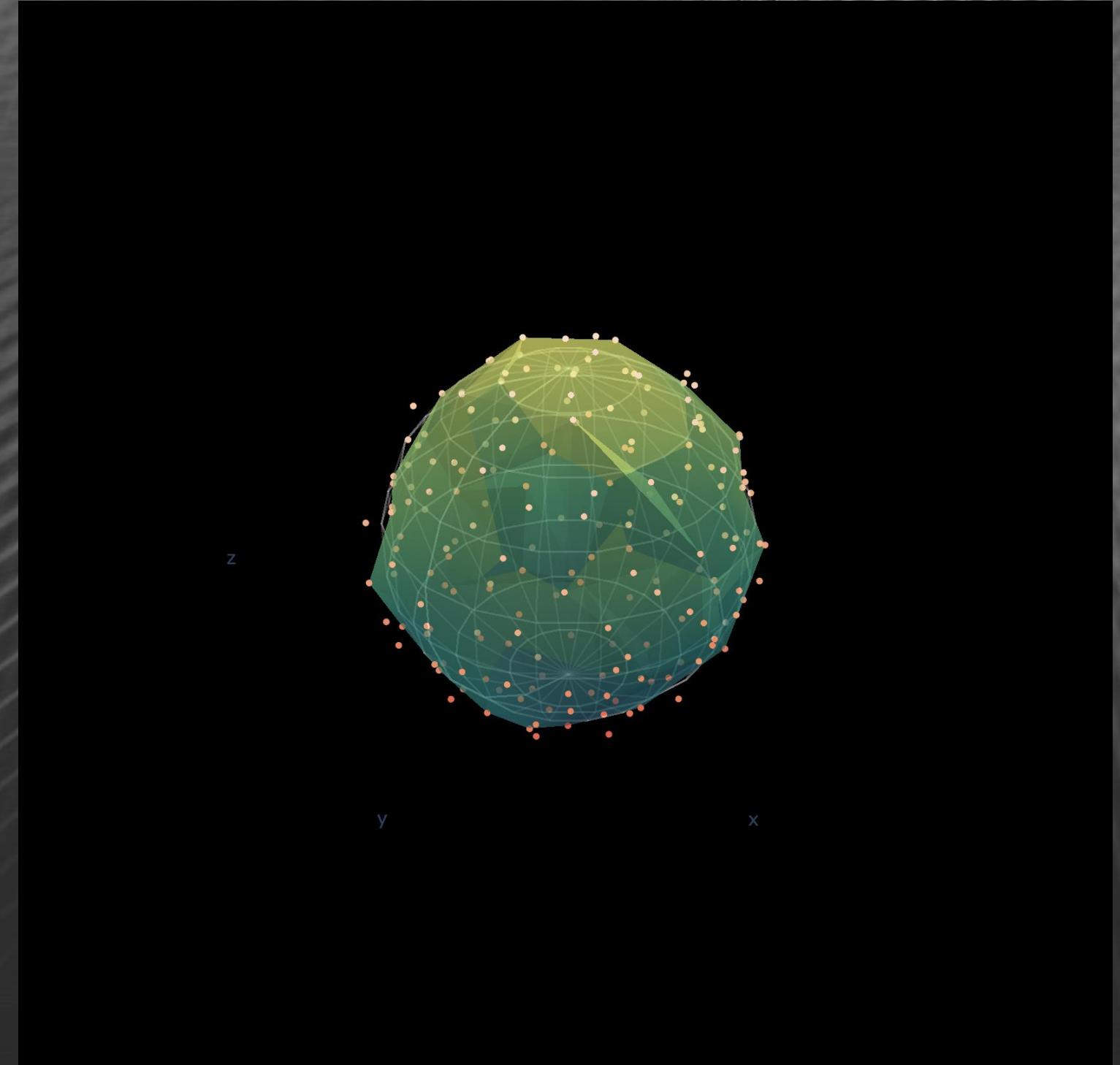




Pacific
Northwest
NATIONAL LABORATORY

Open Applied Topology

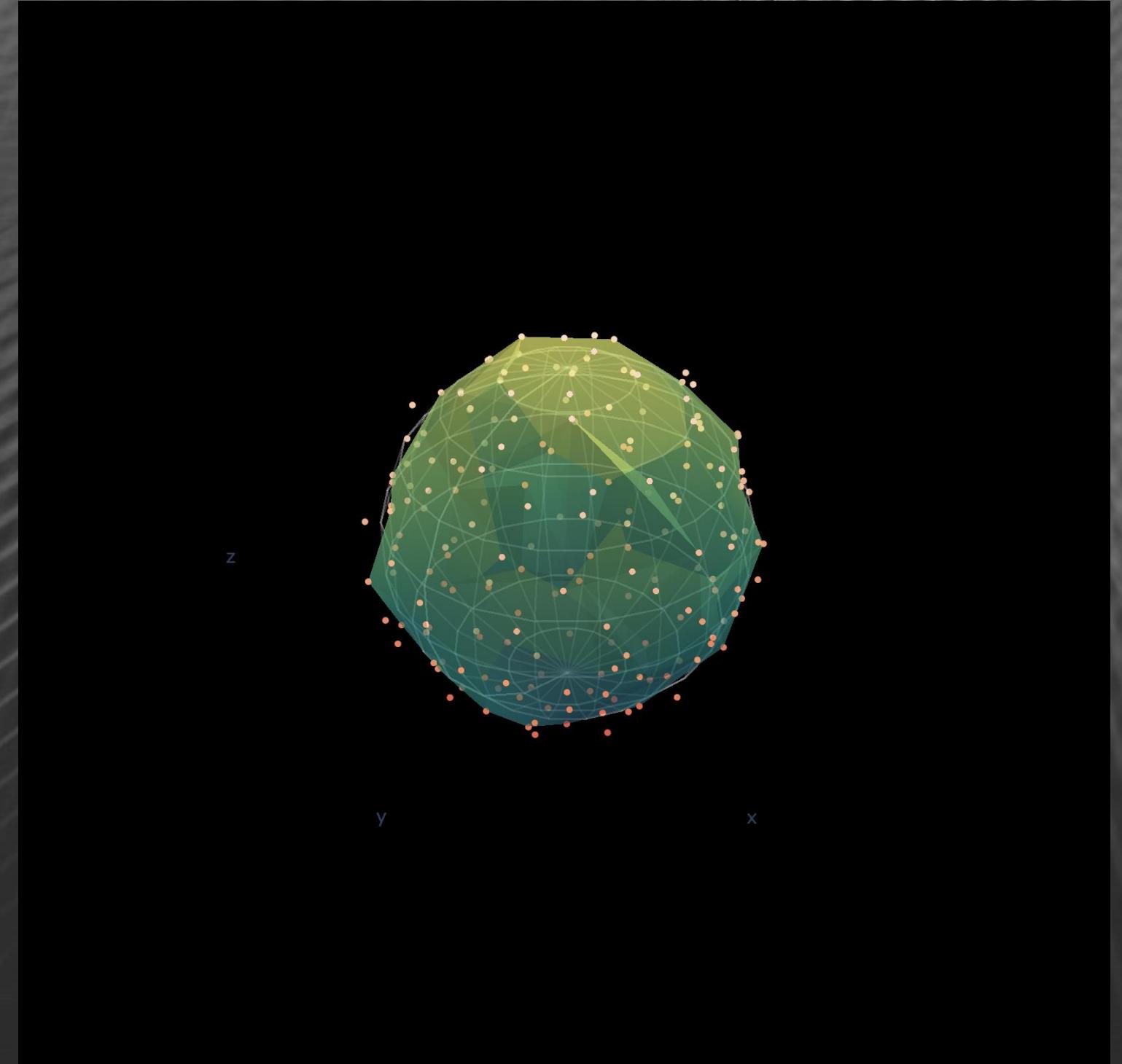
Gregory Henselman-Petrusek Roek
Pacific Northwest National Laboratory





Pacific
Northwest
NATIONAL LABORATORY

Fast, user-friendly matrix algebra for applied topology



U.S. DEPARTMENT OF
ENERGY **BATTELLE**

PNNL is operated by Battelle for the U.S. Department of Energy

Applied Topology

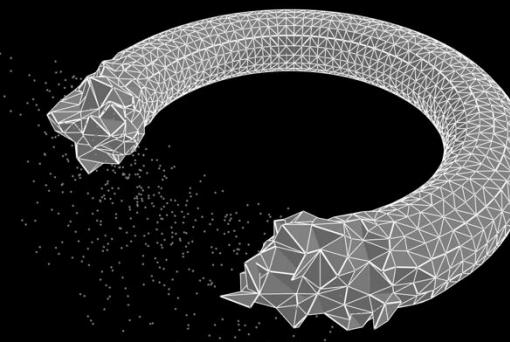
- Network science
- Cybersecurity
- Biology
- Genomics
- Medicine
- Dynamics
- Signal processing
- Robotics
- Chemistry
- Material Science
- Artificial Intelligence
- Image analysis
- Neuroscience
- Nonlinear dimension reduction
- Additive manufacturing

Applied Topology Pipeline

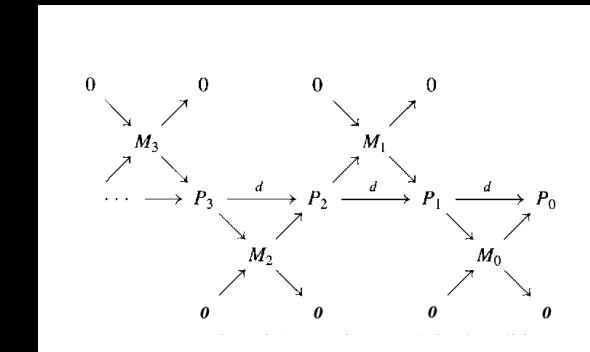
Data



Simplicial Complex



Homological Algebra

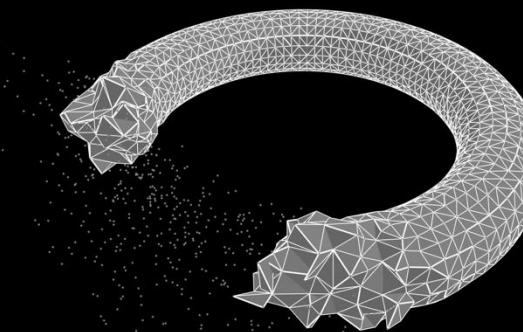


Applied Topology Pipeline

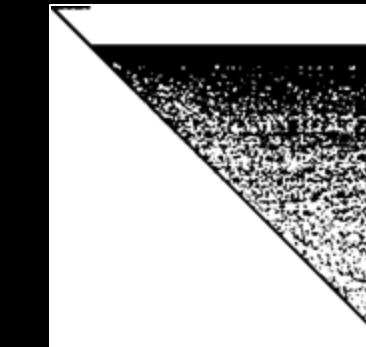
Data



Simplicial Complex

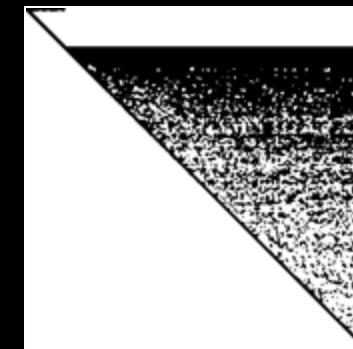


Matrix Algebra



Examples

Matrix Algebra

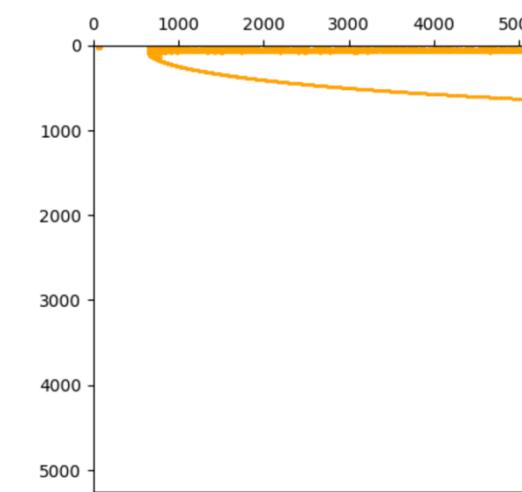
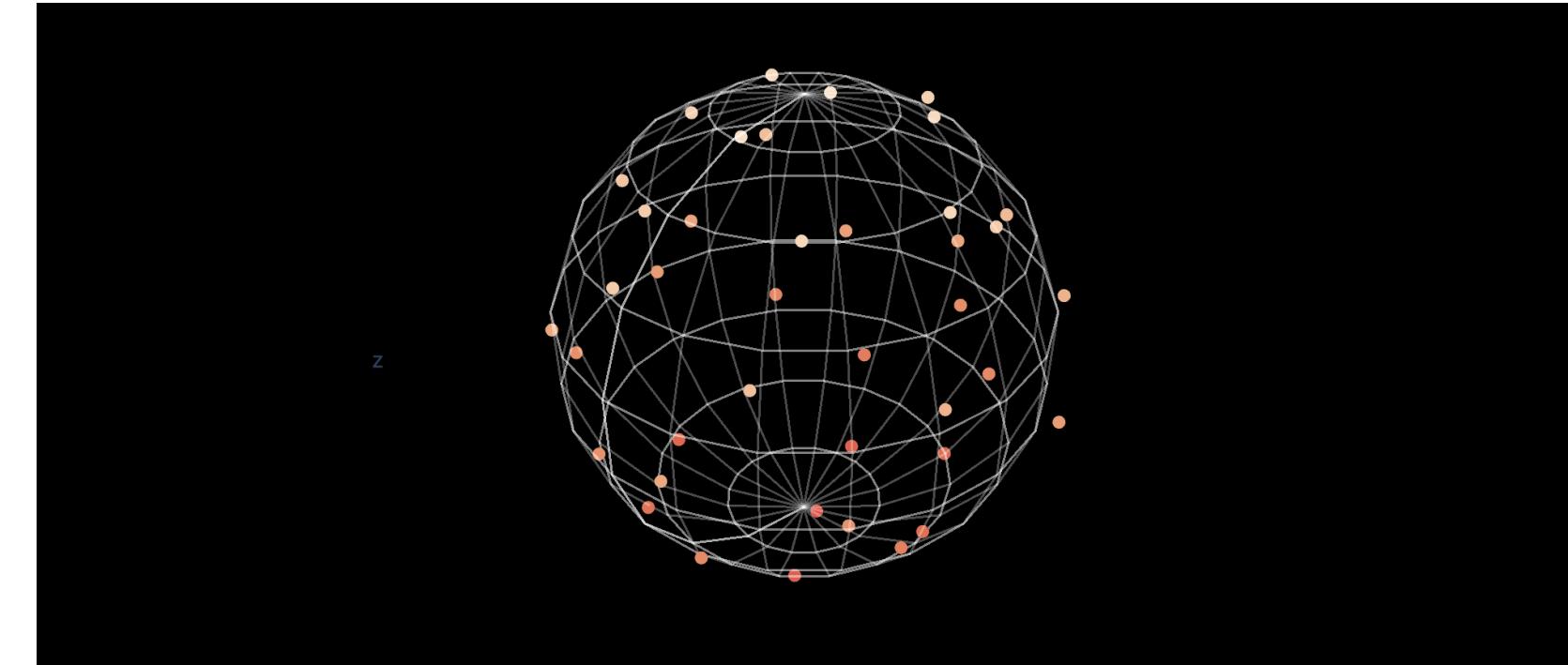




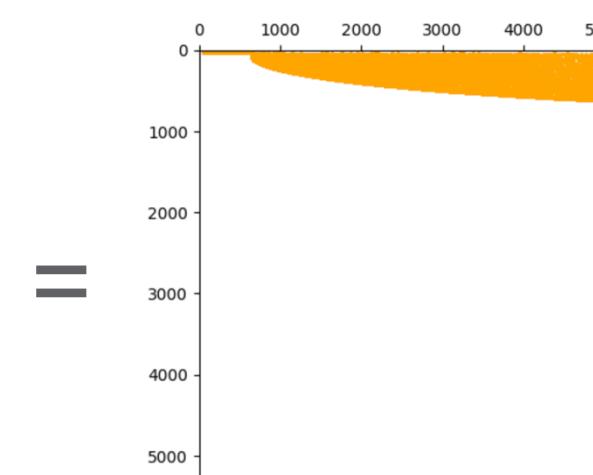
Pacific
Northwest
NATIONAL LABORATORY

Matrix operations

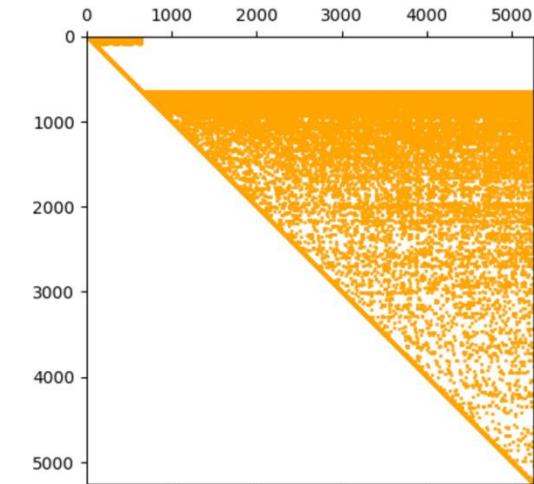
- Multiplication
- Inversion
- Factorization



R



D



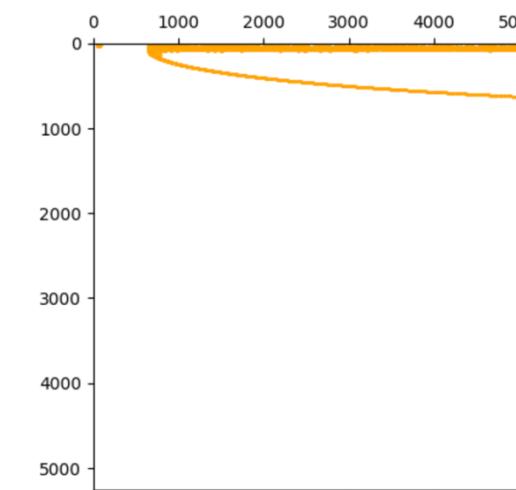
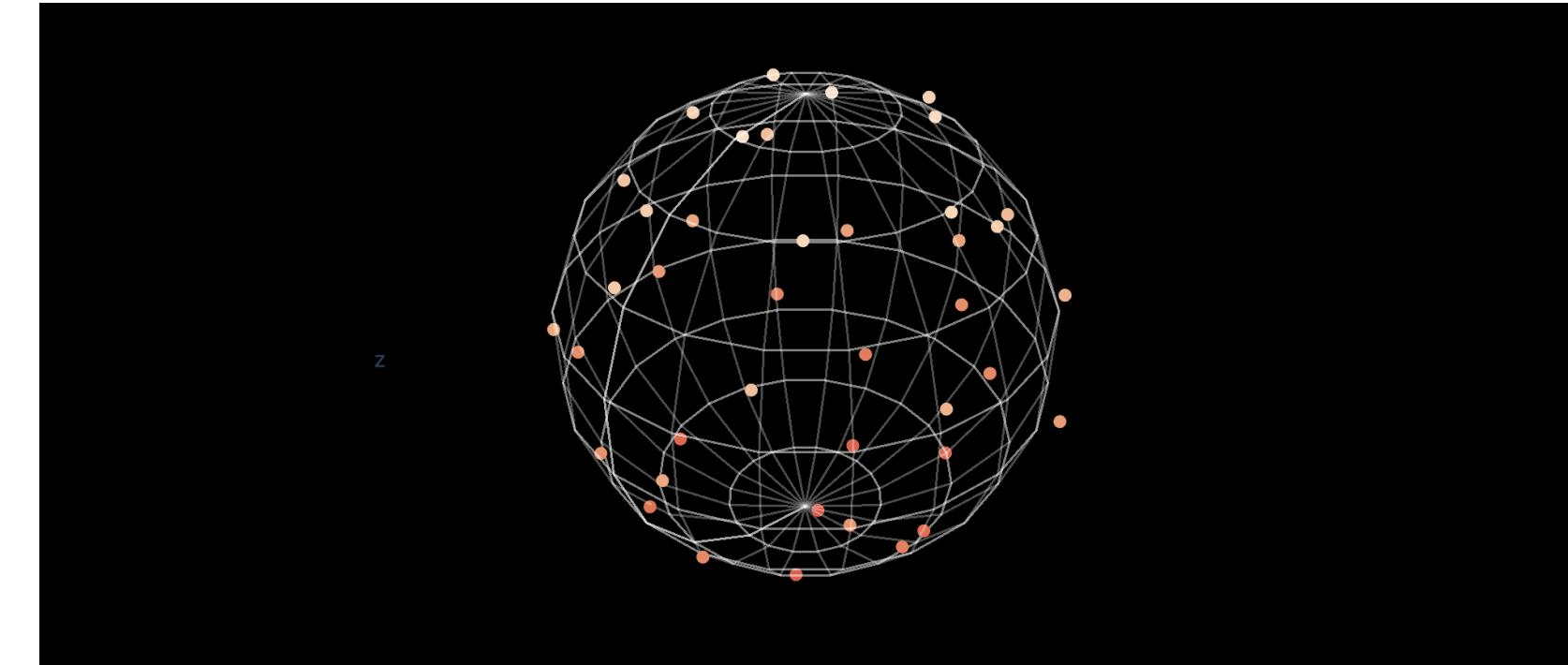
V



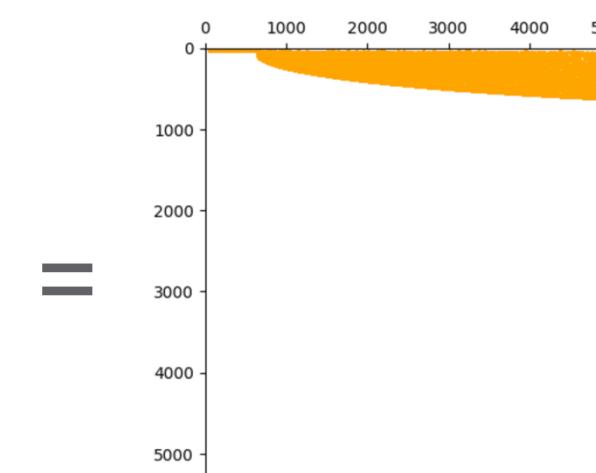
Pacific
Northwest
NATIONAL LABORATORY

Vector operations

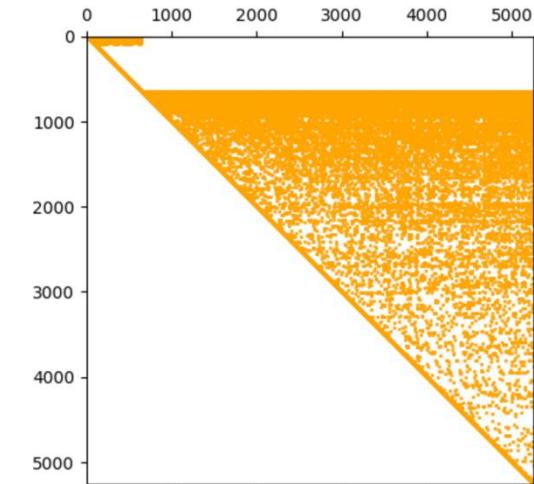
- Linear combination
- Matrix-multiplication
- Optimization
- Back-substitution
(triangular solve)



R



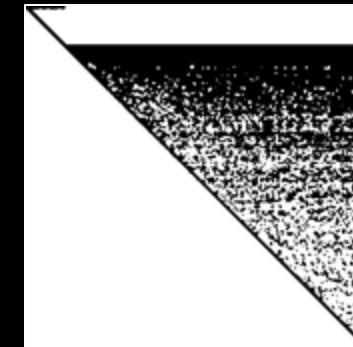
D



V

Challenges

Matrix Algebra



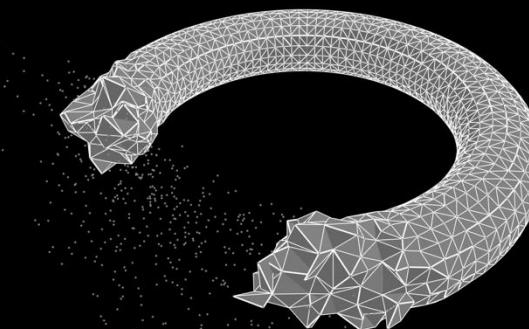
Large Matrices

Data



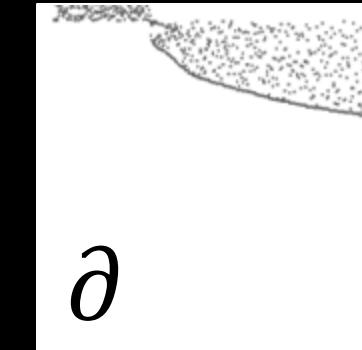
$10k$
Points

Simplicial Complex



$167B$
Triangles

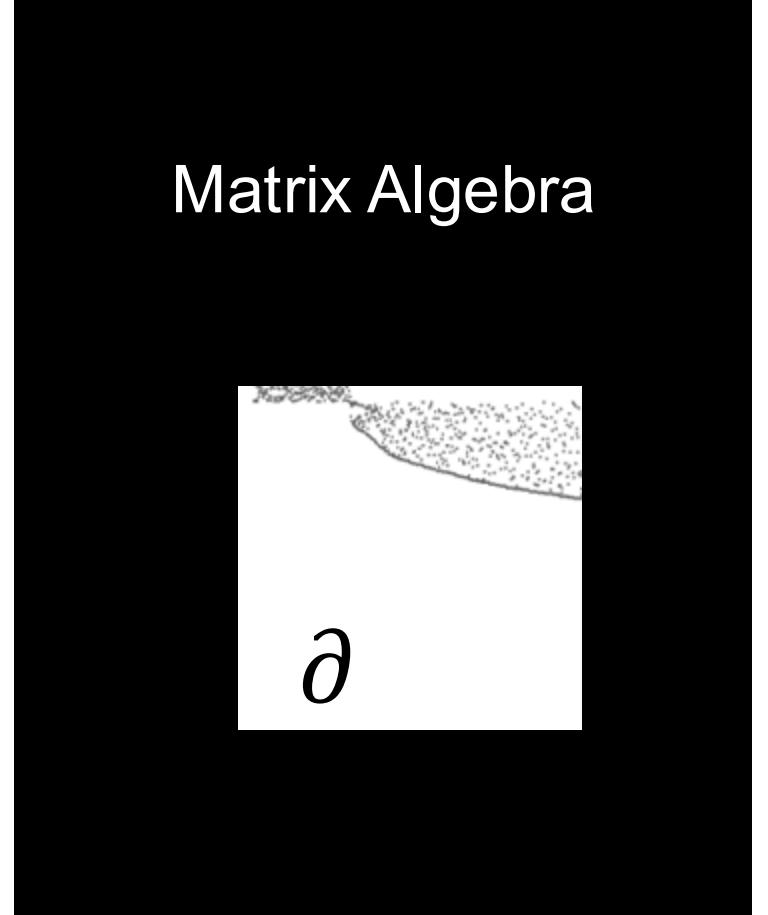
Matrix Algebra



$167B$
Rows and Columns

Large Matrices

5 terabytes to store a unique ID number for each simplex



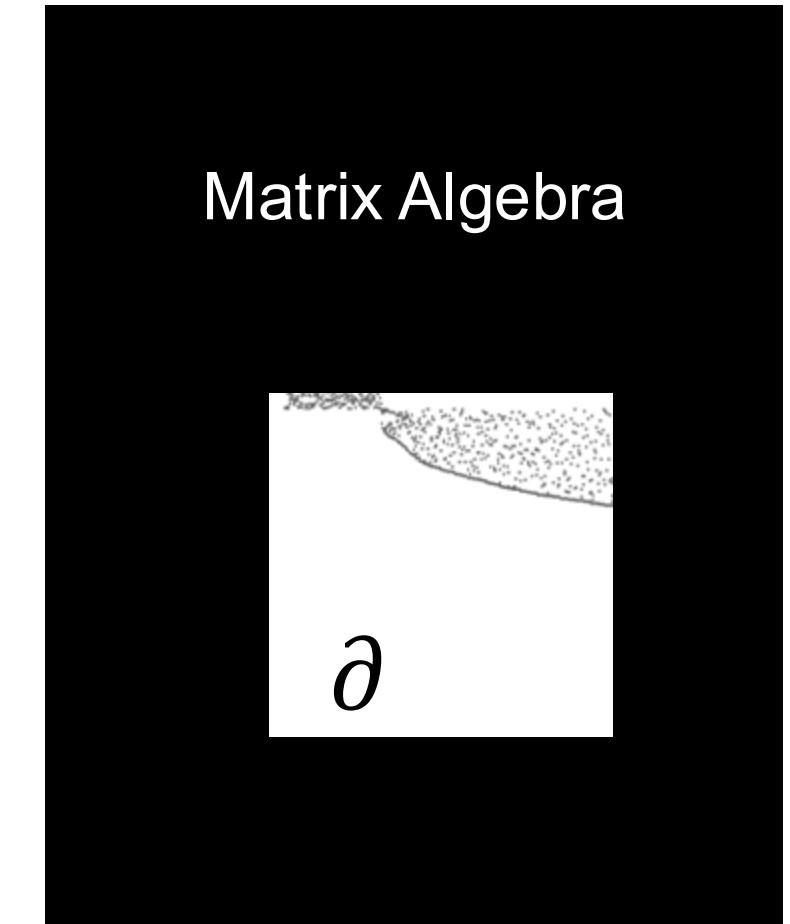
Matrix Algebra

$167B$
Rows and Columns

Large Matrices

Far too large for state-of-the-art
matrix algebra libraries
(BLAS, LAPACK, Eigen)

Matrix Algebra

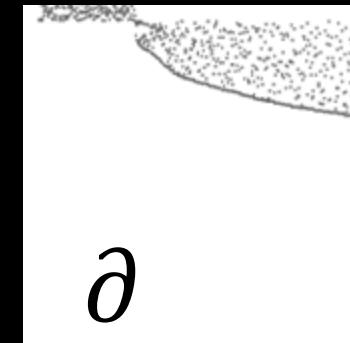


$167B$
Rows and Columns

Floating point arithmetic

Few high-performance linear algebra solvers calculate rank (key to homology computation) due to **numerical error**

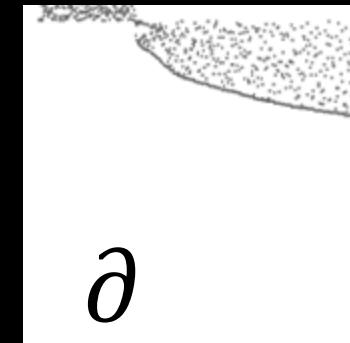
Matrix Algebra



Floating point arithmetic

Incompatible with coefficient rings
other than \mathbb{R} or \mathbb{Q}

Matrix Algebra



Human interface

Code must be human readable and **accessible to mathematicians**

∂ .

simplex	coefficient
[7, 17, 31, 34]	1
[4, 7, 17, 34]	-1
[3, 7, 17, 31]	1
[0, 4, 7, 17]	-1
[0, 3, 7, 17]	1
[7, 18, 31, 34]	-1
[4, 7, 18, 34]	1
[17, 24, 31, 34]	1
[3, 17, 24, 31]	1
[4, 9, 18, 34]	-1
[4, 8, 17, 34]	1
[4, 8, 9, 34]	-1



simplex	coefficient
[36, 37, 39]	1
[34, 36, 37]	-1
[18, 28, 33]	-1
[18, 23, 33]	1
[2, 4, 8]	-1
[1, 2, 4]	1
[9, 22, 23]	1
[9, 12, 23]	-1
[30, 34, 37]	-1
[7, 9, 12]	1
[27, 32, 34]	-1

Human interface

∂ .

simplex	coefficient
[7, 17, 31, 34]	1
[4, 7, 17, 34]	-1
[3, 7, 17, 31]	1
[0, 4, 7, 17]	-1
[0, 3, 7, 17]	1
[7, 18, 31, 34]	-1
[4, 7, 18, 34]	1
[17, 24, 31, 34]	1
[3, 17, 24, 31]	1
[4, 9, 18, 34]	-1
[4, 8, 17, 34]	1
[4, 8, 9, 34]	-1

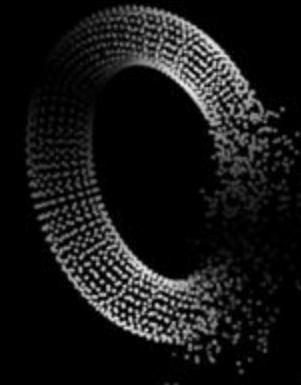


simplex	coefficient
[36, 37, 39]	1
[34, 36, 37]	-1
[18, 28, 33]	-1
[18, 23, 33]	1
[2, 4, 8]	-1
[1, 2, 4]	1
[9, 22, 23]	1
[9, 12, 23]	-1
[30, 34, 37]	-1
[7, 9, 12]	1
[27, 32, 34]	-1

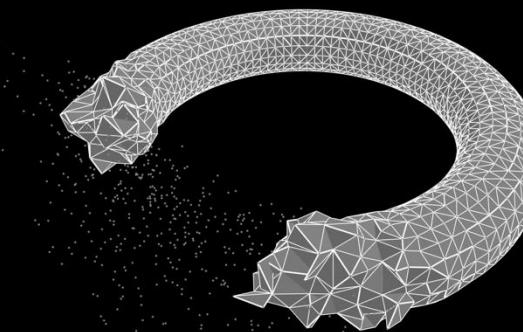
If you can write it on a page, you should be able to write it in code.

Open Applied Topology

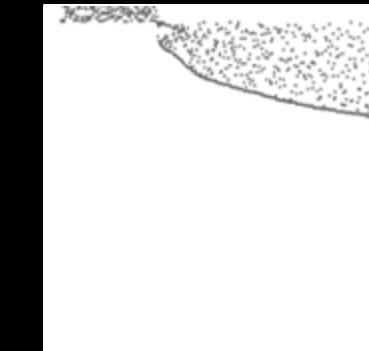
Data



Simplicial Complex



Matrix Algebra



User friendly software for big matrices

Matrix Oracles

The core of
Open Applied Topology



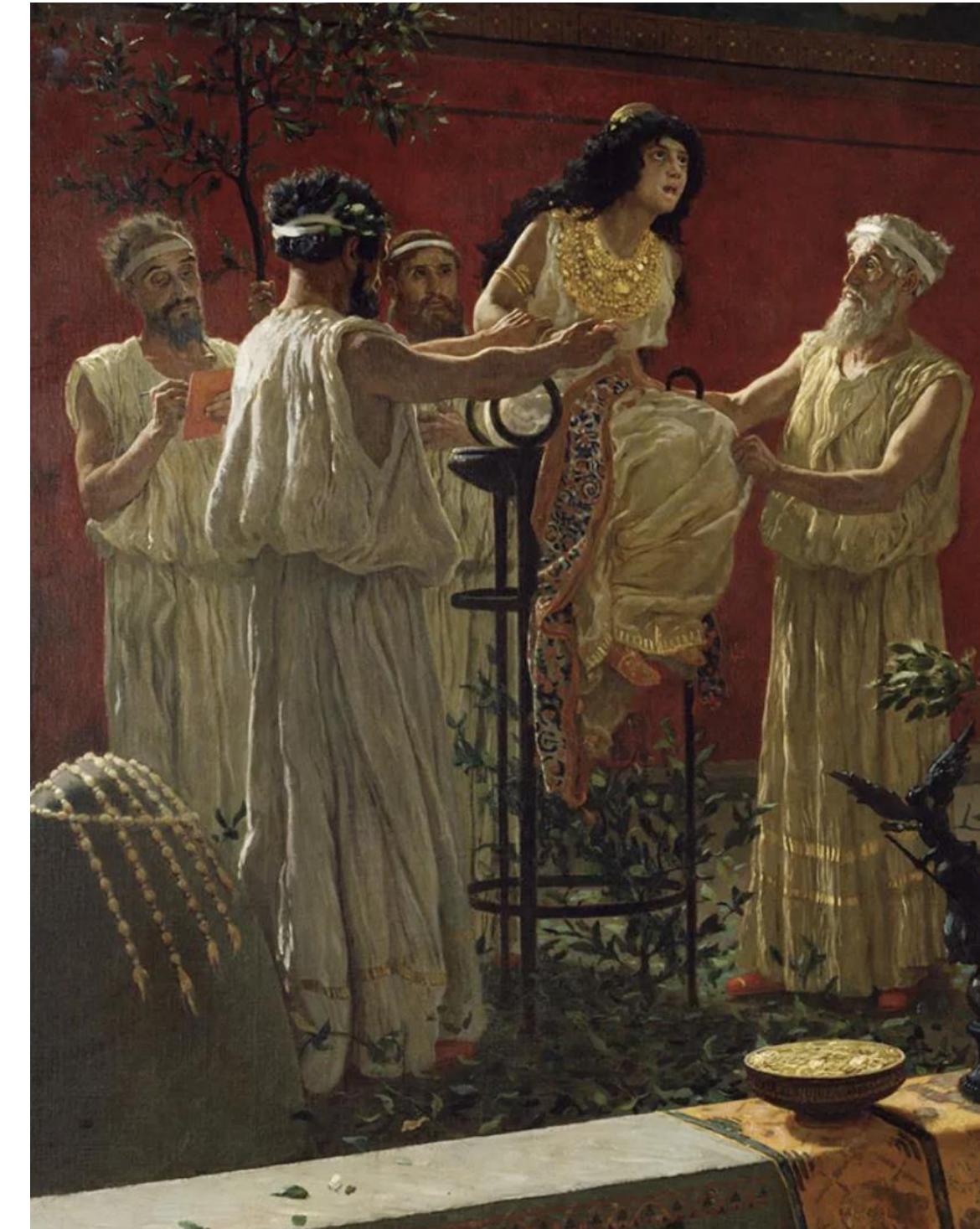
Matrix Oracles

Allow the user to look up entries in a matrix.



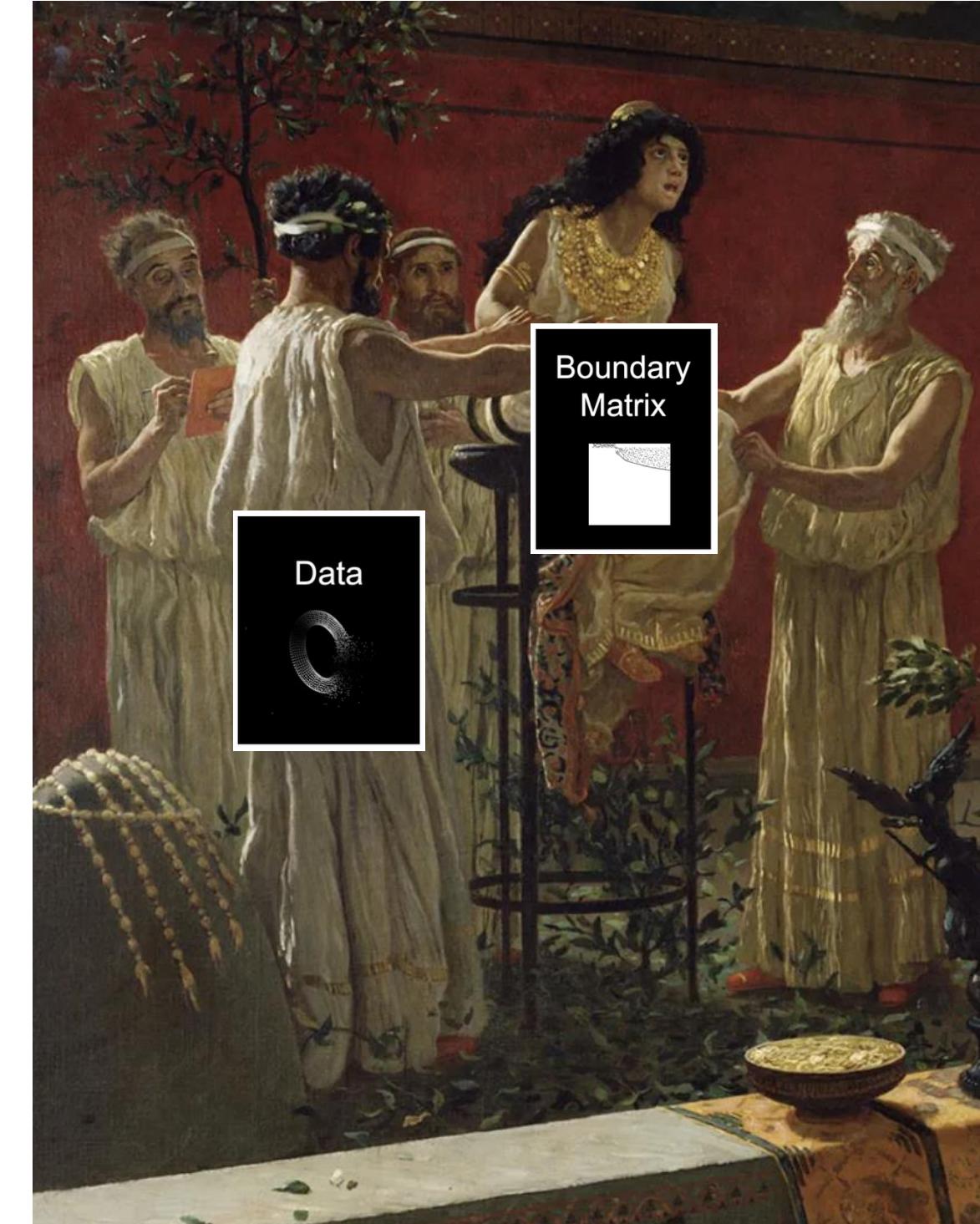
Matrix Oracles

Do not store entries in memory.



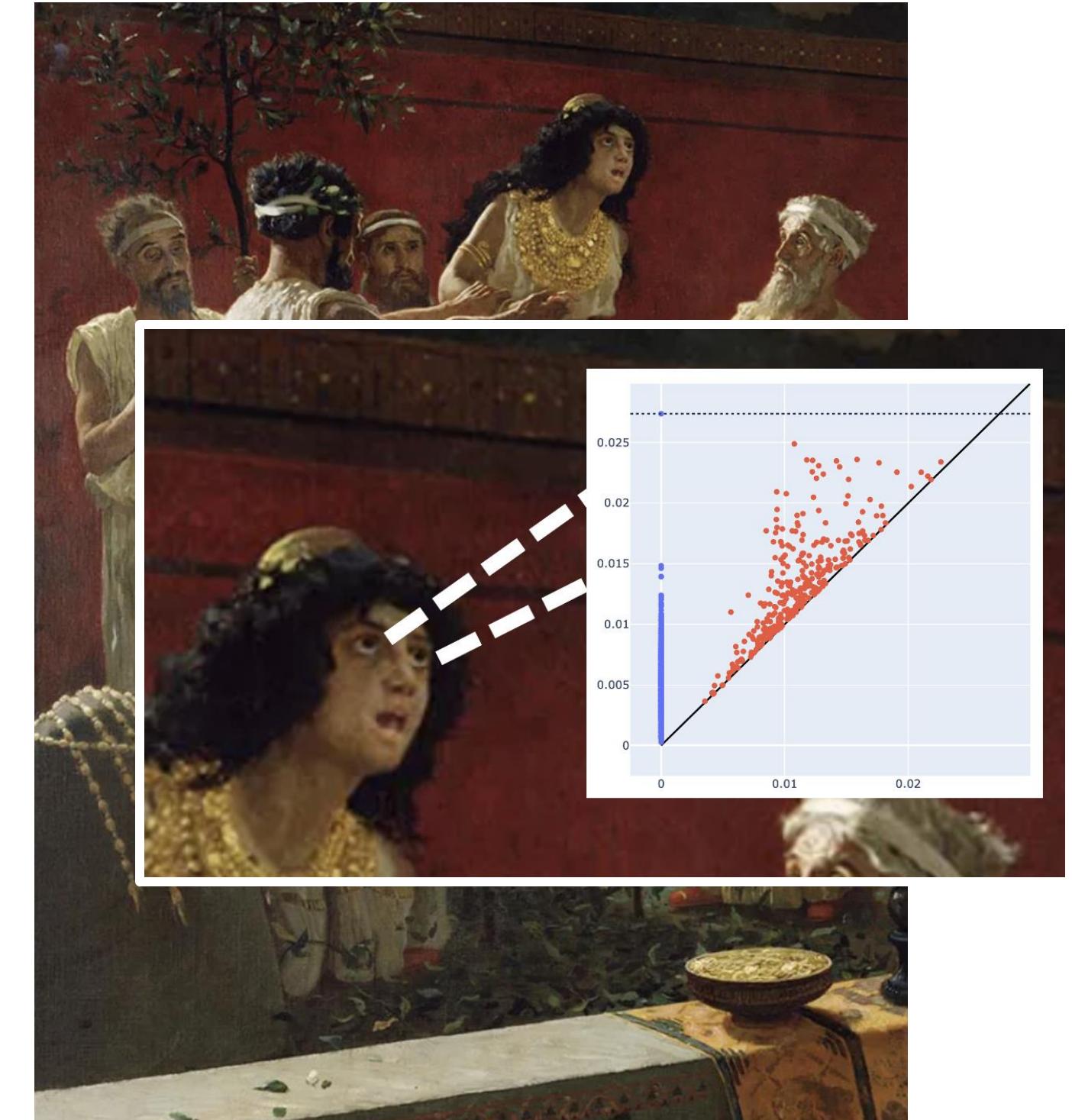
Matrix Oracles

Store only enough data to generate entries on demand.



Matrix Oracles

Revolutionized persistent
homology computation in
2017



Oracles in Open Applied Topology

Can be used interchangeably with ordinary matrices



$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Multiply



×



=



Multiply



$$\times \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}$$

Solve



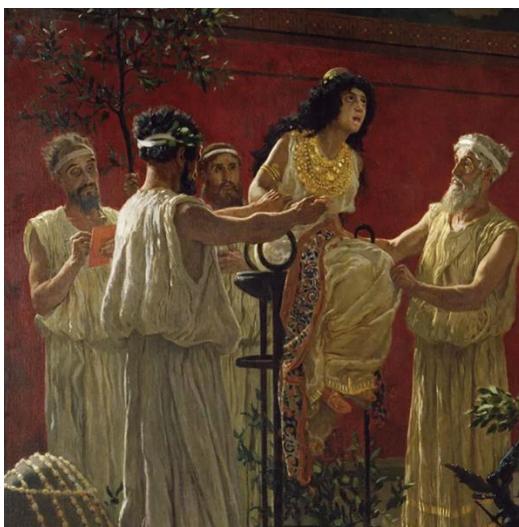
$$\times \begin{bmatrix} ? \\ \vdots \\ ? \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}$$

Invert

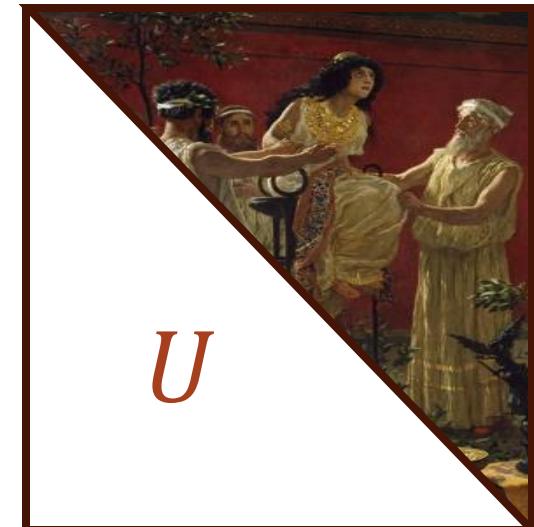
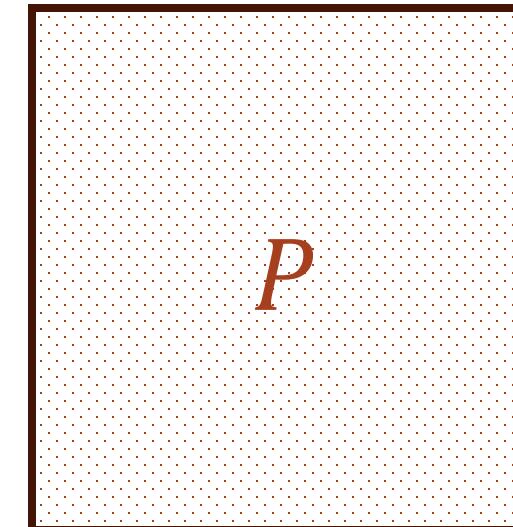


- 1

Factor



=



Optimize

minimize
subject to



$$c^T x \cdot x \leq b$$

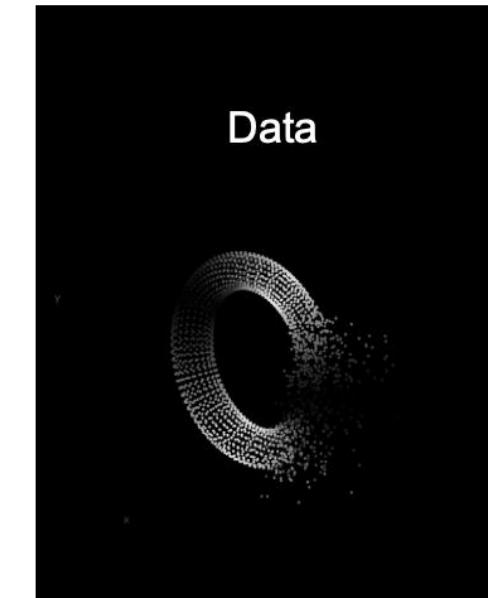
Index with simplices

```
simplex      = [ 0 , 1 , 2 , 3 ]  
column       = oracle.column( simplex )  
display( column )
```

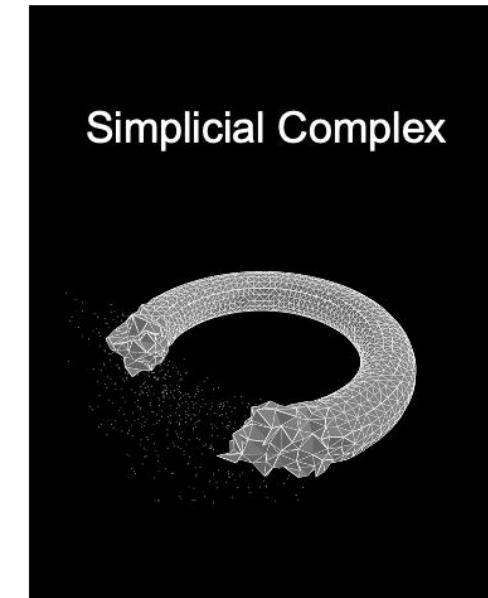
simplex	coefficient
[1,2,3]	1
[0,2,3]	-1
[0,1,3]	1
[0,1,2]	-1

Index with simplices

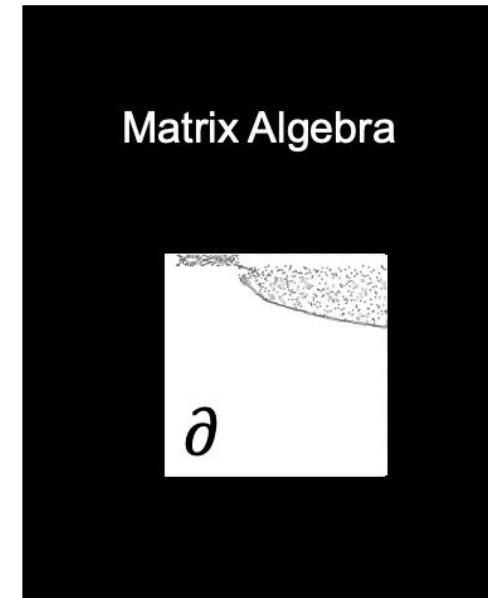
This indexing scheme avoids placing simplices in bijection with integers using a hash map.



$10k$
Points



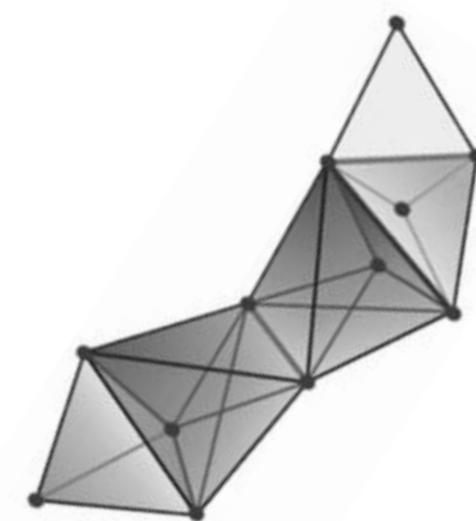
$167B$
Triangles



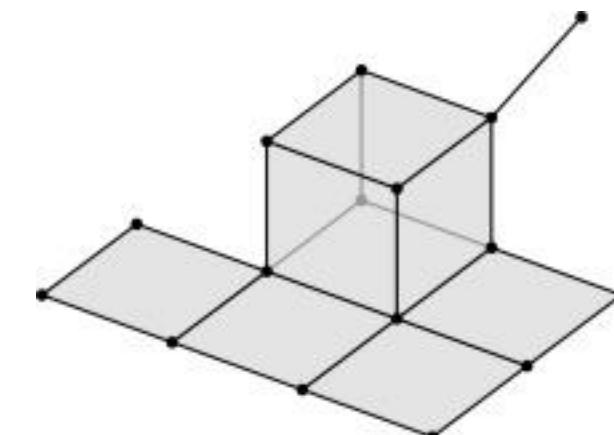
$167B$
Rows and Columns

5 terabytes
required to hash

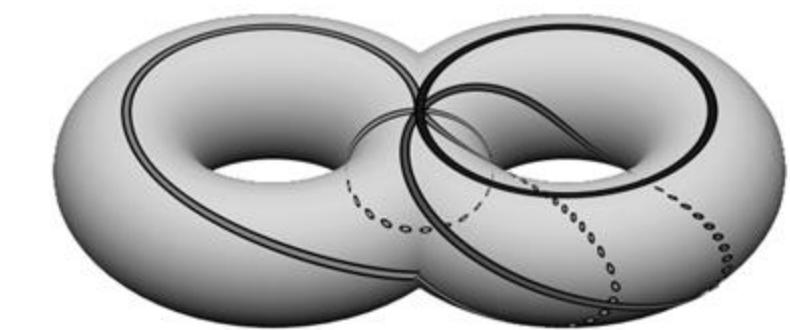
Index with anything



Simplices



Cubes



Cells

DIY coefficient ring

Users can construct customized coefficient rings.

The rings $\mathbb{Q}, \mathbb{R}, \mathbb{N}, \mathbb{Z}, \mathbb{Z}_p$ are provided.

DIY oracle

Users can construct new types of oracles.

```
class NewOracle:  
  
    def row(self, index): return [1,1,1]  
  
    def column(self, index): return [1,1,1]
```

*Pseudocode is for illustration only and does not compile. See OAT for details.

Matrix Oracles

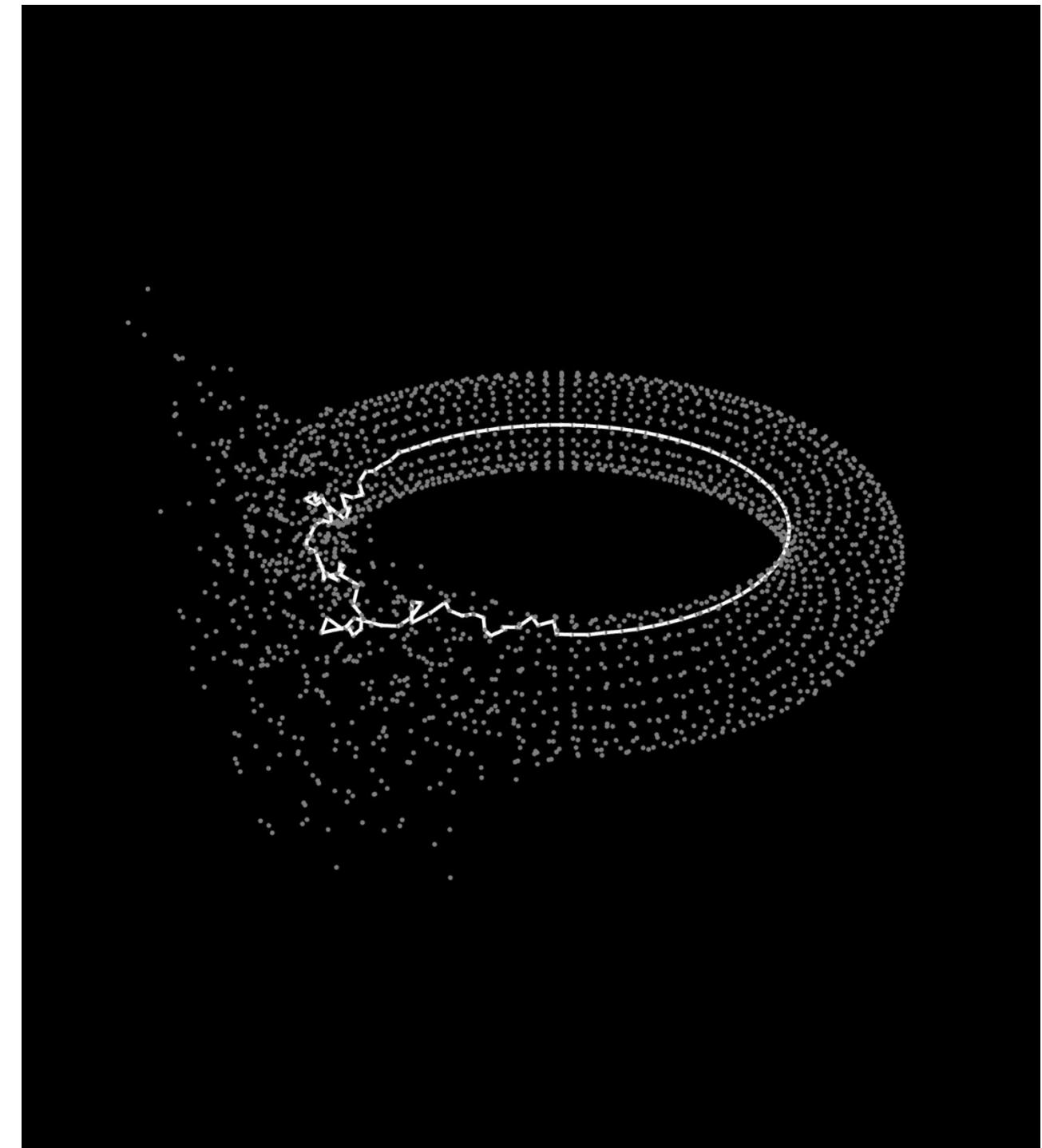
The key to big matrix algebra





Pacific
Northwest
NATIONAL LABORATORY

Applications



Fast (not fastest) computation of persistent homology barcodes

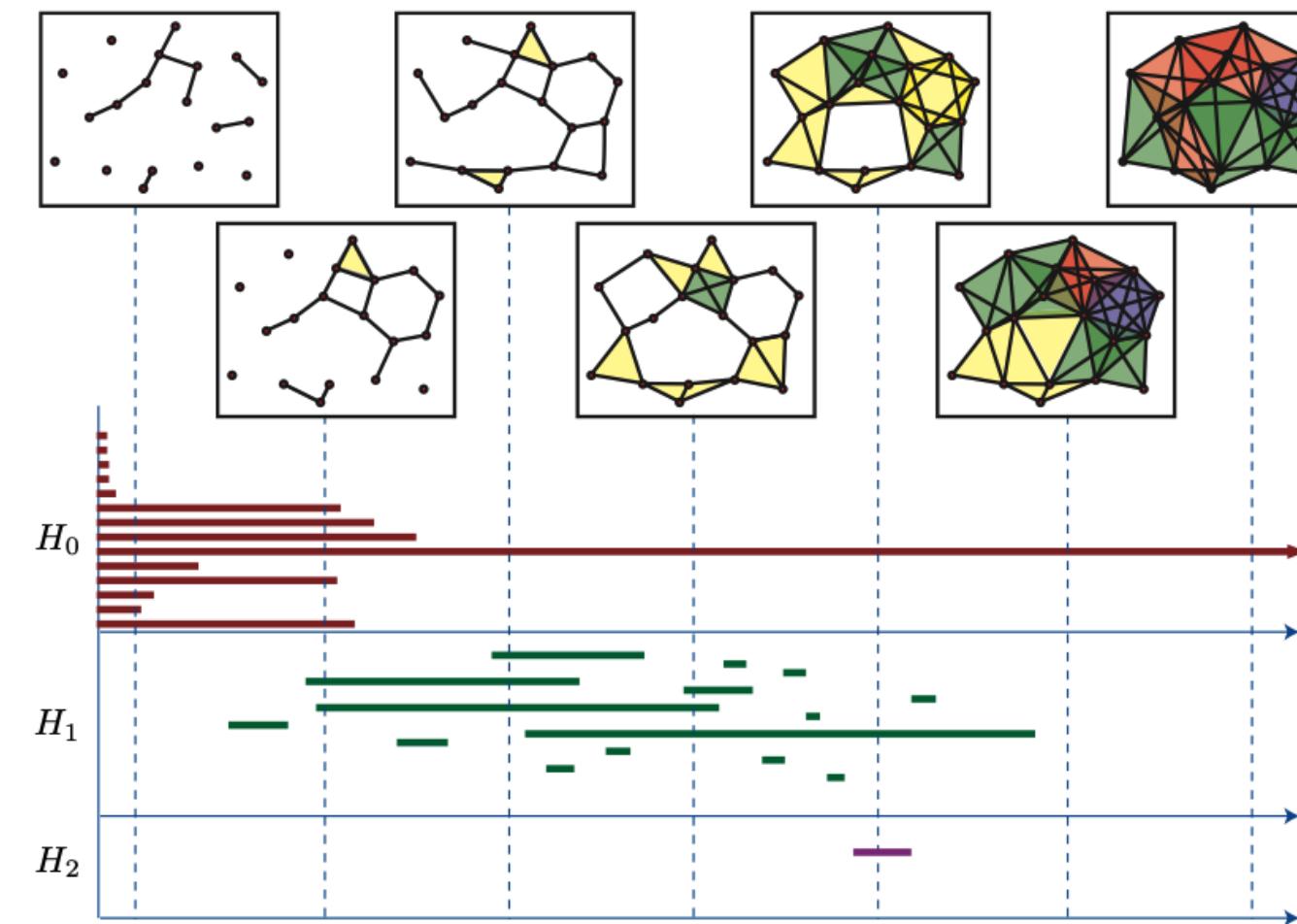
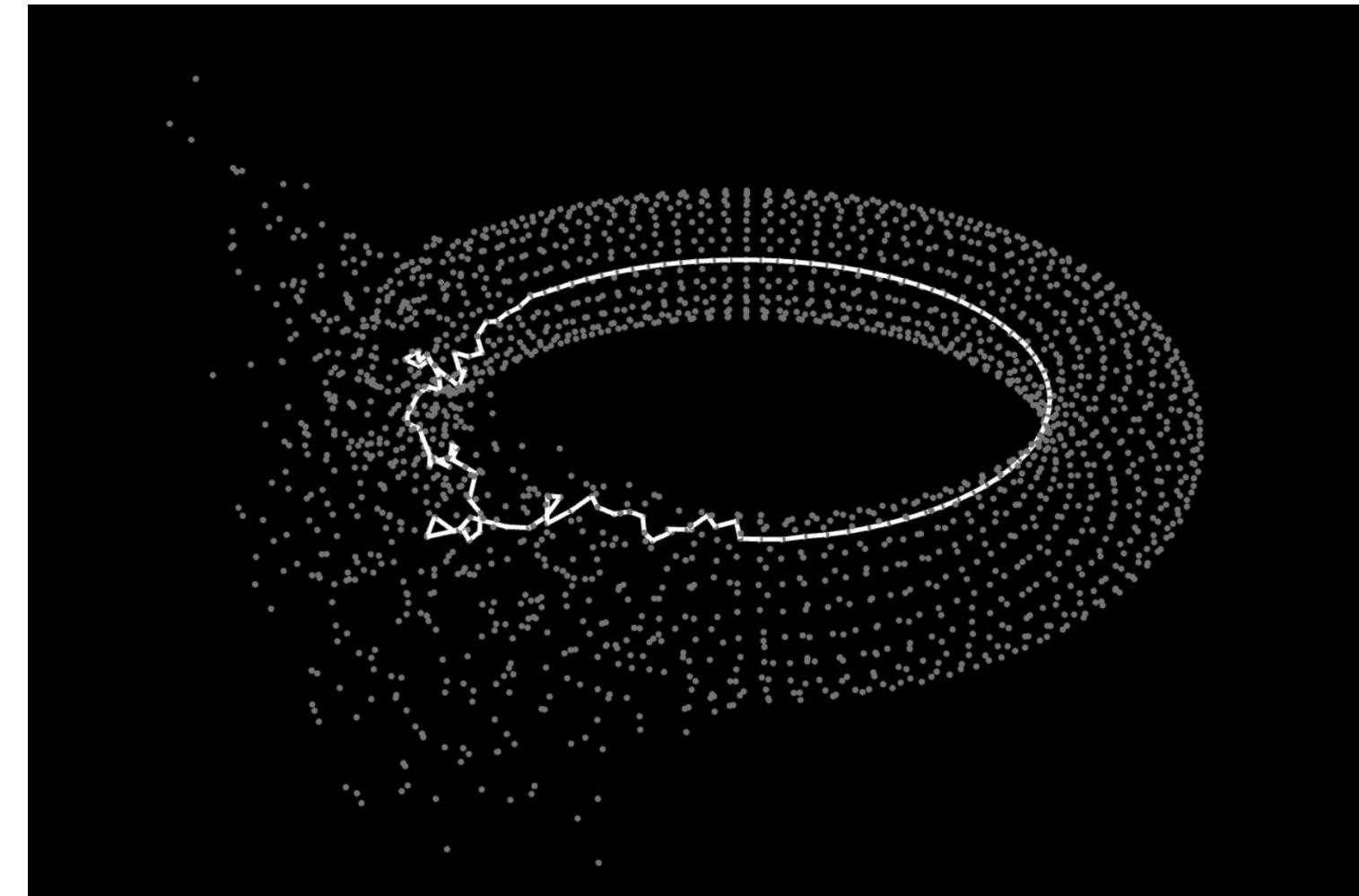


Figure credit: Ghrist, *The Persistent Homology of Data*

**Philosophy: Break scaling barriers with
new models, not fast models**

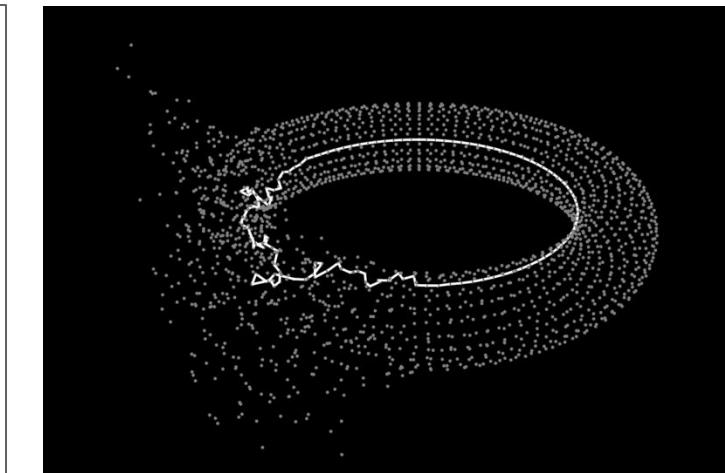
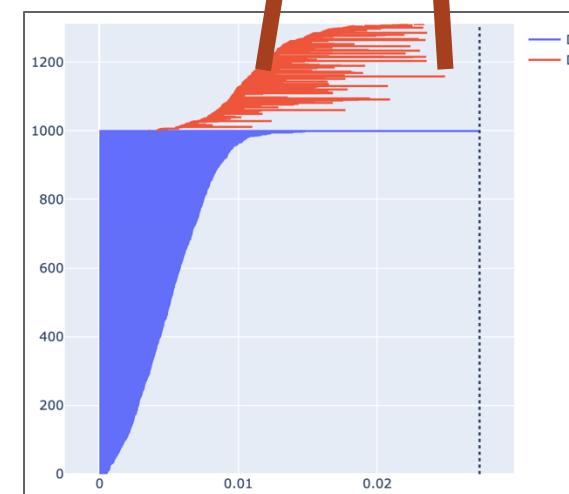
Cycle representatives in persistent homology



Cycle representatives in persistent homology

id	dimension	birth	death	birth simplex	death simplex	cycle representative	cycle nnz	bounding chain	bounding nnz
0	0	0.000000	0.006098	[999]	[476, 999]	simplex filtration coefficient 0 [999] ...	2	simplex filtration coefficient 0 [476,...	1.0
1	0	0.000000	0.005790	[998]	[53, 998]	simplex filtration coefficient 0 [998] ...	2	simplex filtration coefficient 0 [53, 9...	1.0
2	0	0.000000	0.005510	[997]	[539, 997]	simplex filtration coefficient 0 [997] ...	2	simplex filtration coefficient 0 [539,...	3.0
3	0	0.000000	0.001022	[996]	[889, 996]	simplex filtration coefficient 0 [996] ...	2	simplex filtration coefficient 0 [889,...	1.0
4	0	0.000000	0.008283	[995]	[936, 995]	simplex filtration coefficient 0 [995] ...	2	simplex filtration coefficient 0 [936,...	3.0
...
1306	1	0.004276	0.004310	[305, 404]	[305, 454, 581]	simplex filtration coefficient 0 [305,...	4	simplex filtration coefficient 0 ...	2.0
1307	1	0.004200	0.004236	[6, 384]	[6, 384, 650]	simplex filtration coefficient 0 [6,...	4	simplex filtration coefficient 0 [6...	2.0
1308	1	0.00413	0.004359	[418, 443]	[200, 387, 994]	simplex filtration coefficient 0 [200,...	6	simplex filtration coefficient 0 ...	4.0
1309	1	0.00414	0.00439	[623, 957]	[578, 623, 957]	simplex filtration coefficient 0 [623,...	4	simplex filtration coefficient 0 ...	2.0
1310	1	0.00450	0.00346	[76, 246]	[76, 246, 567]	simplex filtration coefficient 0 [76,...	4	simplex filtration coefficient 0 [...	2.0

1311 rows x 9 columns



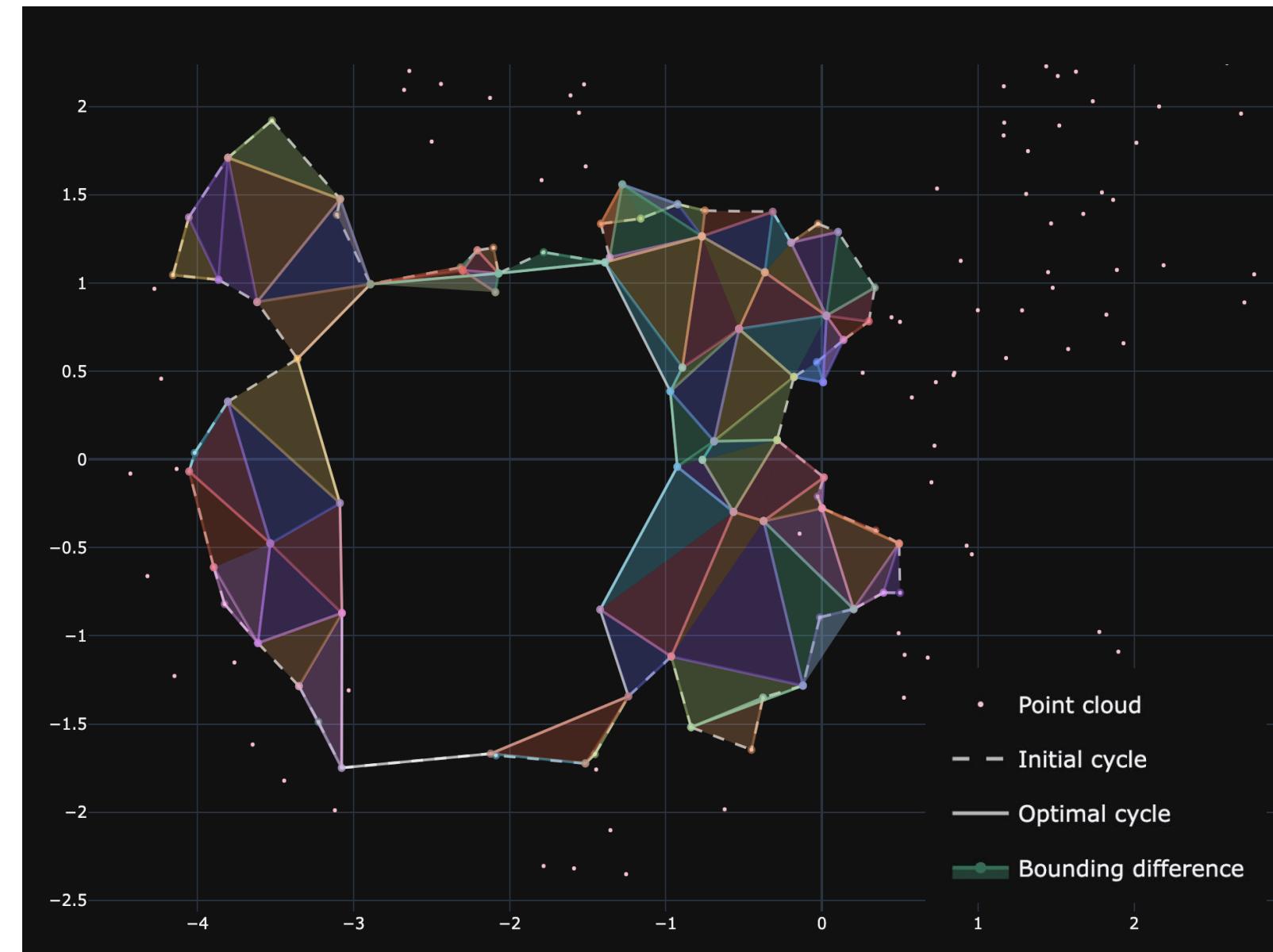
simplex	coefficient
[7, 17, 31, 34]	1
[4, 7, 17, 34]	-1
[3, 7, 17, 31]	1
[0, 4, 7, 17]	-1
[0, 3, 7, 17]	1
[7, 18, 31, 34]	-1
[4, 7, 18, 34]	1
[17, 24, 31, 34]	1
[3, 17, 24, 31]	1
[4, 9, 18, 34]	-1
[4, 8, 17, 34]	1
[4, 8, 9, 34]	-1

Bounding chains in persistent homology



Stanford Dragon
Persistent cycle representative (orange)
Bounding chain (white)

Feature localization by cycle optimization



Software Backend

Fast (speed of C)



Software Backend

Written in Rust

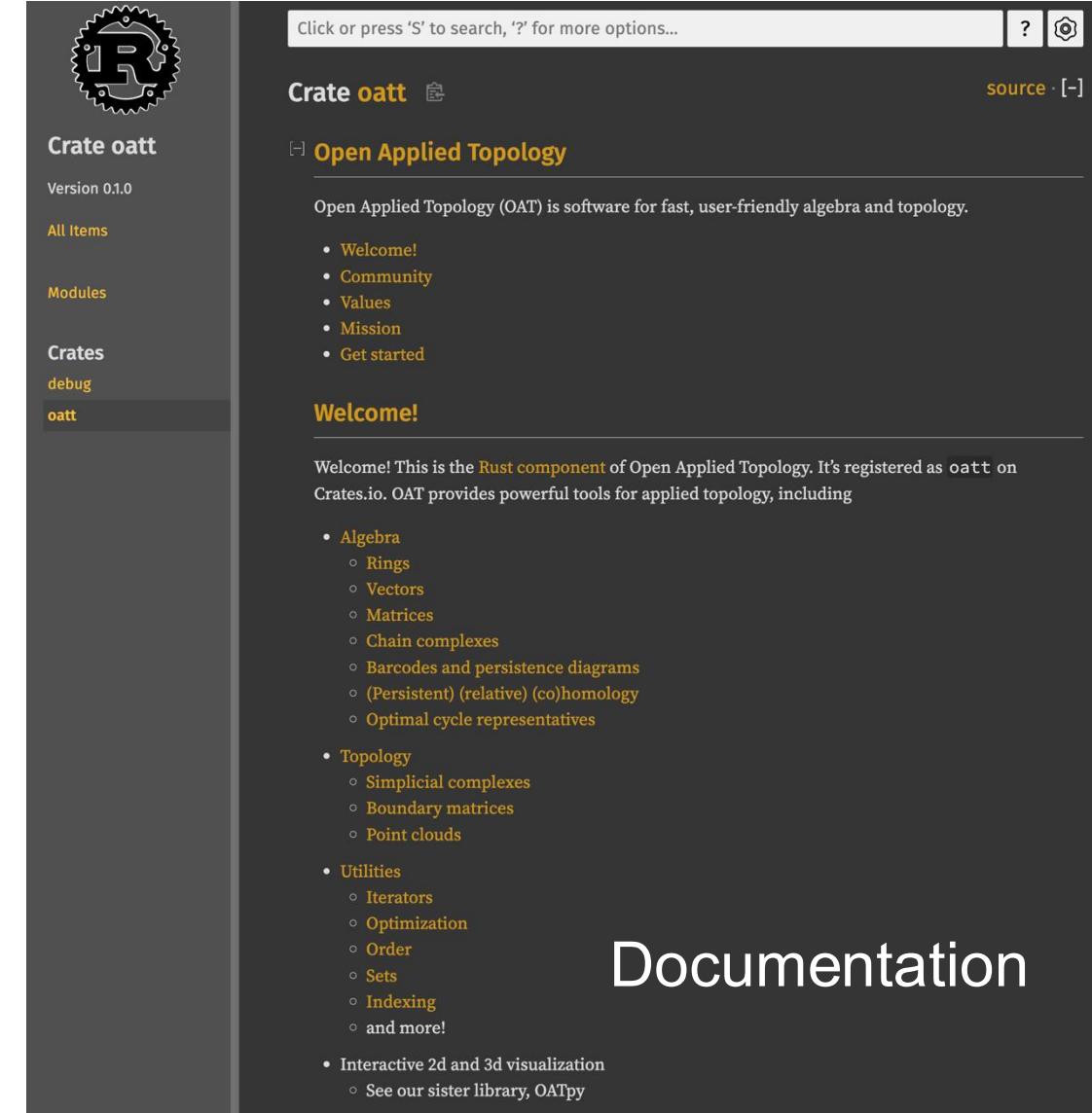
- Easy to install
- Easy dependency management
- Most loved language 9 years running*



*As measured by Stack Overflow surveys.

Software Backend

Written to be read



The screenshot shows the crates.io website interface. On the left, there's a sidebar with a Rust logo icon, navigation links for 'Crate oatt', 'Version 0.1.0', 'All Items', 'Modules', 'Crates' (with 'debug' and 'oatt' listed), and a search bar at the top right with placeholder text 'Click or press 'S' to search, '?' for more options...'. The main content area has a header 'Crate oatt' with a copy icon, a 'source [-]' link, and a 'source' link with a gear icon. Below the header, a section titled 'Open Applied Topology' is shown with a brief description: 'Open Applied Topology (OAT) is software for fast, user-friendly algebra and topology.' A 'Welcome!' section follows, stating: 'Welcome! This is the **Rust component** of Open Applied Topology. It's registered as `oatt` on Crates.io. OAT provides powerful tools for applied topology, including'. A large bulleted list details the features: Algebra (Rings, Vectors, Matrices, Chain complexes, Barcodes and persistence diagrams, (Persistent) (relative) (co)homology, Optimal cycle representatives); Topology (Simplicial complexes, Boundary matrices, Point clouds); Utilities (Iterators, Optimization, Order, Sets, Indexing, and more!); and Interactive 2d and 3d visualization (See our sister library, OATpy).

Documentation

Software Backend

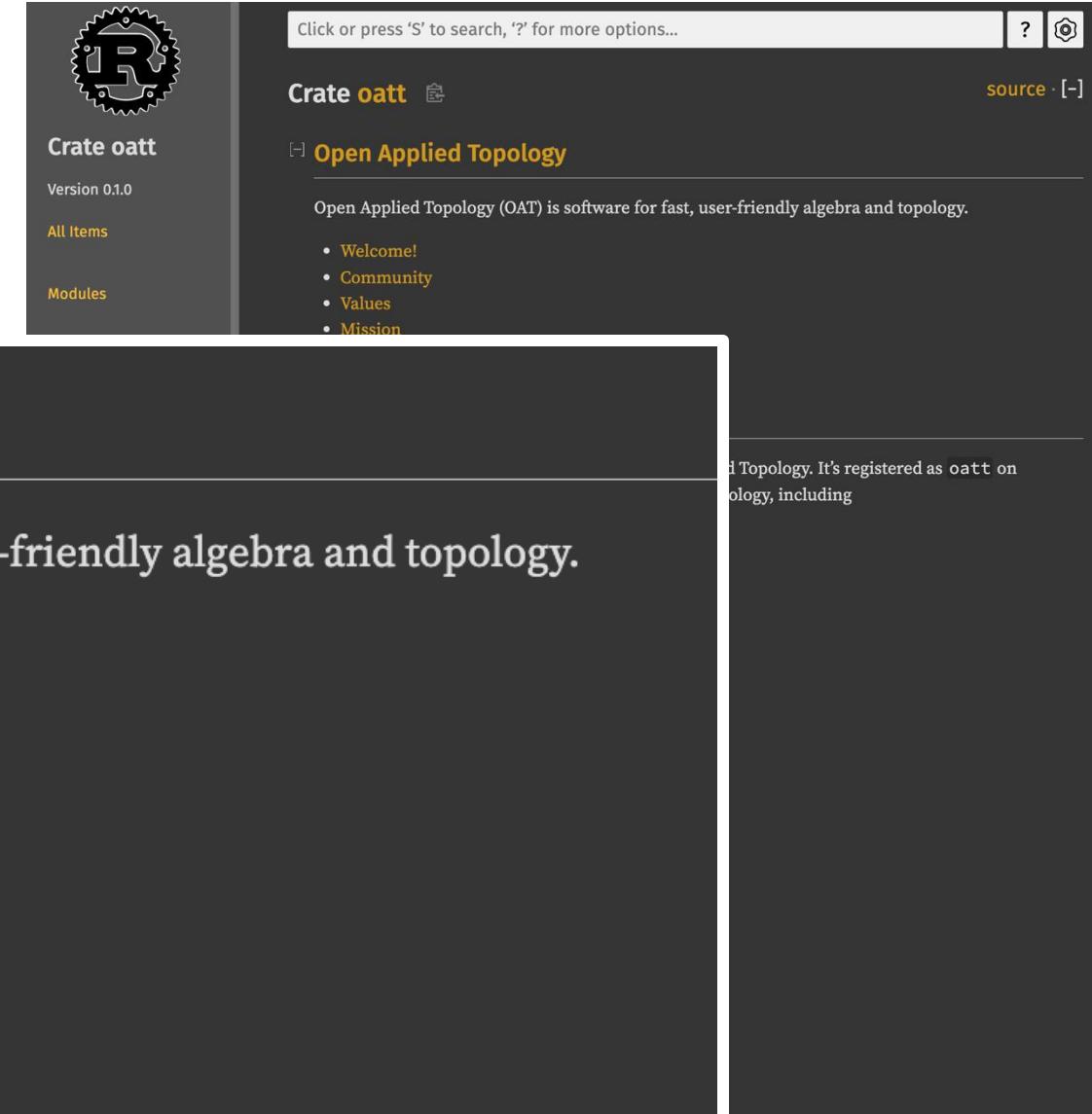
Written to be inclusive

[–] **Open Applied Topology**

Open Applied Topology (OAT) is software for fast, user-friendly algebra and topology.

- Welcome!
- Community
- Values
- Mission
- Get started

Welcome!



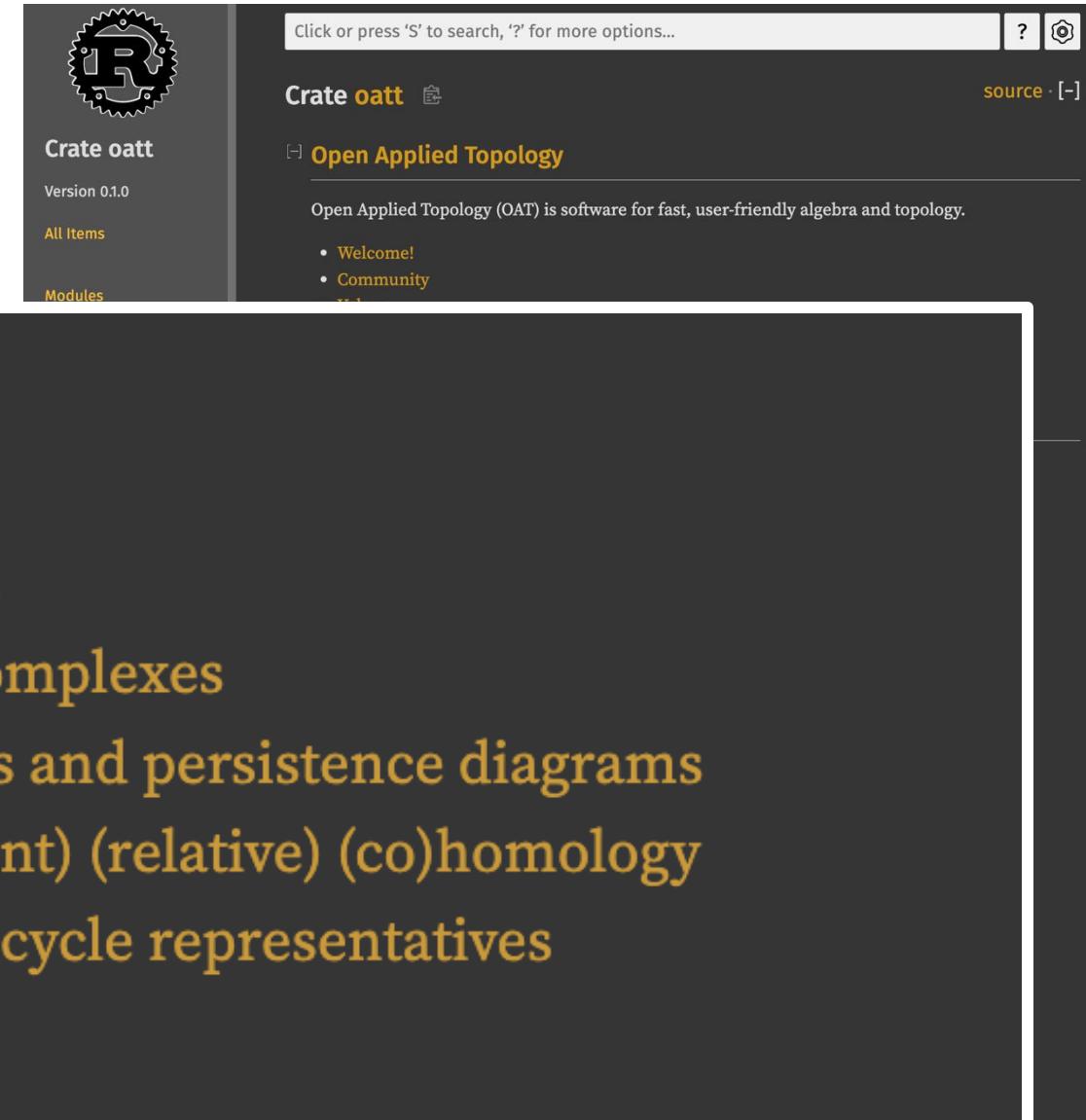
The screenshot shows a software application window titled "Crate oatt". At the top right are buttons for help (?), settings (gear), and source (-). A search bar says "Click or press 'S' to search, '?' for more options...". Below the title, there's a large button labeled "Open Applied Topology". The main content area contains a brief description of OAT as "software for fast, user-friendly algebra and topology", followed by a bulleted list of links: "Welcome!", "Community", "Values", "Mission", and "Get started". At the bottom of the content area, it says "Topology. It's registered as oatt on topology, including".

47

Software Backend

Written to be modular

- Algebra
 - Rings
 - Vectors
 - Matrices
 - Chain complexes
 - Barcodes and persistence diagrams
 - (Persistent) (relative) (co)homology
 - Optimal cycle representatives
- Topology
 - Simplicial complexes
 - Boundary matrices



Software Backend

Written to be modified

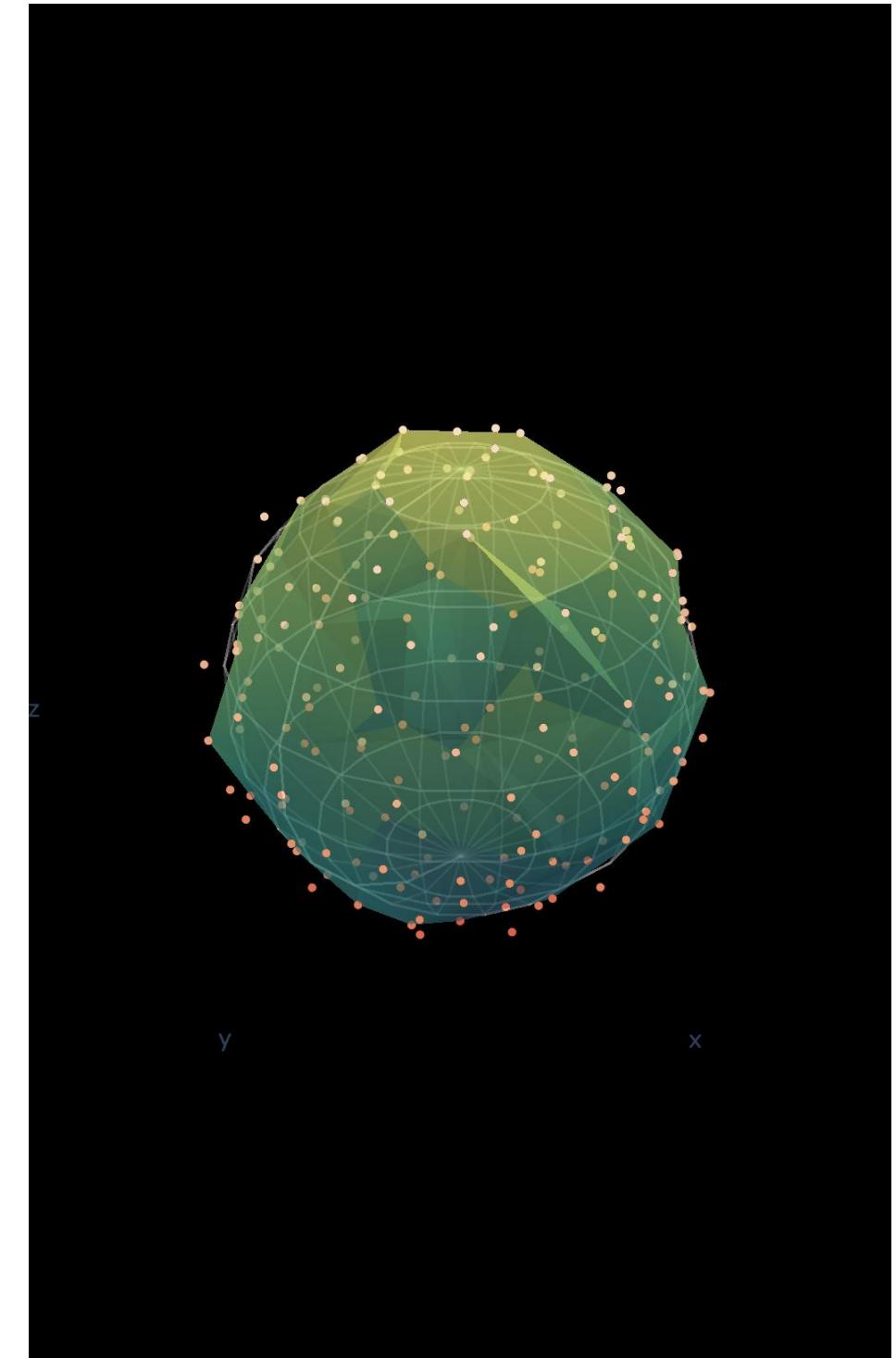
Modify and create your own components

- Algebra
 - Rings
 - Vectors
 - Matrices
 - Chain complexes
 - Barcodes and persistence diagrams
 - (Persistent) (relative) (co)homology
 - Optimal cycle representatives
- Topology
 - Simplicial complexes
 - Boundary matrices
 - Point clouds
- Utilities
 - Iterators
 - Optimization
 - Order
 - Sets
 - Indexing

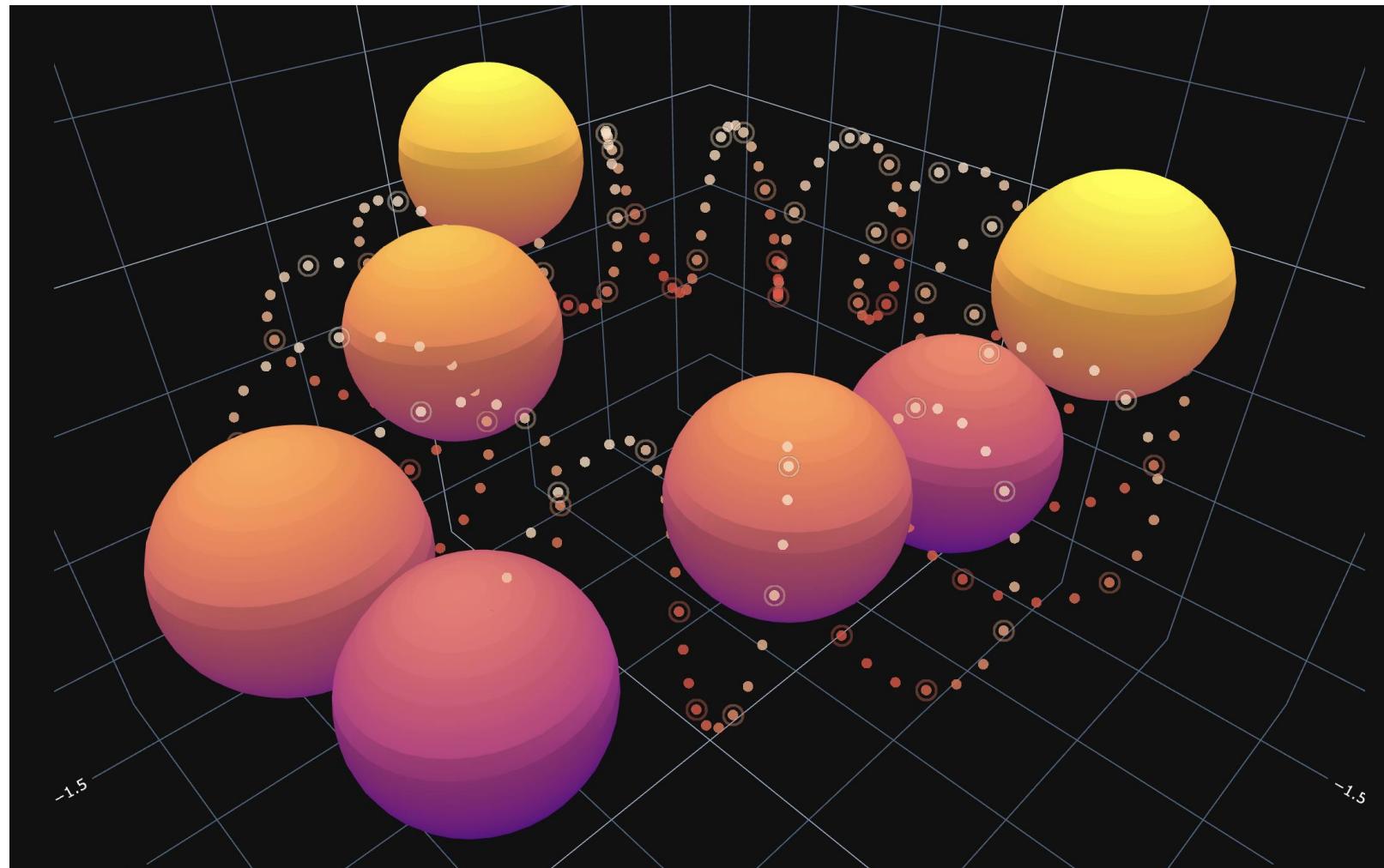
*As measured by Stack Overflow surveys.

Software Frontend

Written in Python
Pip-installable
Does not require Rust



Software Frontend



Topological Spaces

Simplicial complexes

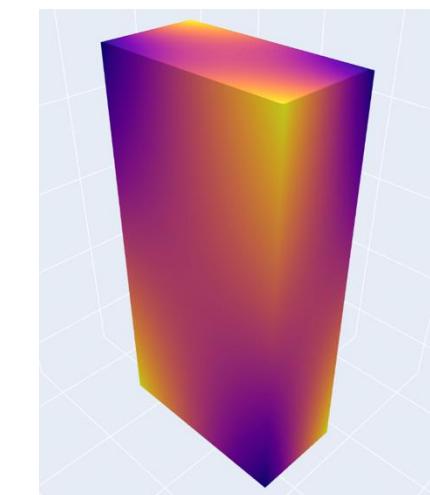
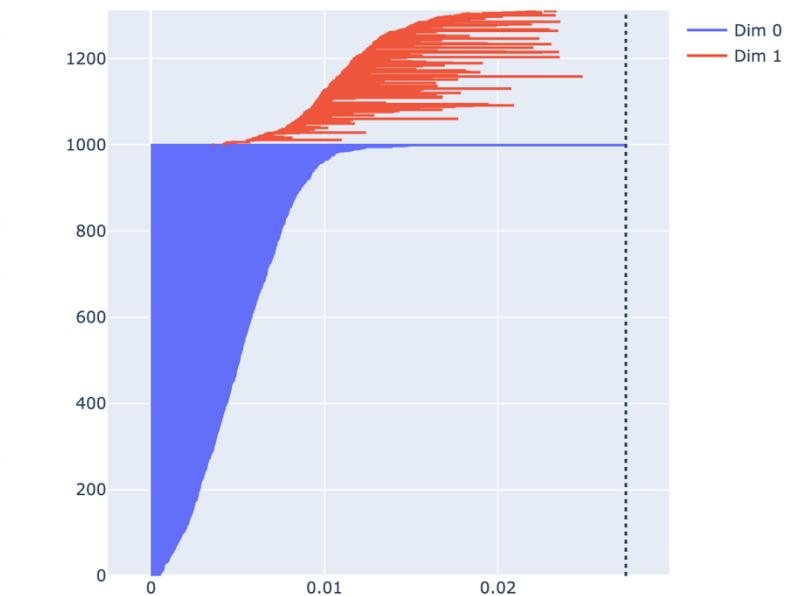
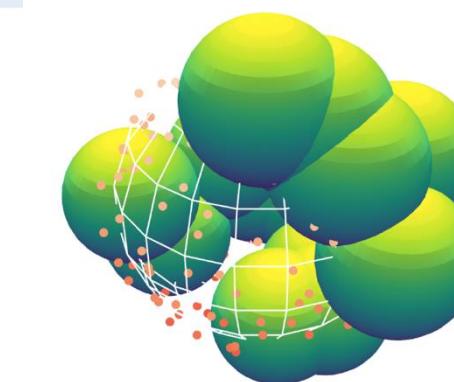
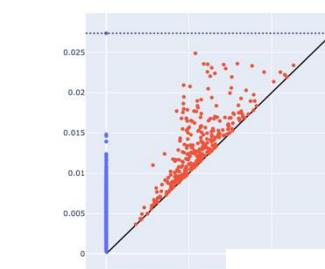
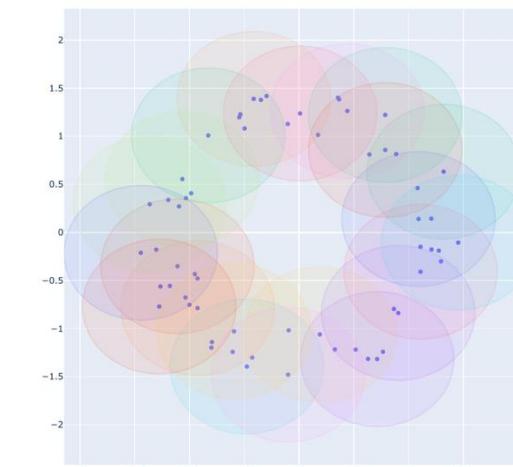
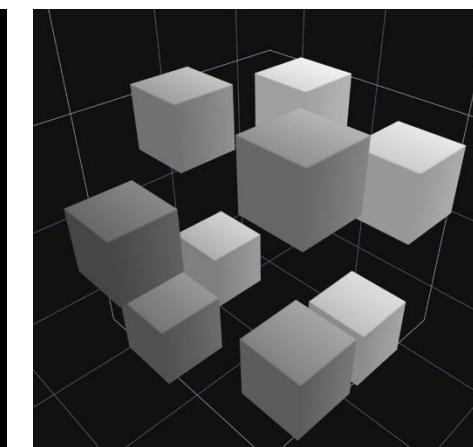
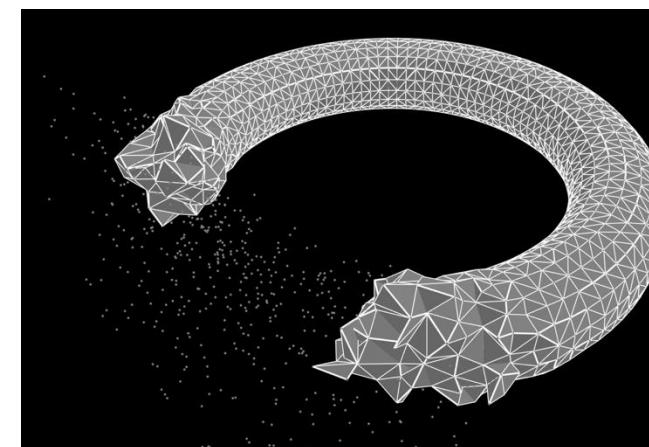
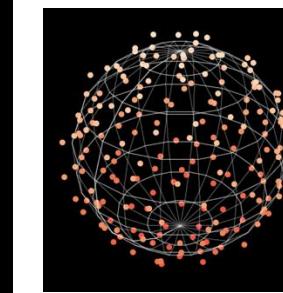
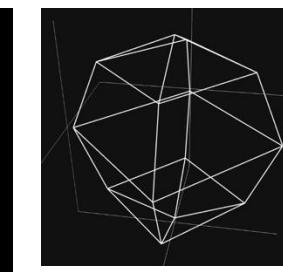
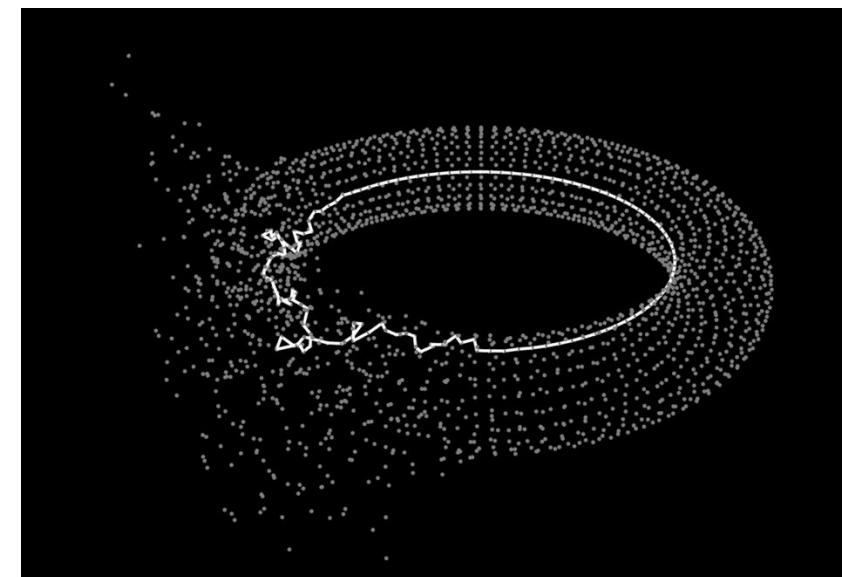
- Vietoris-Rips (point cloud)
- Nerve (poset)
- Dowker (toplex)

Coming soon

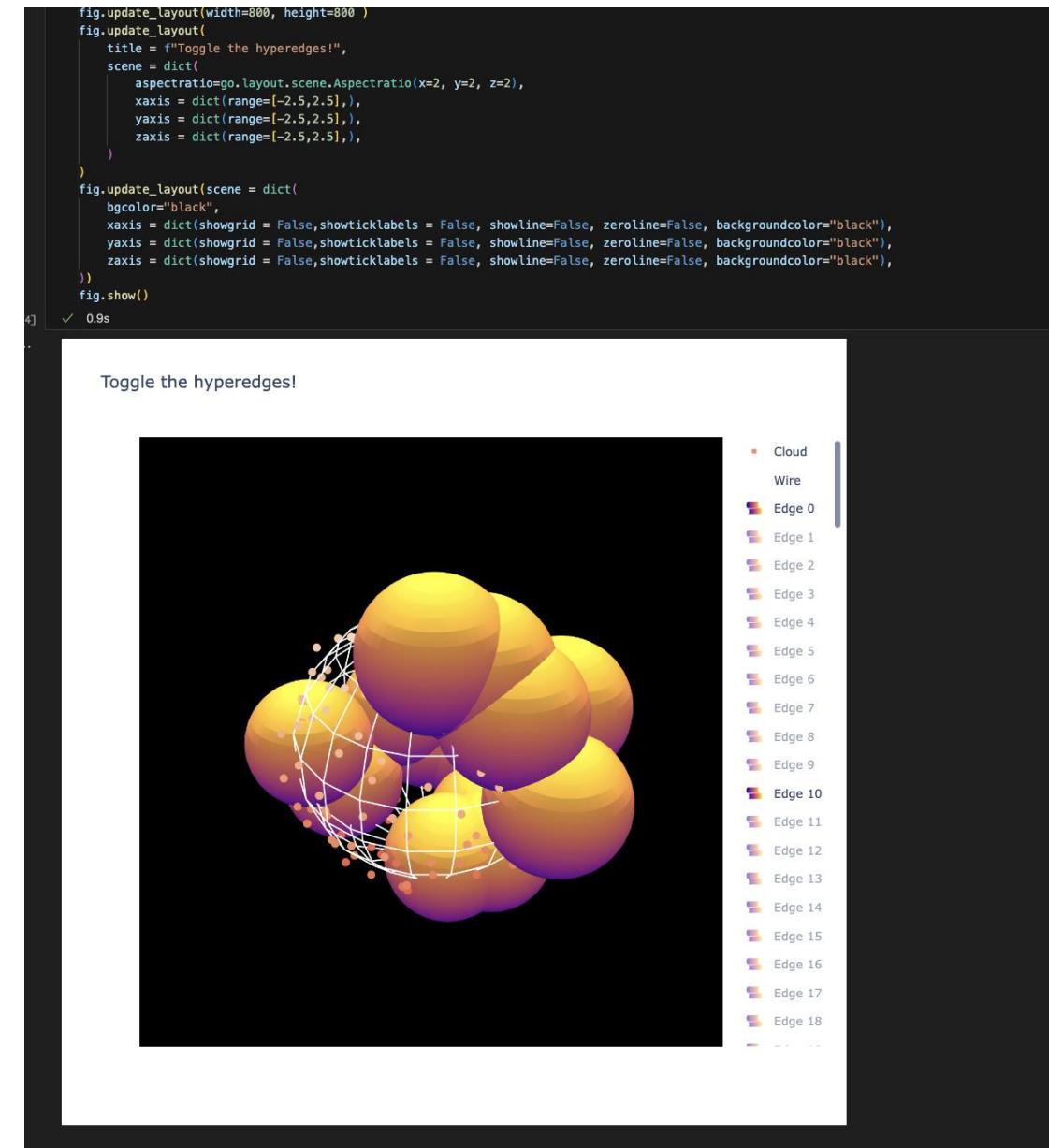
- Cubical (digital images)
- Delaunay
- Čech
- Etc.

Software Frontend

Visualization



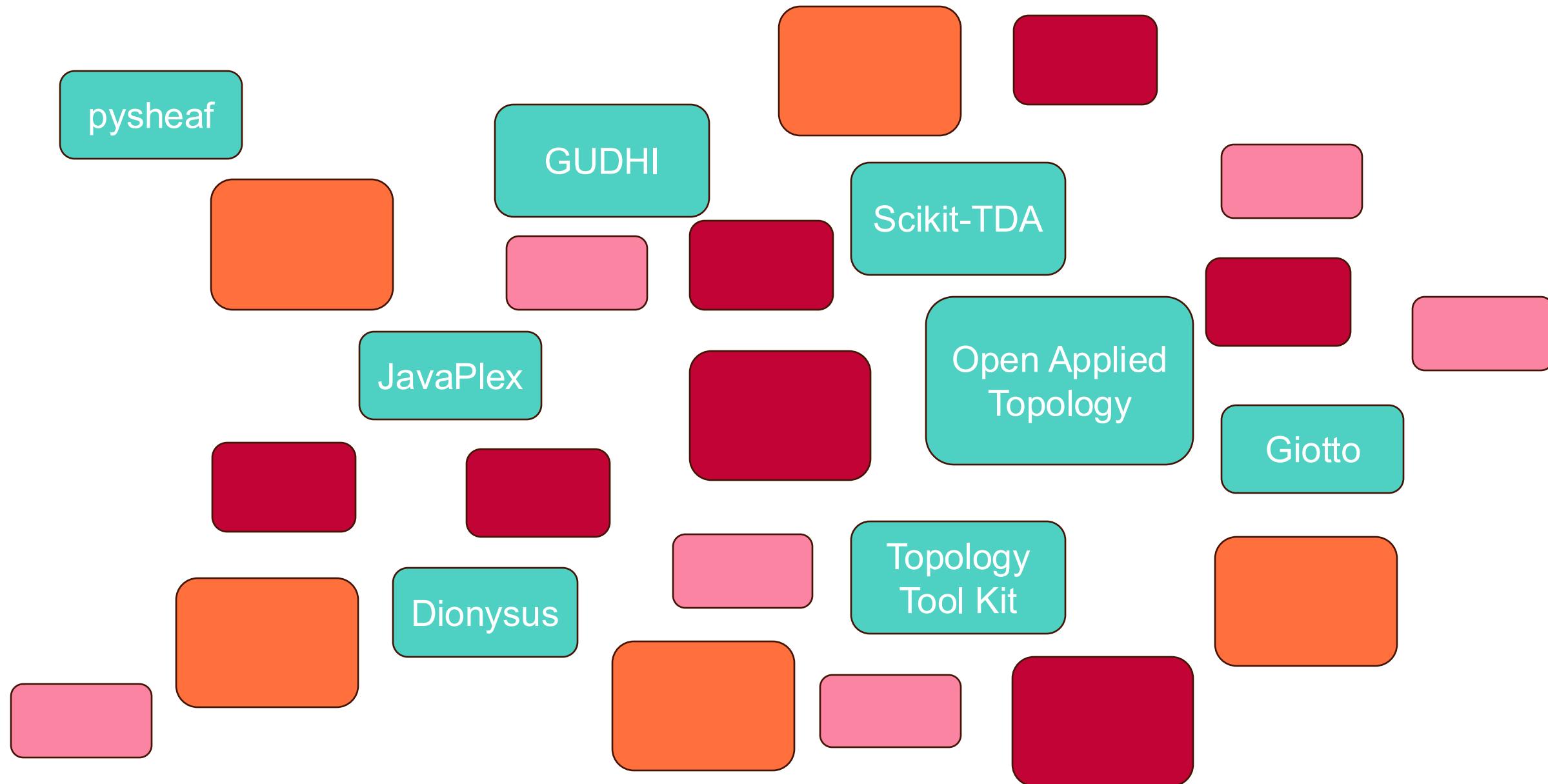
Jupyter Tutorials



Program Executables

```
    \ executable_files
        ⌄ compute_homology.exe
        ⌄ construct_boundary_matrix.exe
        ⌄ optimize_cycle.exe
```

Ecosystem development

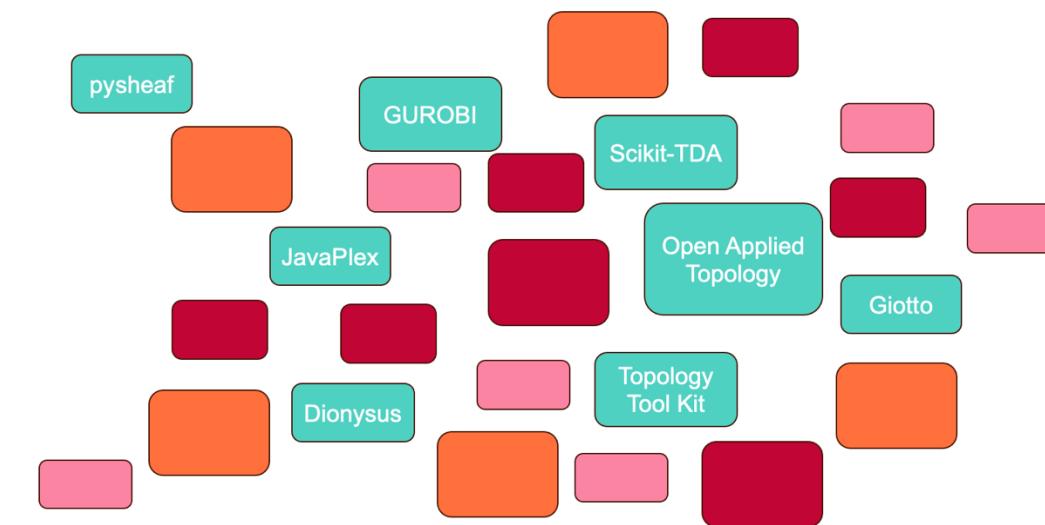


Ecosystem development

Community outreach (Banff 2025)

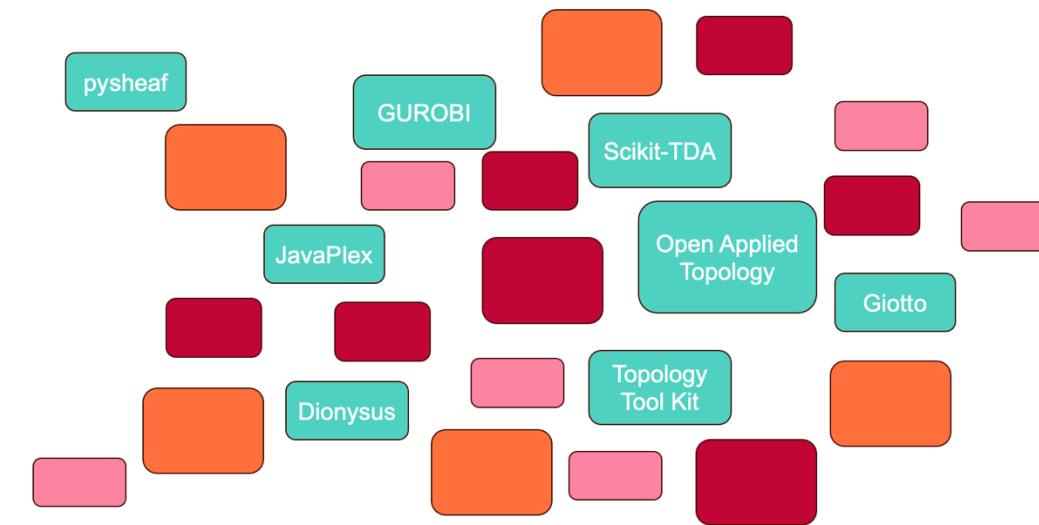
Student research

Support for developers (consultation, training)



Ecosystem development

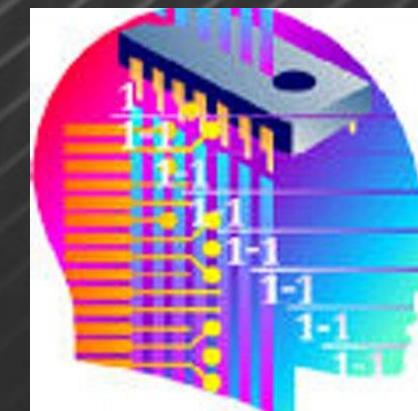
Help us improve!
(We'll help you get started.)





Support

- ONR N00014-16-1-2010
- NSF-1934960
- NSF DMS-1854748
- EP/R018472/1
- Swartz Center for Theoretical Neuroscience, Princeton University



EPSRC

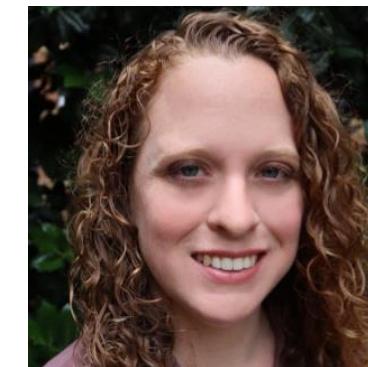


PNNL is operated by Battelle for the U.S. Department of Energy

Contributors



Emilie Purvine



Alyson Bittner



Helen Jenne



Audun Myers



Brian Kritzstein



Cliff Joslyn



William Kay



Garret Seppala



Stephen J. Young



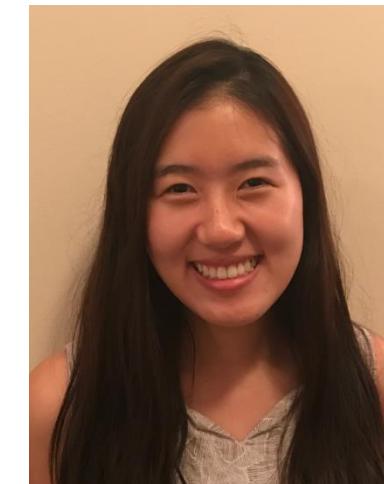
Sinan Aksoy



Contributors



Lori Ziegelmeier



Iris Yoon



Heather Harrington



Thomas Chaplin



Ethan Schleek



Lu Li



Haibin Hang



Chad Giusti



Nikolas Shonsheck



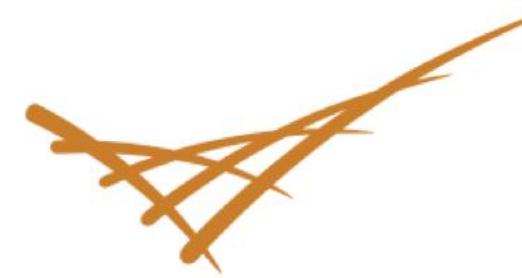
Jacob Leygonie



Bryn Keller



Sophia Wiedmann



Pacific
Northwest
NATIONAL LABORATORY



BROWN



MACALESTER



MICHIGAN STATE
UNIVERSITY



UNIVERSITY OF
OXFORD

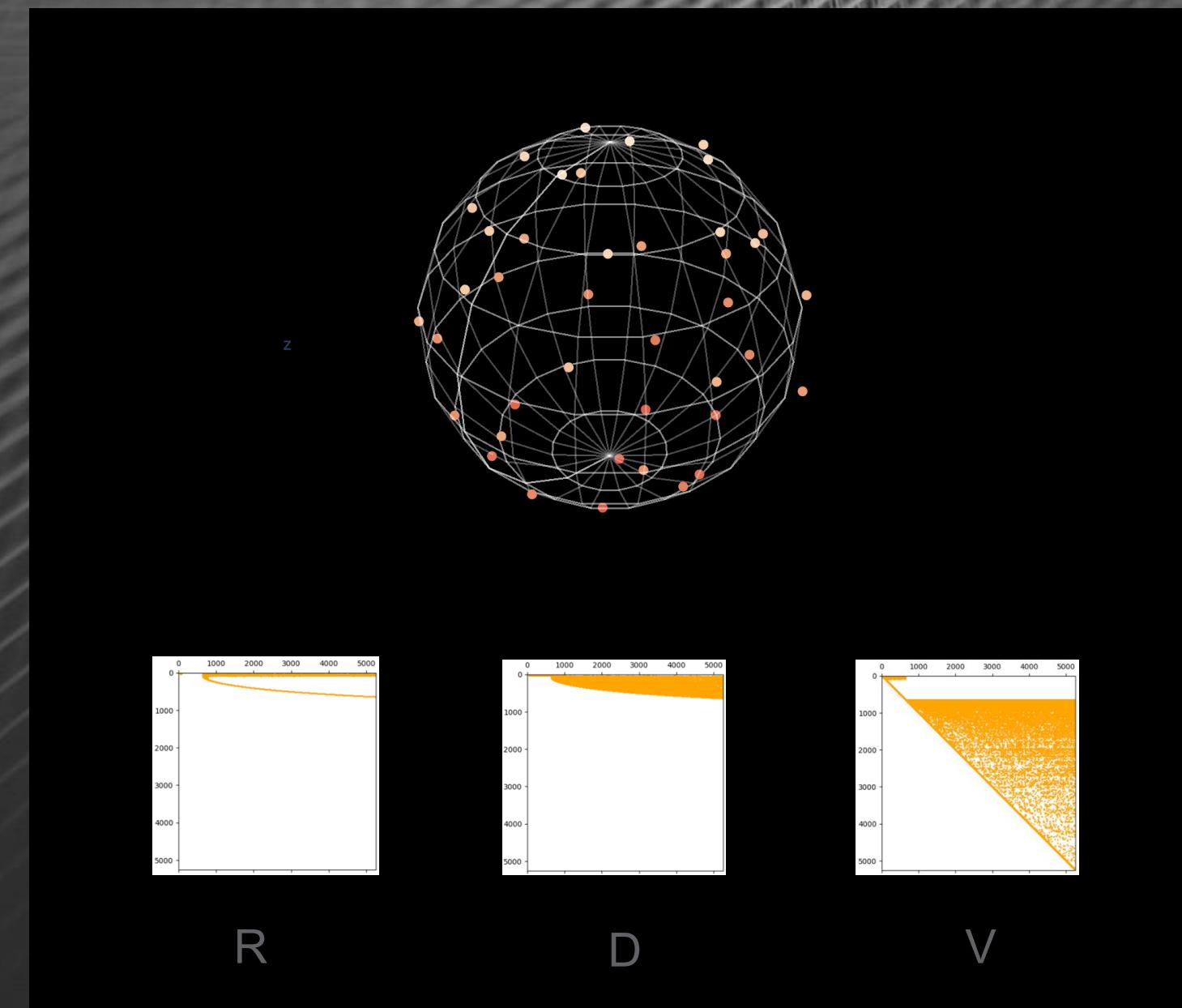


Pacific
Northwest
NATIONAL LABORATORY

Open Applied Topology

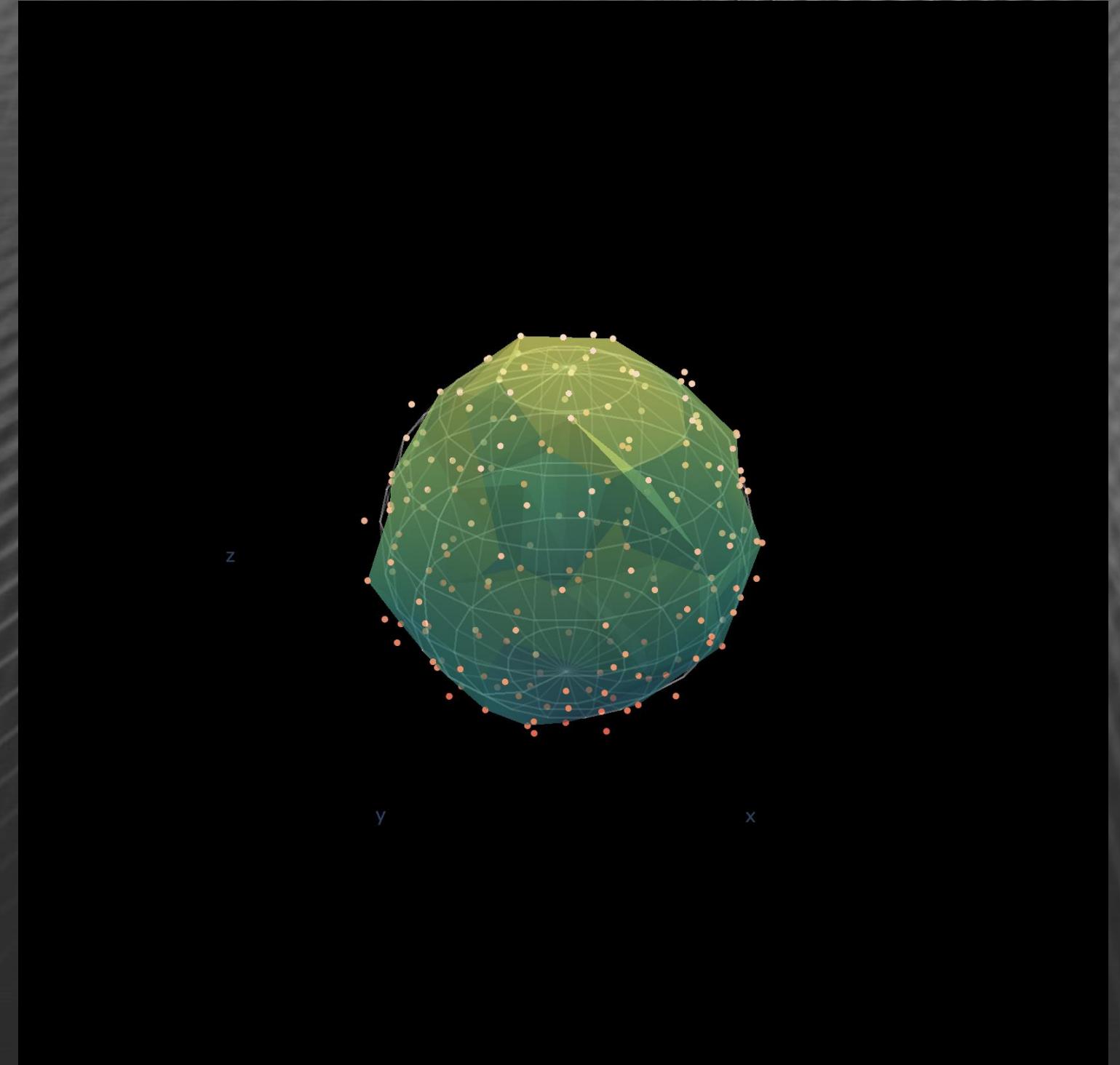
Available at
<https://openappliedtopology.github.io>

(Major updates coming soon)





Pacific
Northwest
NATIONAL LABORATORY



U.S. DEPARTMENT OF
ENERGY **BATTELLE**

PNNL is operated by Battelle for the U.S. Department of Energy