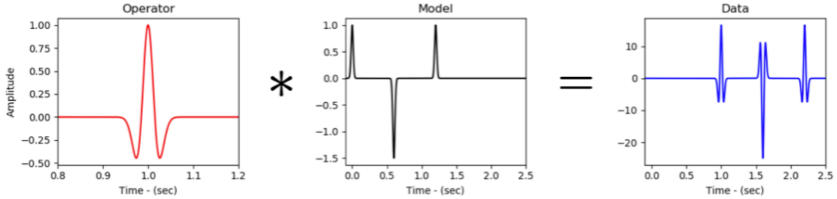
Solving (geophysical) inverse problems on GPUs with PyLops + CuPy

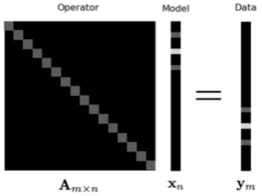
M. Ravasi
L. Mosser
D. Vargas

software underground presents
Subsurface Rendezvous

Inverse Problems

$$\mathbf{y = Gx}$$



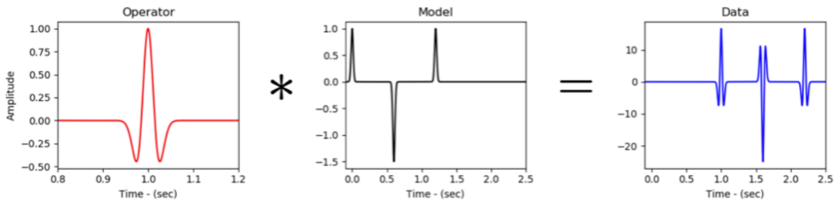


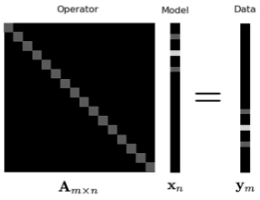
Numpy/scipy

```
# Model
x = np.*(nt)

# Operator
G = convmtx(g, nt)

# Data
y = np.dot(G, x)

# Inverse
xinv = sp.sparse.linalg.*(G, x)
```

Inverse Problems

$$\mathbf{y = Gx}$$



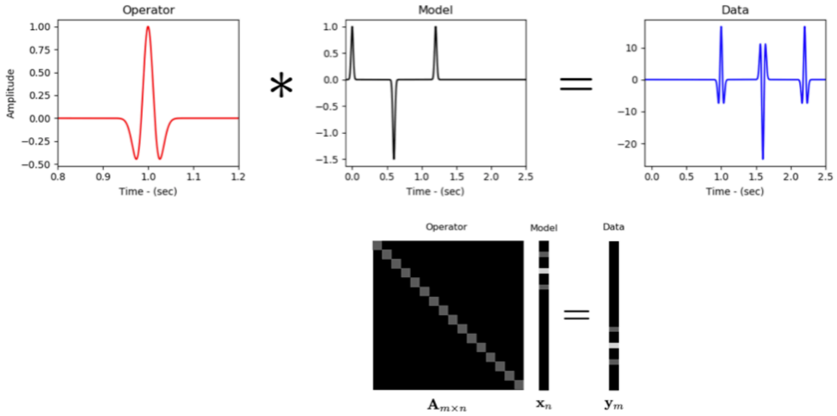


PyLops CPU

```
# Model
x = np.*(nt)

# Operator
G = Convolve1D(nt, g)

# Data
y = G * x

# Inverse
xinv = sp.sparse.linalg.*(G, x)
```

Inverse Problems

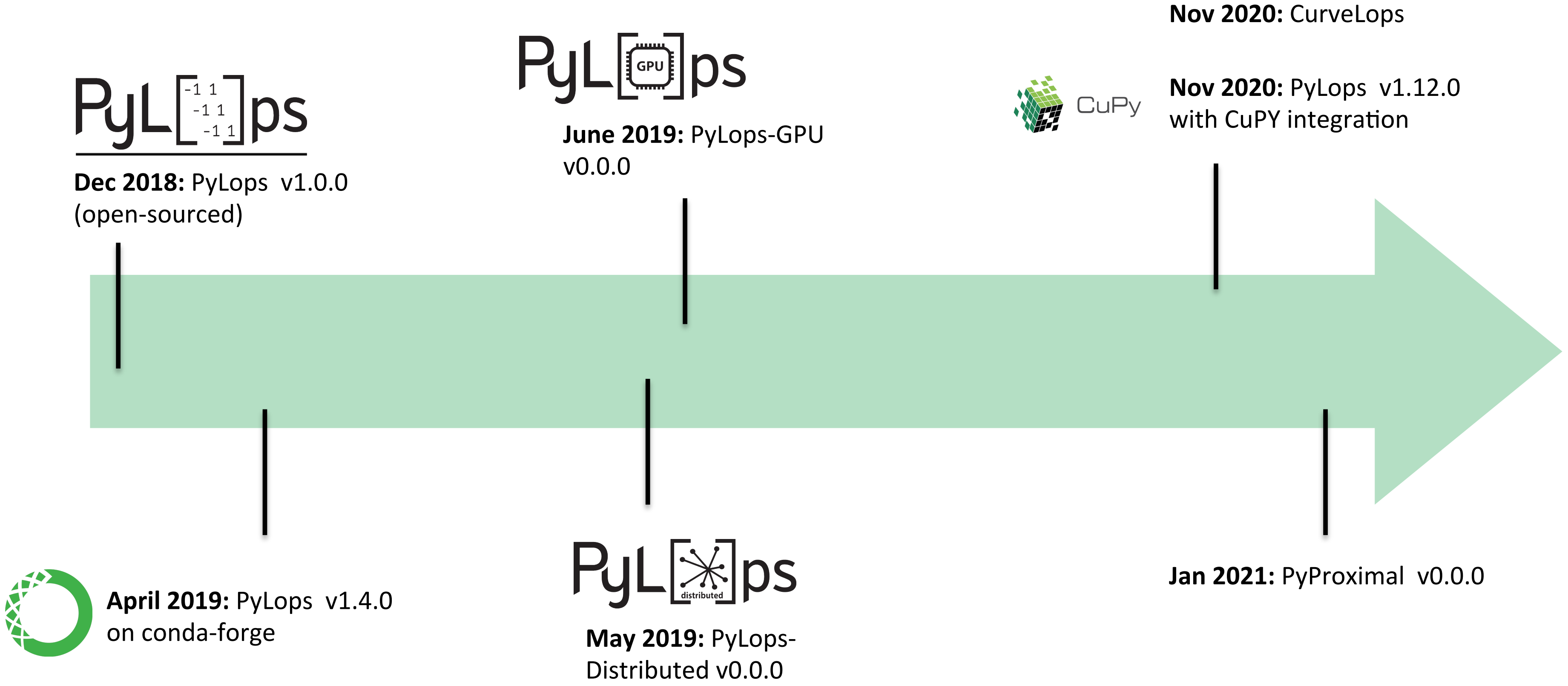
$$\mathbf{y} = \mathbf{G}\mathbf{x}$$



PyLops GPU

```
# Model
x = cp.*(nt)

# Operator
G = Convolve1D(nt, cp.asarray(g))

# Data
y = G * x

# Inverse
xinv = pylops.optimization.*(G, x)
```

History



Dec 2018: PyLops v1.0.0 (open-sourced)

April 2019: PyLops v1.4.0 on conda-forge

May 2019: PyLops-Distributed v0.0.0

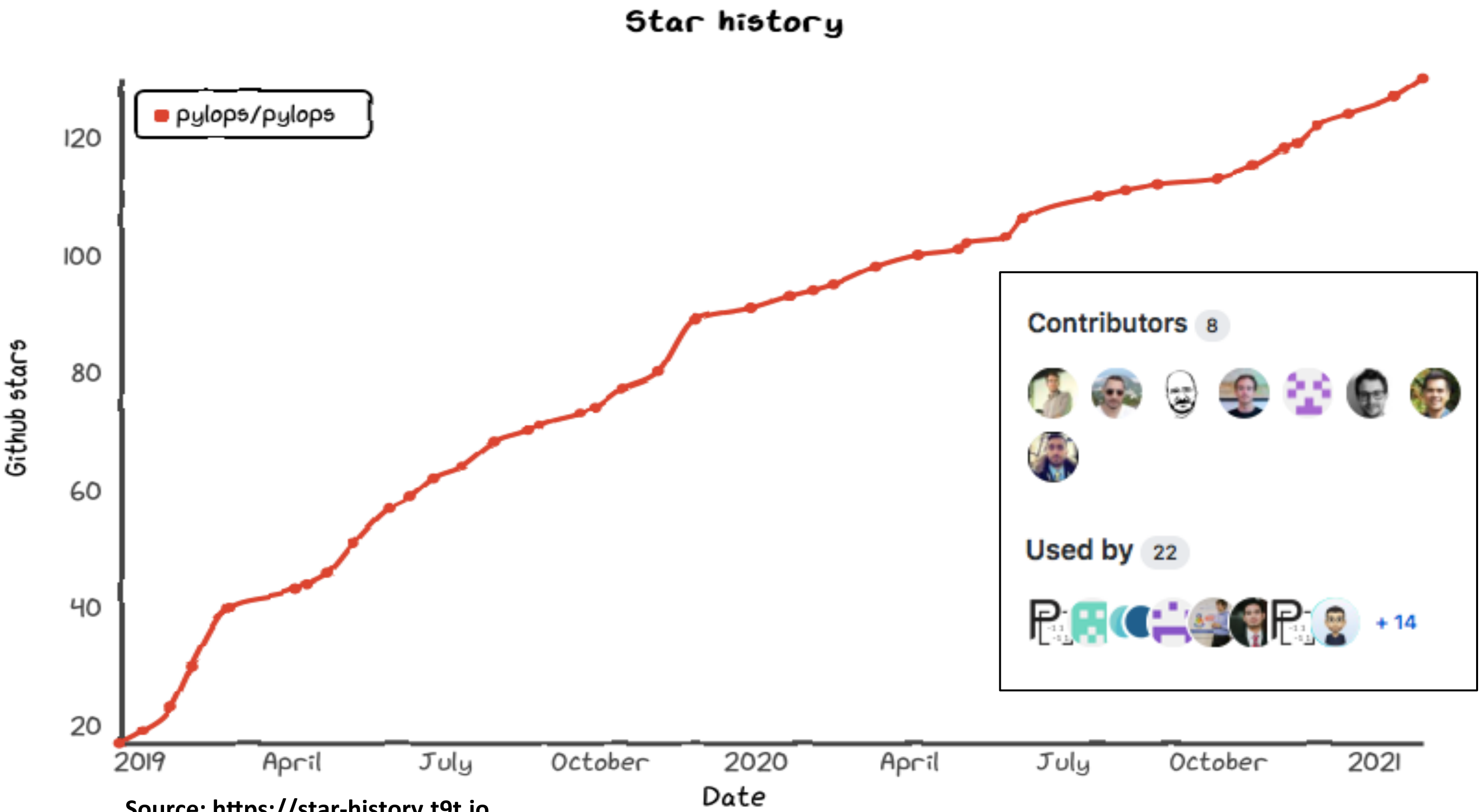
June 2019: PyLops-GPU v0.0.0

Nov 2020: CurveLops

Nov 2020: PyLops v1.12.0 with CuPY integration

Jan 2021: PyProximal v0.0.0

History



Contributors 8

Used by 22

+ 14

Source: https://star-history.t9t.io

Summary

An entire ecosystem of tools to ease research in inverse problems

-> +50 Operators (CPU + GPU)

-> Least-squares, L1 (sparsity), and Proximal solvers

-> Sparsity transforms (e.g., FFT, FFTN, DWT, Curvelet, Seislet)

-> Dask integration for distributed operators

-> PyTorch integration for Autograd (and GPU)

-> Visit: https://github.com/PyLops