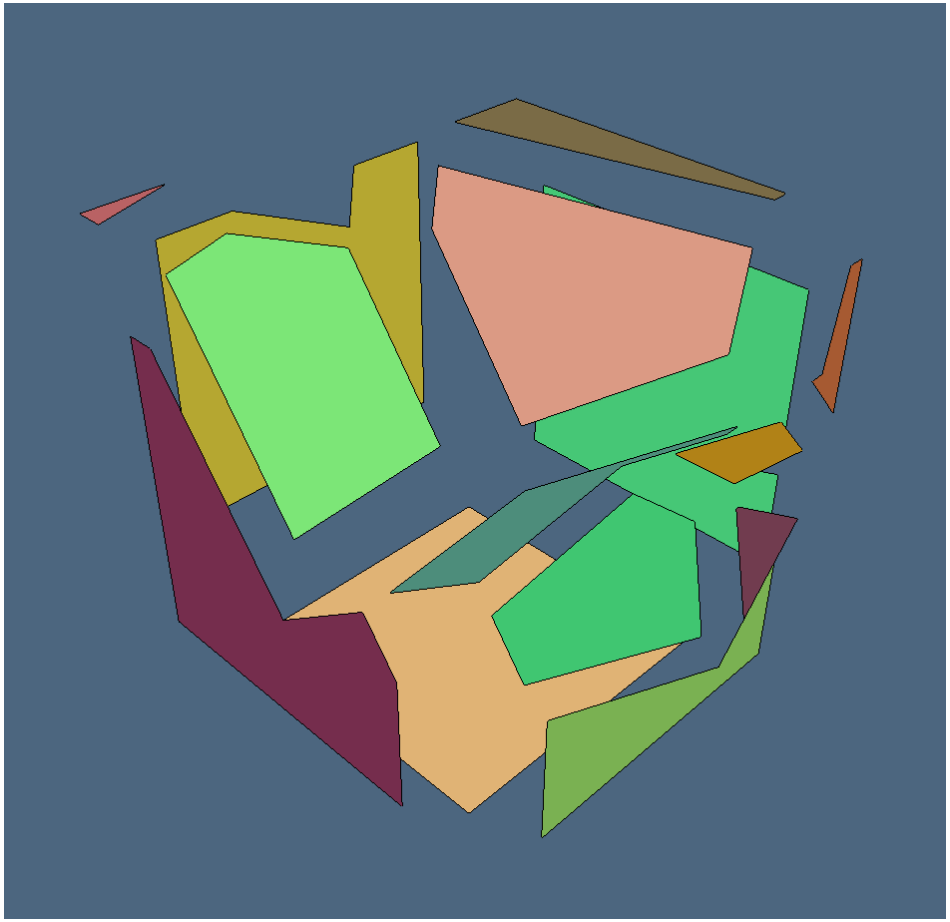
```
A = boolmatrix[:,2]
B = boolmatrix[:,3]
C = boolmatrix[:,4]
AorB = A .| B
AandB = A .& B
AorBorC = A .| B .| C
AorBorC = .|(A, B, C)
AandBandC = A .& B .& C
AandBandC = .&(A, B, C)
AminBminC = .&(A, .!B, .!C) # A - B - C
```

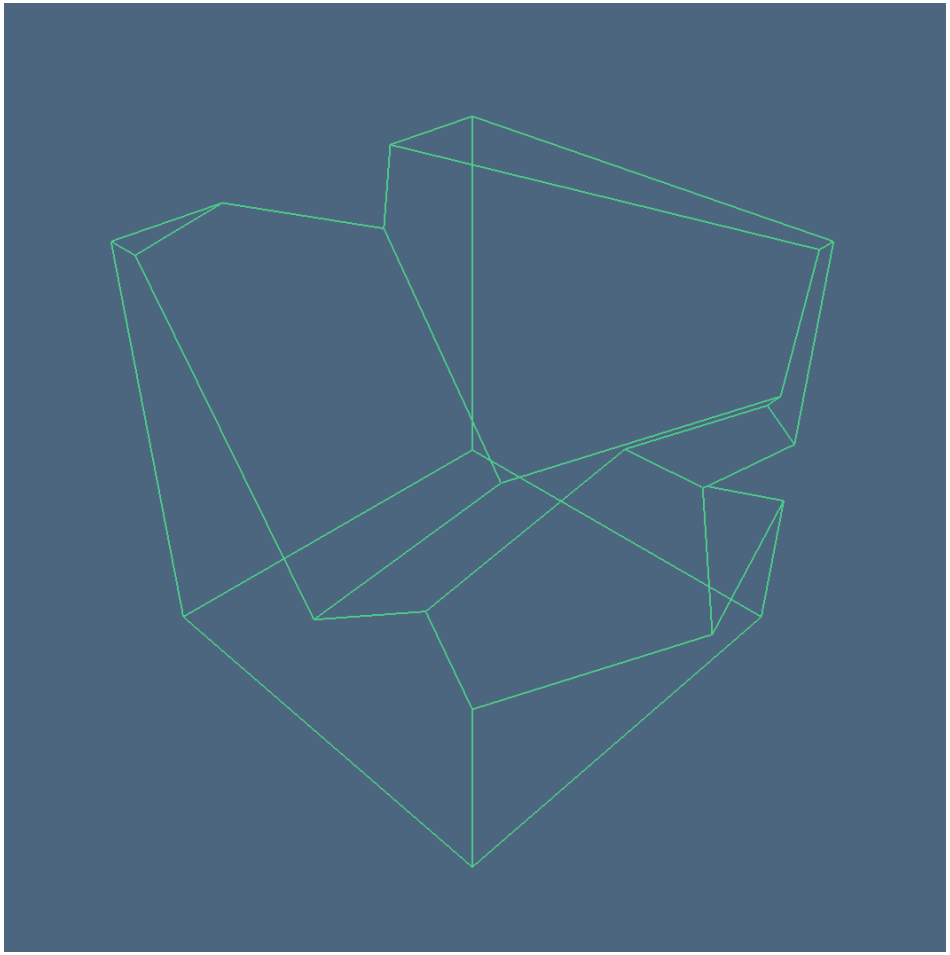
Some examples

Decompose the example (4/5)

```
unione = Matrix(copCF)' * Int.(AorBorC) # coord vector of Faces
intersection = Matrix(copCF)' * Int.(AandBandC) # coord vector of Faces
difference = Matrix(copCF)' * Int.(AminBminC) # coord vector of Faces

V,CVs,FVs,EVs = Lar.pols2tria(W, copEV, copFE, copCF) # whole assembly
Xs = difference
V,CVs,FVs,EVs = Lar.pols2tria(W, copEV, copFE, copCF, Xs) # part of assembly
```

Decompose the example (5/5)





Follow the algorithms

```
.
├─ function spaceindex(point3d::Array{Float64,1})::Function
│  └─ function spaceindex0(model::Lar.LAR)::Array{Int,1}
│     ├─ boundingbox
│     ├─ coordintervals
│     └─ IntervalTrees.IntervalMap
│
├─ function rayintersection(point3d)
│  └─ function rayintersection0(V, FV, face::Int)
│     └─ LinearAlgebra.normalize
│
├─ function planemap(V,copEV,copFE,face)
│  ├─ Lar.vcycle
│  └─ function planemap0(point)
│
├─function settestpoints(V,EV,FV,Fs, copEV,copFE)
│  └─ LinearAlgebra.normalize
│
├─ function testinternalpoint(V,EV,FV)
│  ├─ Lar.lar2cop
│  └─ function testinternalpoint0(testpoint)
│     ├─ spaceindex
│     ├─ rayintersection
│     ├─ planemap
│     ├─ Lar.pointInPolygonClassification
│     └─ classify
```

```
├─ function getinternalpoint(V,EV,FV,Fs, copEV,copFE)
│  ├─ settestpoints
│  ├─ rayintersection
│  ├─ planemap
│  ├─ Lar.pointInPolygonClassification
│  └─ classify
│
├─ function chainbasis2solids(V,copEV,copFE,copCF)
│
├─ function internalpoints(V,copEV,copFE,copCF)
│  ├─ chainbasis2solids
│  └─ getinternalpoint
│
├─ function bool3d(assembly)
│  ├─ Lar.struct2lar
│  ├─ Lar.coboundary_0
│  ├─ Lar.coboundary_1
│  ├─ Lar.Arrangement.spatial_arrangement
│  ├─ internalpoints
│  ├─ Lar.evalStruct
│  ├─ cumsum
│  ├─ testinternalpoint
│  └─ containmenttest
│
└─ function bool3d(expr, assembly)
```

Browsing the package

Source directory

Test directory

```
(v1.1) pkg> test LinearAlgebraicRepresentation

Updating registry at `~/.julia/registries/General`
Updating git-repo `https://github.com/JuliaRegistries/
    General.git`


 Testing LinearAlgebraicRepresentation
Resolving package versions...
[9e28174c] BinDeps v0.8.10
[34da2185] Compat v2.1.0
[864edb3b] DataStructures v0.15.0
[b4f34e82] Distances v0.8.0
[524e6230] IntervalTrees v1.0.0
[95167b0c] LinearAlgebraicRepresentation v0.1.0
       [`~/Documents/dev/LinearAlgebraicRepresentation.jl`]
[b8a86587] NearestNeighbors v0.4.3
[bac558e1] OrderedCollections v1.1.0
[b77e0a4c] InteractiveUtils  [`@stdlib/InteractiveUtils`]
...       ...        ...
[9abbd945] Profile  [`@stdlib/Profile`]
[3fa0cd96] REPL  [`@stdlib/REPL`]
[9a3f8284] Random  [`@stdlib/Random`]
[ea8e919c] SHA  [`@stdlib/SHA`]
[9e88b42a] Serialization  [`@stdlib/Serialization`]
[1a1011a3] SharedArrays  [`@stdlib/SharedArrays`]
[6462fe0b] Sockets  [`@stdlib/Sockets`]
[2f01184e] SparseArrays  [`@stdlib/SparseArrays`]
[10745b16] Statistics  [`@stdlib/Statistics`]
[8dfed614] Test  [`@stdlib/Test`]
[cf7118a7] UUIDs  [`@stdlib/UUIDs`]
```

```
Test Summary:                | Pass  Total
Bounding boxes building test |    1      1
Test Summary:                    | Pass  Total
Bounding boxes containment test |    4      4
Test Summary:              | Pass  Total
Face area calculation test |    1      1
Test Summary:              | Pass  Total
Edge fragmentation tests |    3      3
Test Summary:            | Pass  Total
merge_vertices test set |    2      2
Test Summary:                    | Pass  Total
biconnected_components test set |    1      1
...       ...        ...
Test Summary:    | Pass  Total
Index2addr Tests |   10     10
Test Summary:     | Pass  Total
LarCellProd Tests |   13     13
Test Summary:       | Pass  Total
FilterByOrder Tests |   24     24
...       ...        ...
[10, 10, 10]  |    1      1
Test Summary:    | Pass  Total
LarCuboids Tests |   59     59
Test Summary:         | Pass  Total
LarModelProduct Tests |   15     15
Test Summary:           | Pass  Total
interface.jl file Tests |   35     35
```

```
Test Summary: | Pass  Total
checkStruct   |    2      2
Test Summary: | Pass  Total
2D  1         |    6      6
Test Summary: | Pass  Total
3D  4         |    6      6
Test Summary: | Pass  Total
cuboid Tests  |    6      6
Test Summary: | Pass  Total
traversal     |    3      3
Test Summary: | Pass  Total
Struct Tests  |    3      3
Test Summary:    | Pass  Total
removeDups Tests |    4      4
...       ...        ...
Test Summary: | Pass  Total
torus         |    4      4
Test Summary:     | Pass  Total
Integration Tests |   76     76
Test Summary: | Pass  Total
simplexn      |   15     15
Test Summary:         | Pass  Total
2D containment tests |   20     20
Test Summary:           | Pass  Total
Biconnected components |   11     11
Test Summary:                  | Pass
Refactoring spaceindex tests |   20
Test Summary:                | Pass  Tota
Refactoring fragmentlines |    7
Test Summary:                | Pass  Total
```

Documentation directory