Asymptote: Past, Present, and Future

John Bowman

Department of Mathematical and Statistical Sciences,
University of Alberta

Collaborators:
Andy Hammerlindl, Charles Staats, Ben Bingham

July 19, 2026

www.math.ualberta.ca/∼bowman/talks

Past

- 1979: TEX and METAFONT [Knuth 1986a], [Knuth 1986b]
- 1986: 2D Bézier control point selection [Hobby 1986]
- 1989: MetaPost (Hobby)
- 2004: Asymptote (Hammerlindl, Bowman, Prince)
- 2005: 3D Bézier control point selection [Bowman 2007]
- 2008: 3D interactive TEX in OpenGL and PRC+PDF [Bowman & Shardt 2009]
- 2009: 3D billboard labels — always face camera
- 2010: 3D PDF enhancements [Bowman 2010]

- 2016: Bézier triangle patches (Frohlich)
- 2019: (Jamie Rassameemasmuang)
  - Physically based rendering
  - AsyGL library: interactive 3D in HTML via WebGL
  - Image-based lighting
- 2021:
  - Order-independent transparency
  - Portable compressed V3D file format
- 2024: Templated imports, `using` and `autounravel` keywords

Present (2026)

- Qt6 upgrade of the `Xasy` graphical front end
- Collections Library (Staats)
  - User-defined struct iteration via `operator iter`
  - Bracket indexing: `operator[]` and `operator[=]`
  - Native hashing of ints, strings, reals, int arrays
- Vulkan renderer
- V3D-aware PDF plugin for Okular

Templated Imports

- Types specified at import time: in the module file, declare

```
typedef import(T, S, Number);
```

- Import with concrete types:

```
access myModule(T=string, S=int[], Number=real)
       as myModule_string_int_real;
```

- Enables generic, type-safe library code written entirely in Asymptote.

New keywords `using` and `autounravel`

- `using` provides a modern type alias (like C++):

```
using multipath = path[];
```

- `autounravel` automatically makes struct members available in the surrounding scope:

```
struct rational {
  int p=0, q=1;
  autounravel rational operator +(rational a, rational b) {
    return rational(a.p*b.q+b.p*a.q, a.q*b.q);
  }
}
```

- The + operator is now accessible wherever `rational` is used.

Collections Library

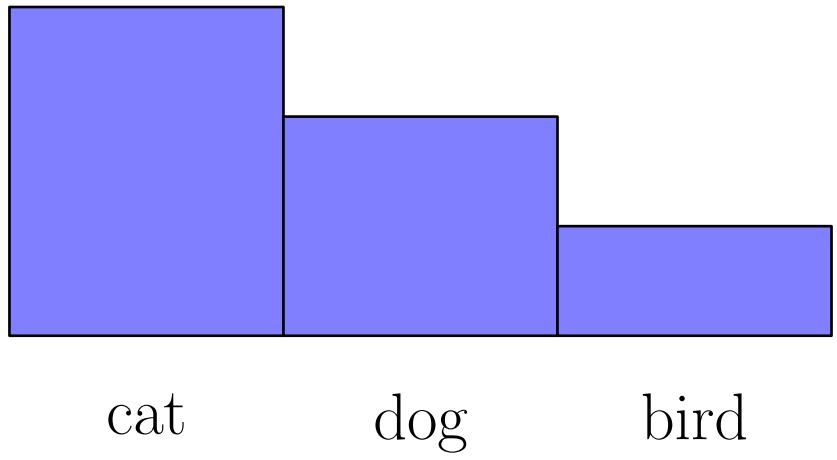
- Hash maps, B-tree maps (maintain sorted order), hash sets, sorted sets, queues.
- Native hashing of ints, strings, reals, and int arrays enables these types as map keys.
- Hash maps (dictionaries) via templated modules:

```
from collections.hashmap(K=string, V=int) access
     HashMap_K_V as HashMap_string_int;

HashMap_string_int h = HashMap_string_int();
h['A'] = 4;
h['B'] = 2;

for (string key : h)
  write(h[key],none);
```

42

Hash Map Frequency Analysis

```
import graph;
string[] pets = {'cat', 'dog', 'cat', 'bird', 'dog', 'cat'};
from collections.hashmap(K=string, V=int) access HashMap_K_V as H;
H counts = H(nullValue=0);
for(string p : pets)
  ++counts[p];
real x=0, width = 25, height=10;
for(string key : counts) {
  filldraw(box((x,0), (x+width,counts[key]*height)), lightblue);
  label(key, (x+width/2,0), 3S);
  x += width;
}
```



Iterator utilities: `zip`, `enumerate`.

```
string[] a = {'A','B','C'};
int[] b = {1,2,3};

from collections.zip2(K=string, V=int) access zip, operator cast;

for(var z : zip(a,b))
  write(z.k + ' = ' + string(z.v));
```

A = 1
B = 2
C = 3

Struct Enhancements: Bracket Indexing

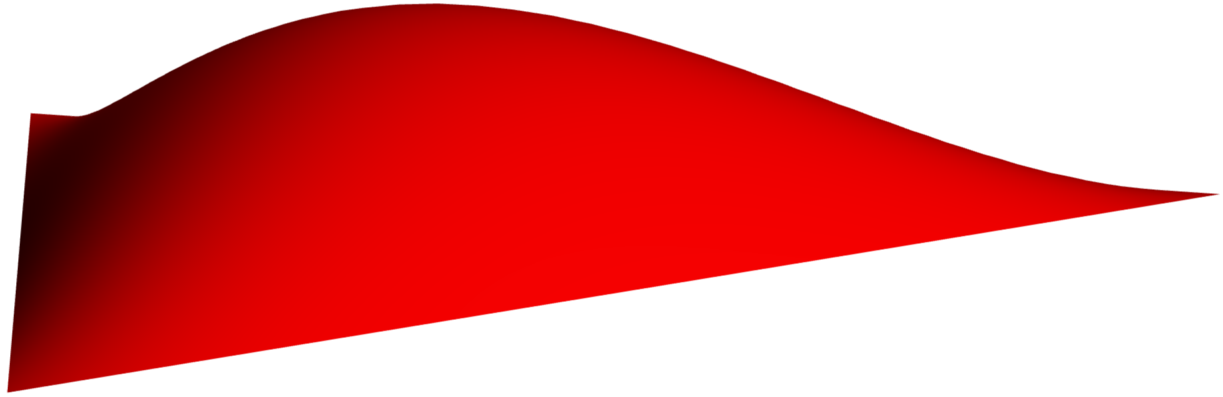
- Custom indexing through `operator[]` and `operator[=]`:

```
struct MyContainer {
  int[] data;
  int operator[](int i) { return data[i]; }
  void operator[=](int i, int v) { data[i] = v; }
}

MyContainer c = new MyContainer;
c.data = {10, 20, 30};
write(c[1]);   // writes 20
c[1] = 99;     // modifies data[1]
```

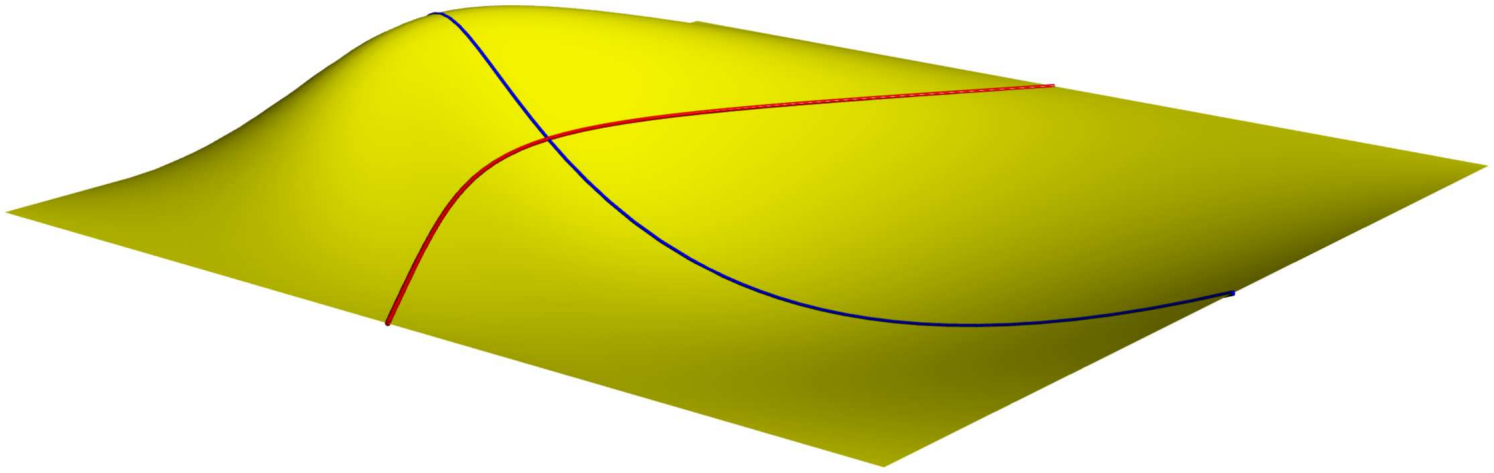
Bézier Surfaces

- Efficient in-house adaptive rendering algorithms have been developed for Bézier triangles and patches.
- Bézier triangles are the natural generalization of Bézier curves to surfaces, just as triangles are the natural generalization of straight lines to planar surfaces.
  - A Bézier triangle can be conveniently expressed in Barycentric coordinates.
- Bézier patches are the natural generalization of quadrilaterals to nonplanar surfaces.
  - A Bézier patch is the direct product of two Bézier curves.

Bézier Triangle



Bézier Patch



Product Representation Compact (PRC)

[ISO/TC171/SC2 2009]

- Only viewable with Adobe Acrobat — no longer widely supported.
- No adaptive renderers available.
- Missing key features:
  - No 3D transparency
  - No vertex-dependent colors
  - No Bezier triangle support

Vector 3D (V3D) Format

- Replaces the legacy PRC format
- Portable compressed vector graphics for 3D scenes
- Embeddable in PDFs via `settings.v3d=true`
- Importable back into `Asymptote`

V3D Features

- 3D transparency
- vertex-dependent colors
- Bezier patches and triangles
- primitives: sphere, hemisphere, disk, cylinder, tube
- triangle groups
- 3D Bezier curves
- 3D pixels

V3D Viewers

- Viewable standalone with `asy -V file.v3d`
- Python `pyv3d` library for programmatic access
- A V3D-aware plugin for Okular enables interactive 3D viewing within PDF documents:

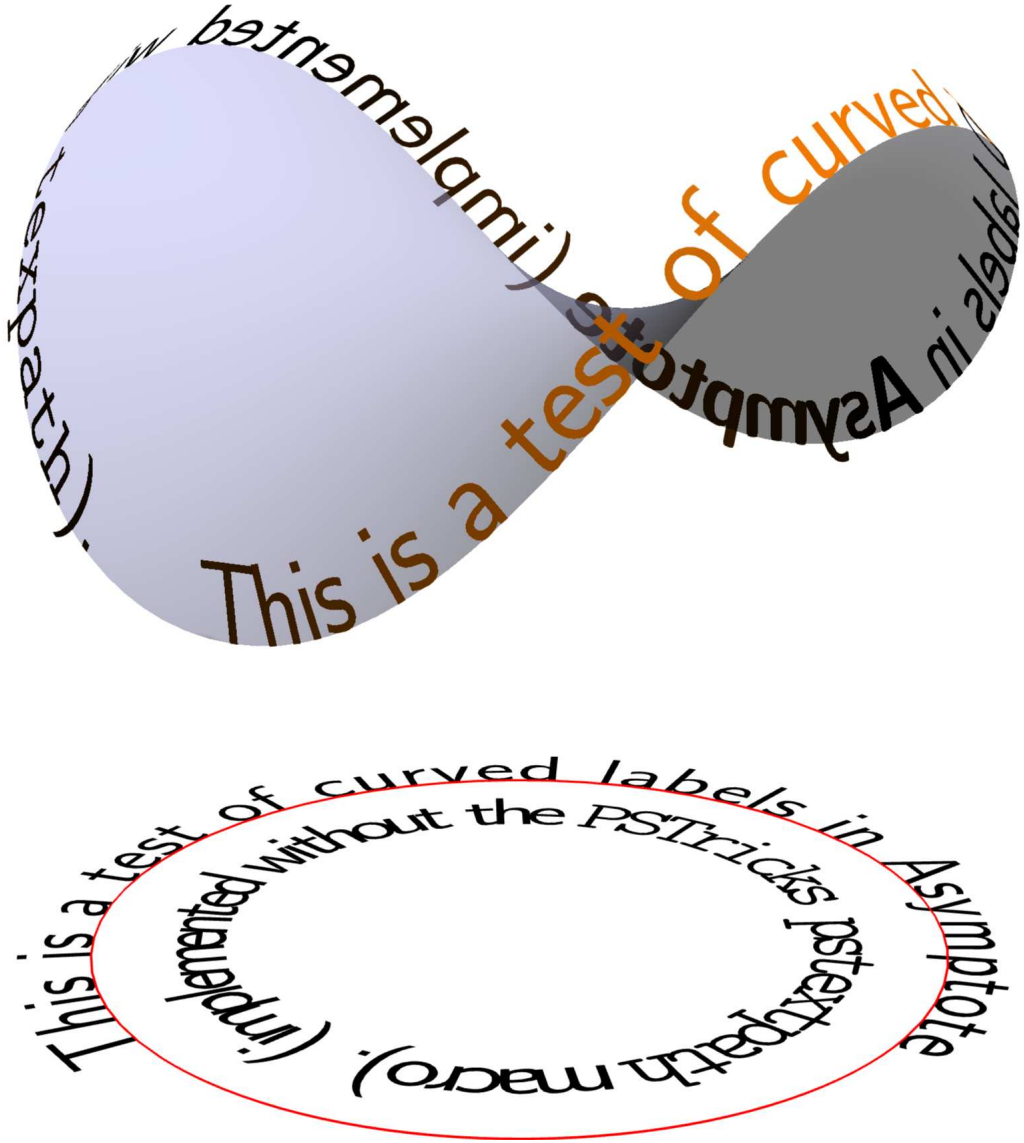
  https://github.com/vectorgraphics/Okular-v3d-Plugins
- Online JavaScript-based V3D PDF viewer:

  https://github.com/vectorgraphics/pdfv3dReader

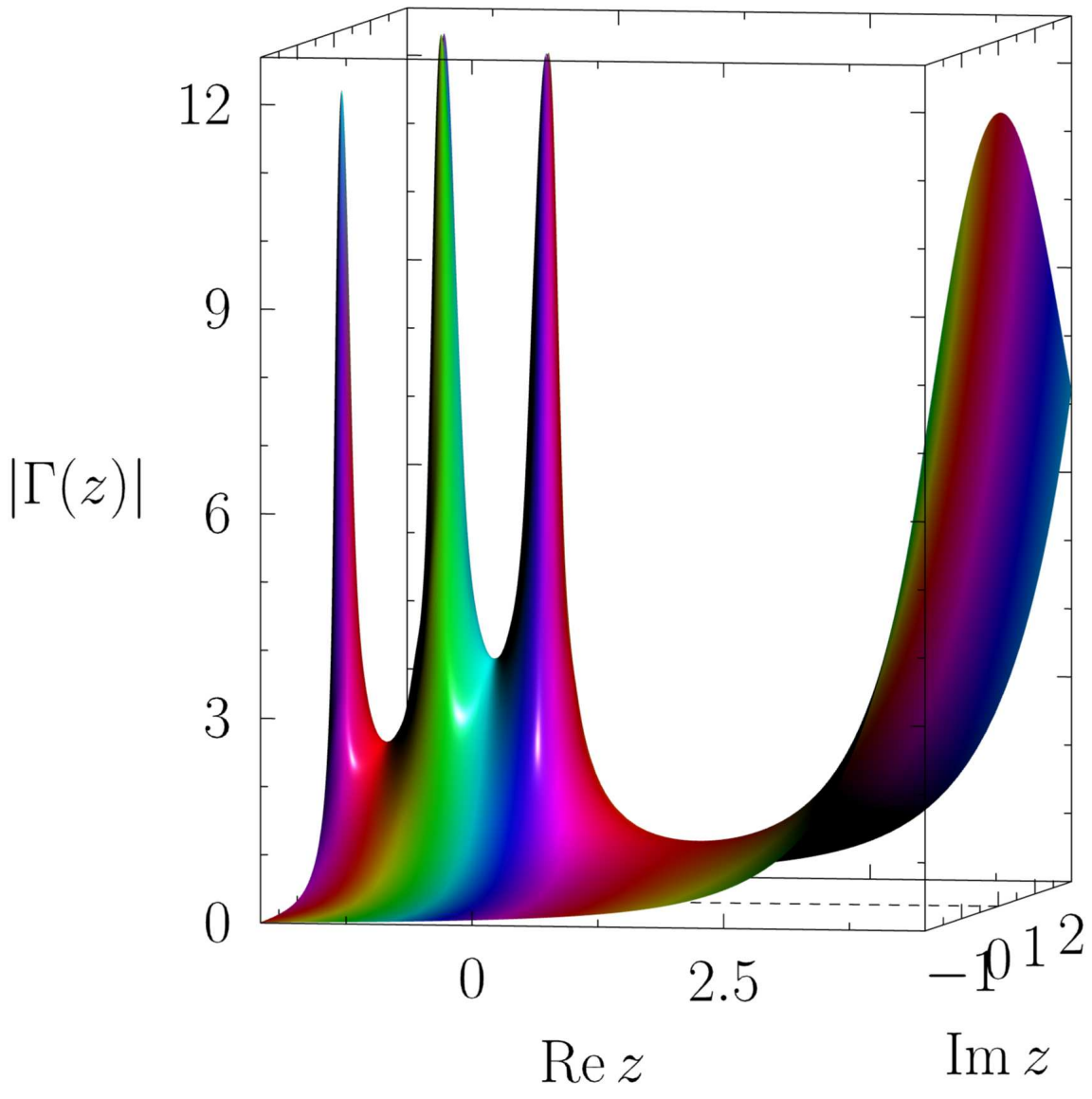
Lifting TeX to 3D with V3D

$$\int_{-\infty}^{+\infty} e^{-\alpha x^2}\,dx = \sqrt{\frac{\pi}{\alpha}}$$

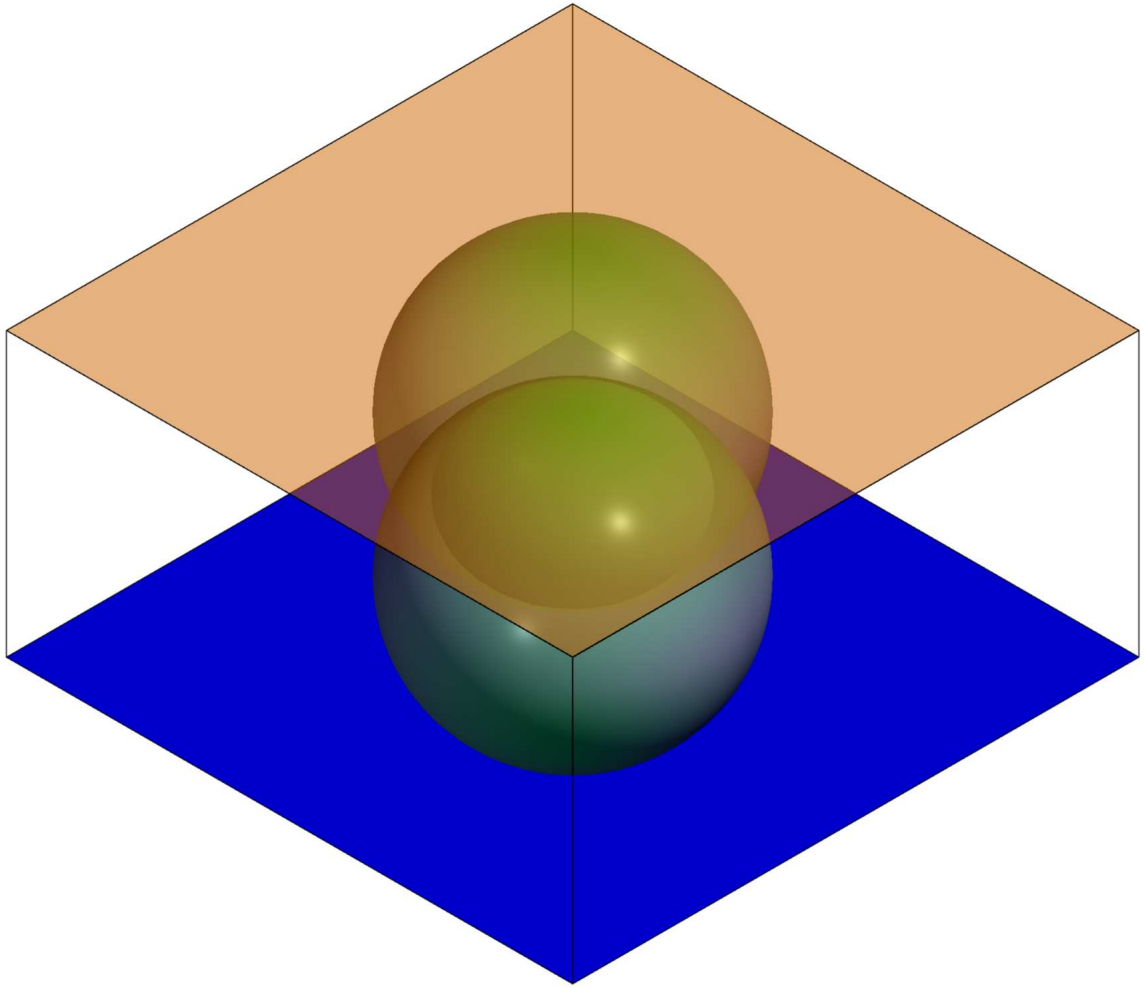
Curved 3D Labels



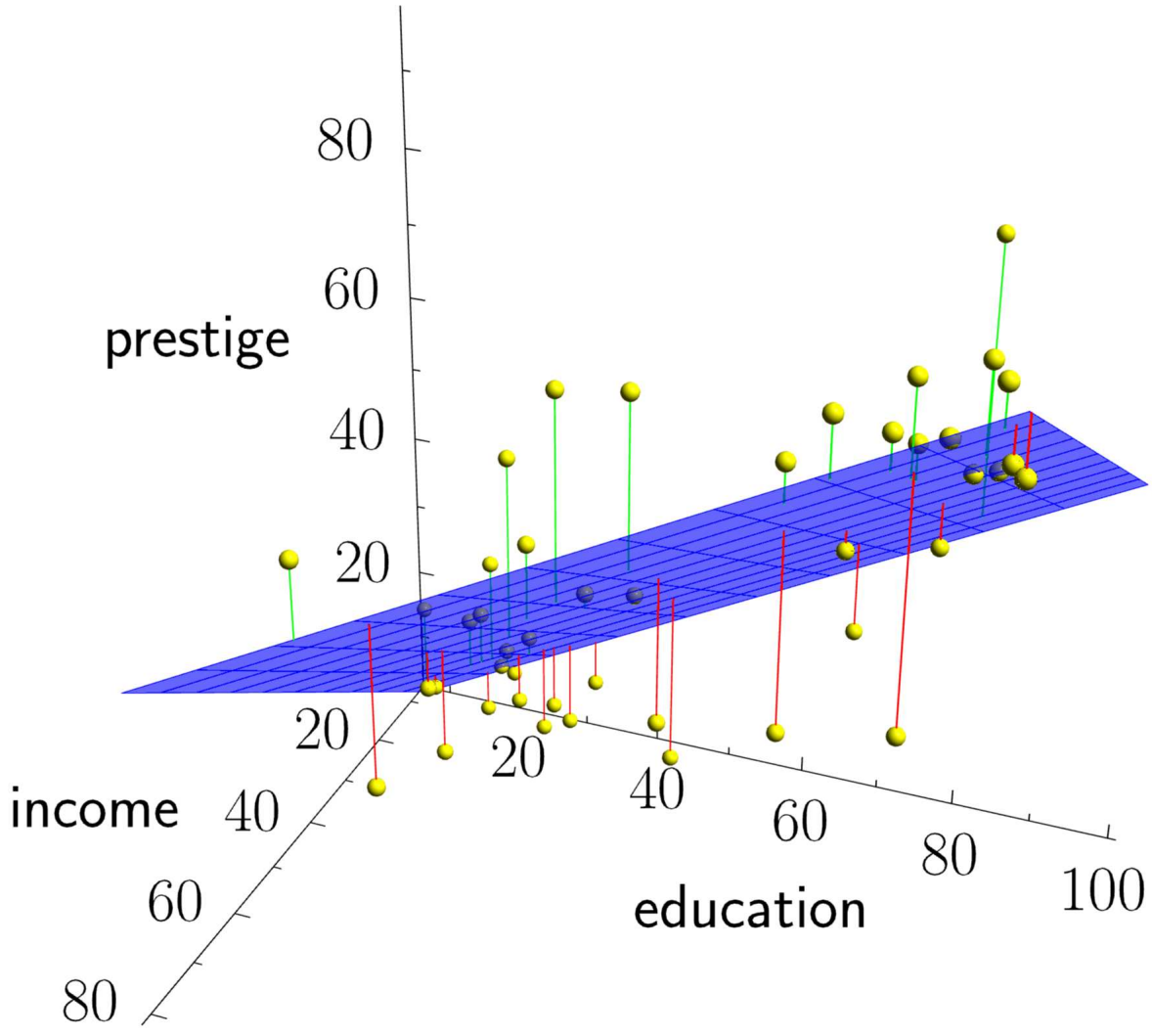
Vertex-Dependent Colors



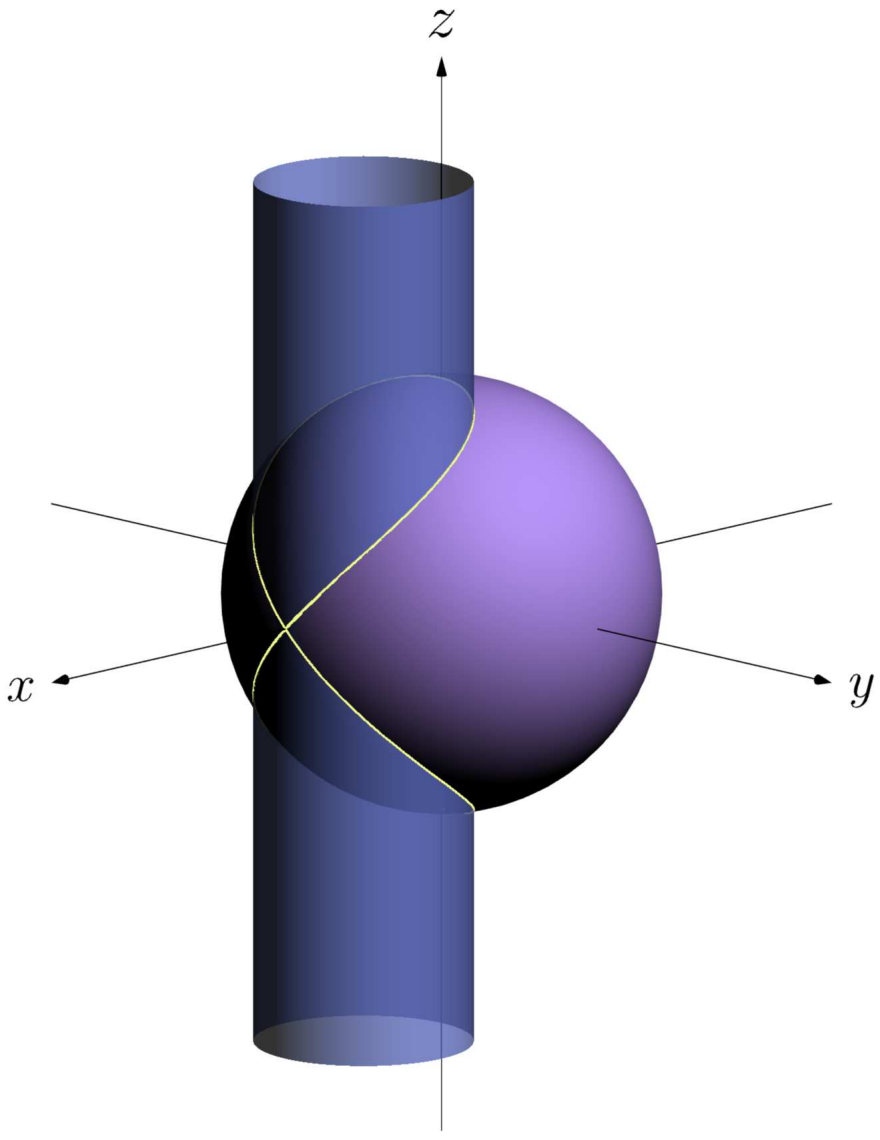
3D Transparency in PDFs



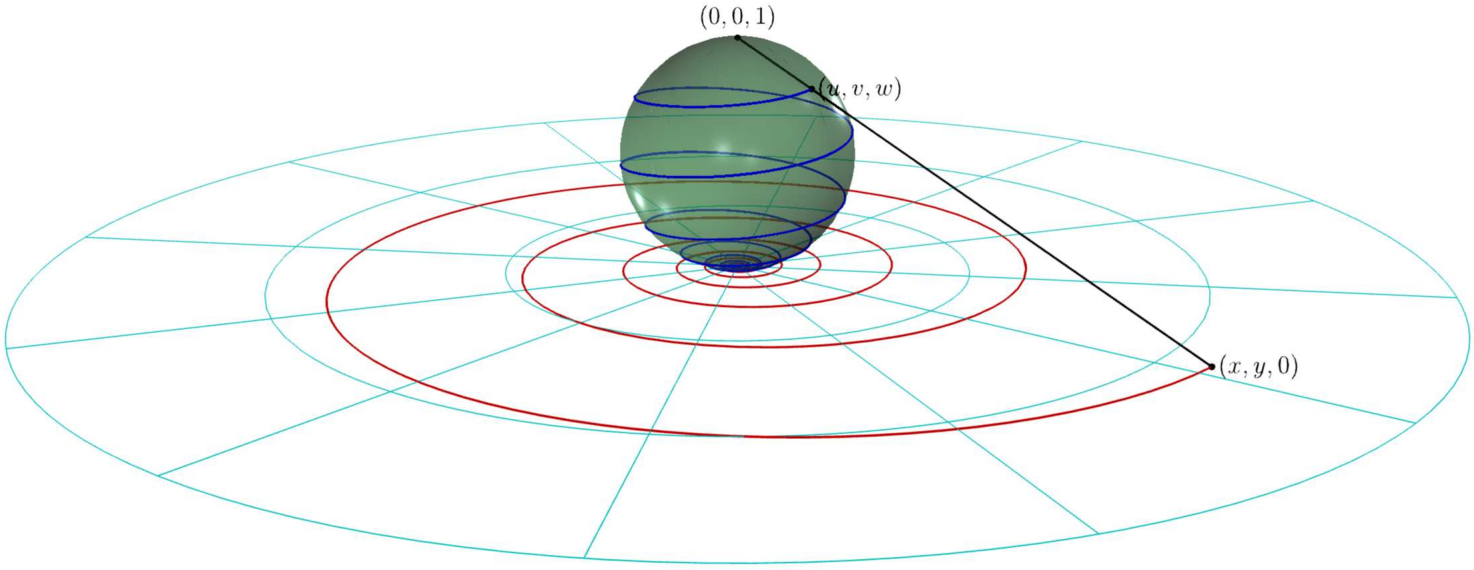
Linear Regression



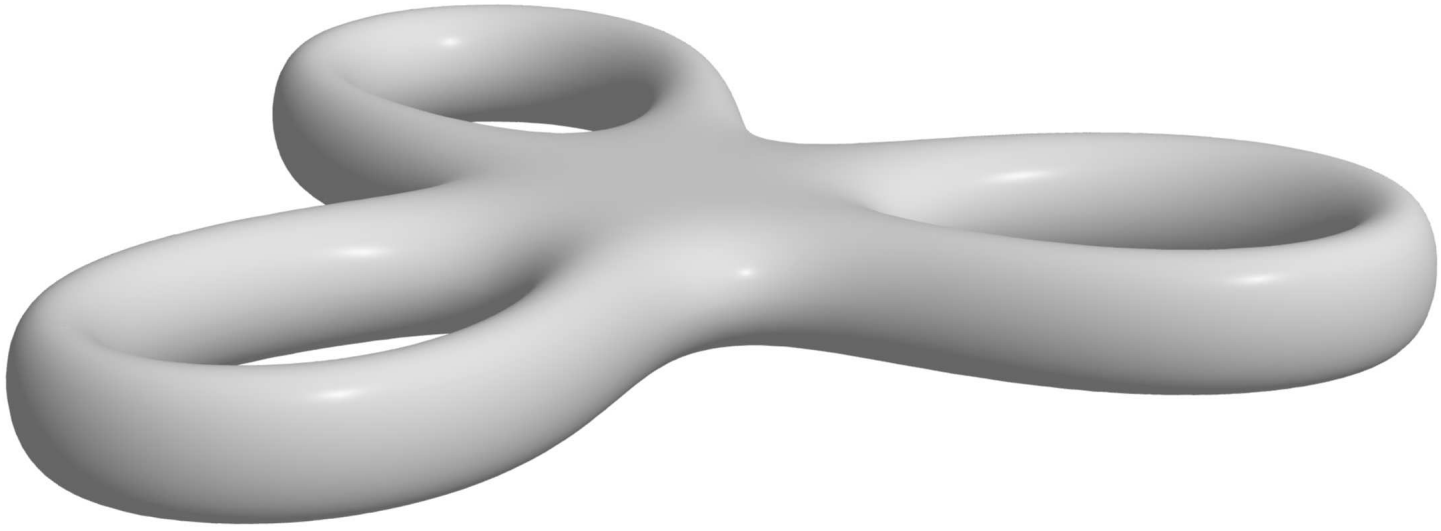
Viviani's Curve



Riemann Sphere



Genus Three Surface



Rendering Pipeline: OpenGL → Vulkan

- The legacy OpenGL renderer is being superseded by a Vulkan port.
- Key advantages:
  - better GPU utilization
  - consistent rendering across operating systems
  - order-independent transparency under macOS, Linux, and Windows

AsyGL: Interactive 3D in HTML

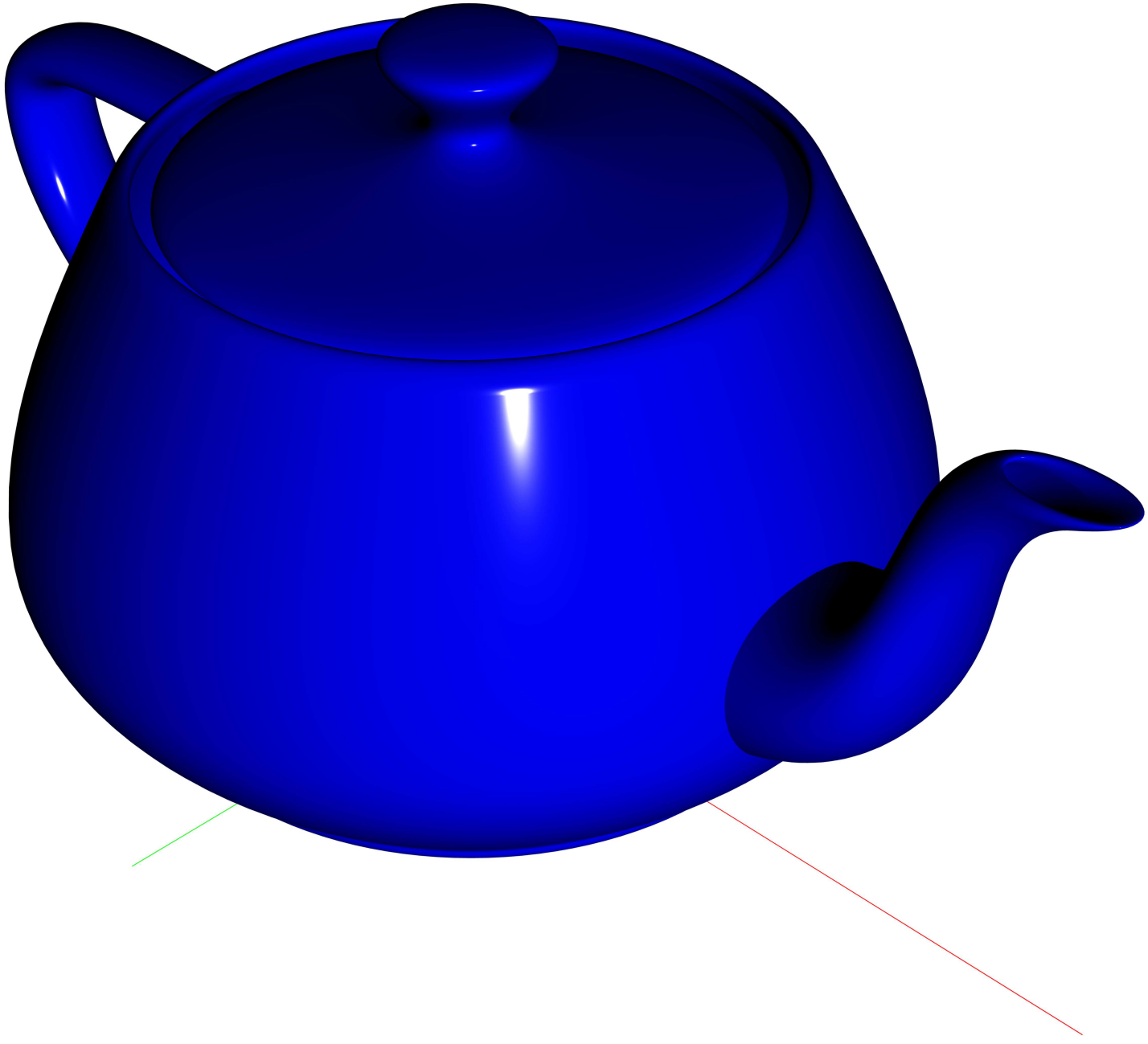
- `Asymptote` can embed interactive 3D figures directly in HTML.
- The AsyGL library is either:
  - downloaded from the network
  - embedded in a stand-alone file
- WebGL rendering works on tablets and phones.
- Supports animations with user-controlled parameters via JavaScript.
- No order-independent transparency in WebGL2 (no SSBOs)!

WebGL Animation Example

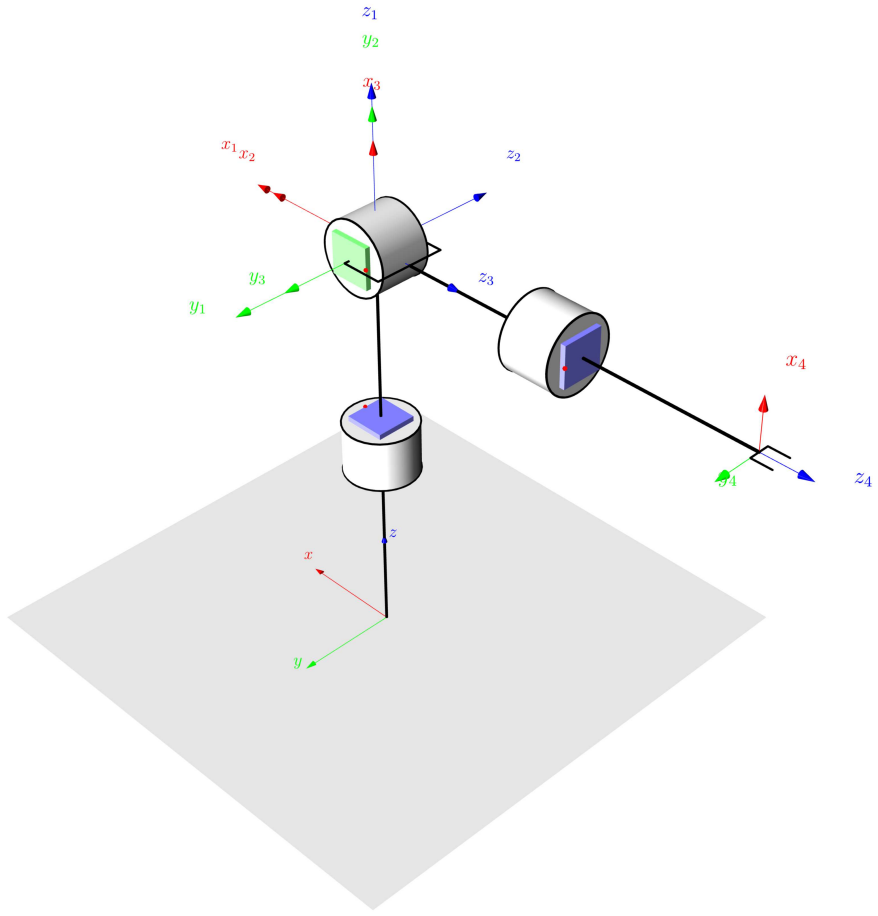
```
beginTransform(geometry="
function(x,t) {return [x[0]-xmax*t,x[1]-ymax*t,x[2]];}",
               color="
function(x,c,t) {return interp(c,green,t);}",
               7);
draw(surface, blue);
endTransform();
```

- JavaScript bridge provides interactive control panel with sliders.

Teapot Animation



Robot Arm [contributed by asmwarrior]



Rendering Pipeline: WebGL → WebGPU

- The upcoming migration from WebGL to WebGPU will bring:
  - order-independent transparency in web browsers
  - compute shader support for more complex effects
  - improved performance on modern GPUs

Image-Based Lighting

- Realistic lighting using pre-rendered environment maps:

  `settings.ibl=true;`

  Uses EXR images from the `ibl` directory or a CDN URL for WebGL.

- Produces physically-based rendering with accurate reflections.

Image-Based Lighting

Xasy: Qt6 Upgrade

- The Xasy graphical front end has been upgraded to Qt6.
- Xasy allows interactive editing of Asymptote figures:
  - modify existing graphical objects by dragging
  - draw new objects with the mouse
  - save modified figures as normal Asymptote files
- It combines the best features of script-driven and GUI-based figure creation.

Summary of New Features: Language

- templated imports, `using`, `autounravel`
- collections library with hash maps, B-tree maps, sets, queues
- `operator iter`, bracket indexing, native hashing

Summary of New Features: Rendering

- Vulkan port with order-independent transparency
- AsyGL for interactive WebGL in HTML
- Image-based lighting
- Bézier triangle patches
- V3D portable format with PDF embedding
- Okular V3D plugin

Future

- Migration from WebGL to WebGPU will
  - extend order-independent transparency to mobile platforms
  - complete the transition to modern GPUs
- V3D animations
- Support accessibility
- Continued evolution as a vector graphics language complementing TEX.

References

[Bowman & Shardt 2009] J. C. Bowman & O. Shardt, TUGboat: The Communications of the TEX Users Group, **30**:58, 2009.

[Bowman 2007] J. C. Bowman, Proceedings in Applied Mathematics and Mechanics, **7**:2010021, 2007.

[Bowman 2010] J. C. Bowman, TUGboat: The Communications of the TEX Users Group, **31**:203, 2010.

[Hobby 1986] J. D. Hobby, Discrete Comput. Geom., **1**:123, 1986.

[ISO/TC171/SC2 2009] ISO/TC171/SC2, 3D use of product representation compact (PRC) format, 2009, http://pdf.editme.com/files/PDFE/SC2N570-PRC-WD.pdf.

[Knuth 1986a] D. E. Knuth, *The TEXbook*, Addison-Wesley, Reading, Massachusetts, 1986.

[Knuth 1986b] D. E. Knuth, *The METAFONTbook*, Addison-Wesley, Reading, Massachusetts, 1986.

Asymptote: 2D & 3D Vector Graphics Language

http://asymptote.sf.net

(freely available under the LGPL license)

Local Inference Manager

https://github.com/statefullm/lim