HEAd '20

Self-Contained Jupyter Notebook Labs Promote Scalable Signal Processing Education

Dominic Carrano, Ilya Chugunov, Jonathan Lee, Babak Ayazifar

University of California, Berkeley

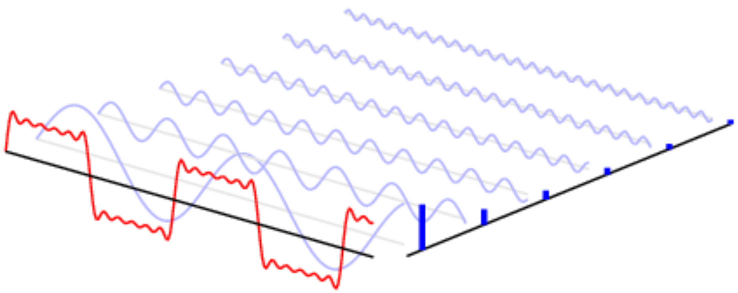
Where Our Course is Situated

Python programming
Complex numbers
Calculus
Linear algebra

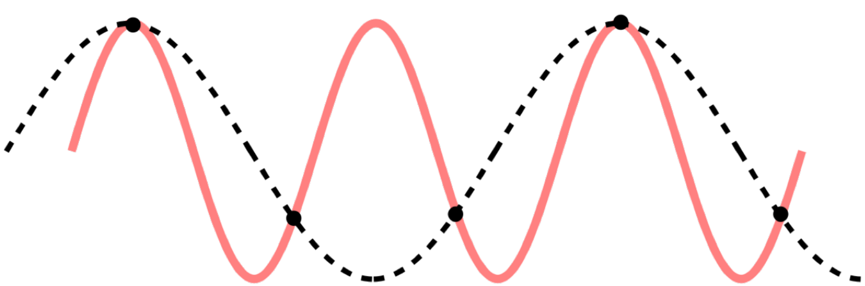
LTI systems
Fourier analysis
A-to-D sampling
Z and Laplace Transforms

Signals & Systems

Fourier Analysis & Decomposition

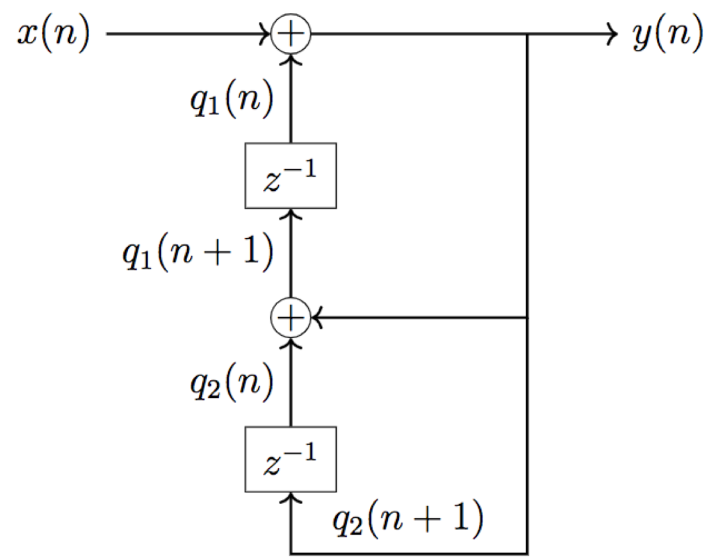


Sampling Theory



Systems: Fibonacci Seq. Generator

$$y(n) = y(n-1) + y(n-2) + x(n)$$



$$x(n) = 1, 0, 0, 0, 0, 0, \ldots$$
$$y(n) = 1, 1, 2, 3, 5, 8, \ldots$$

The Need for Labs

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{i\omega t} d\omega$$

"When will I ever use this in the *real* world?"

Scalable Signals & Systems Labs

Want:

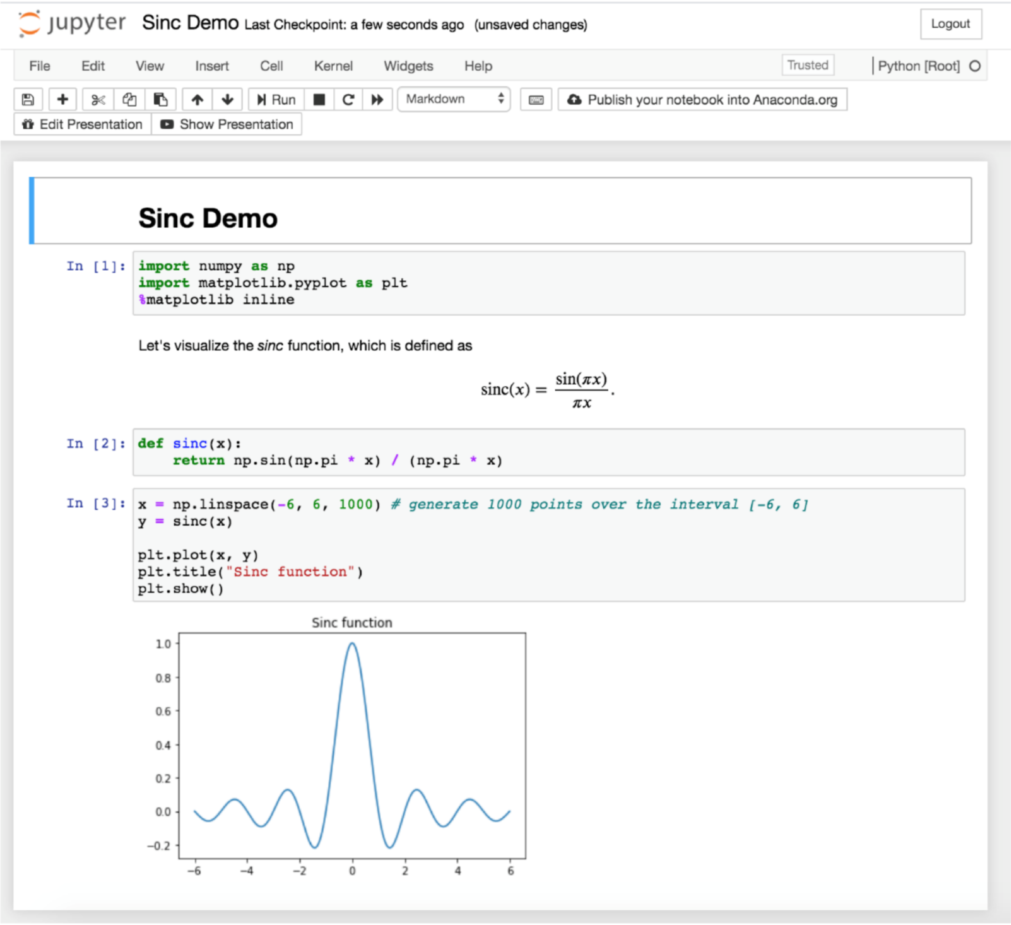
- Applications-driven
- Teach computational skills
- Self-contained

Avoid:

- Need for physical space
- Hardware
- In-person checkoffs

The Jupyter Notebook

- Interactive interface
  - Create and run code cells
  - Seamless integration with plotting/visualization tools
- Runs in browser
- Supports Julia, Python, R, and many other programming languages



The Jupyter Notebook

```
Mathematically, we represent a note with frequency $f$ as a sine wave at
that frequency with some amplitude $A$ and phase $\phi$,

$$A \sin(2\pi f t + \phi),$$

where $A$ would be determined by how hard the note on the piano is
physically pressed, for instance, and the phase $\phi$ corresponding to a
temporal offset is typically just taken to be zero.
```

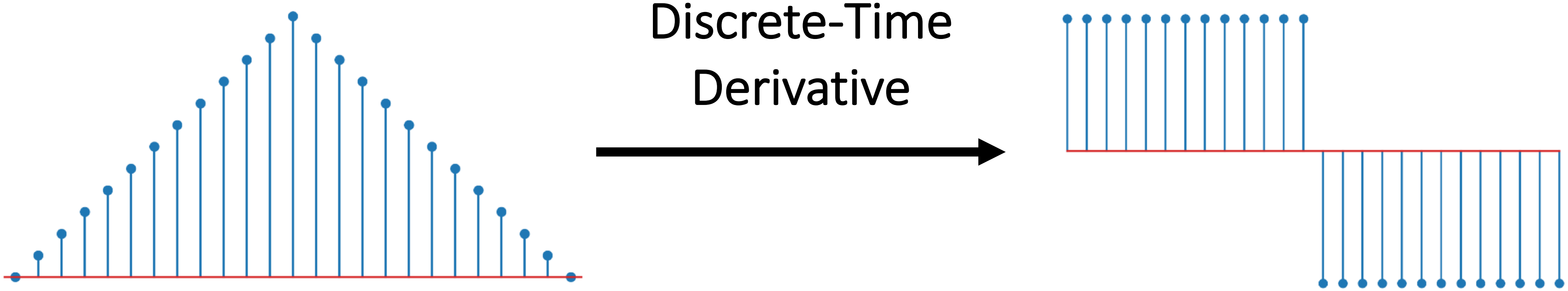
↓

Mathematically, we represent a note with frequency f as a sine wave at that frequency with some amplitude A and phase ϕ,

$$A \sin(2\pi f t + \phi),$$

where A would be determined by how hard the note on the piano is physically pressed, for instance, and the phase ϕ corresponding to a temporal offset is typically just taken to be zero.

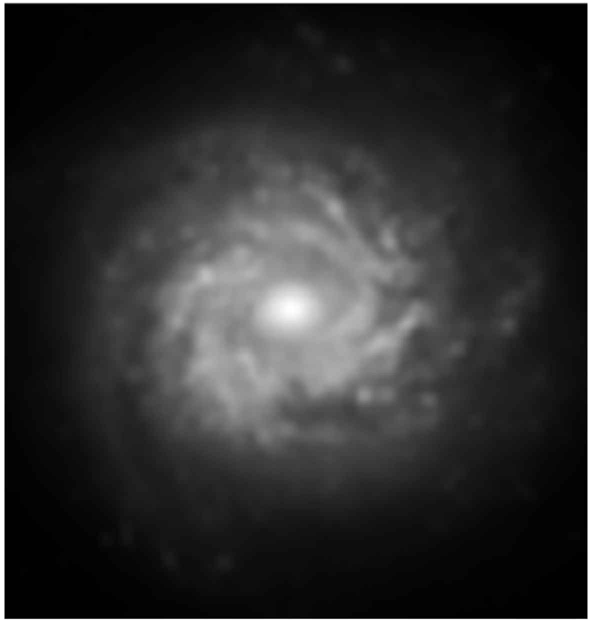
Applications-Driven Labs



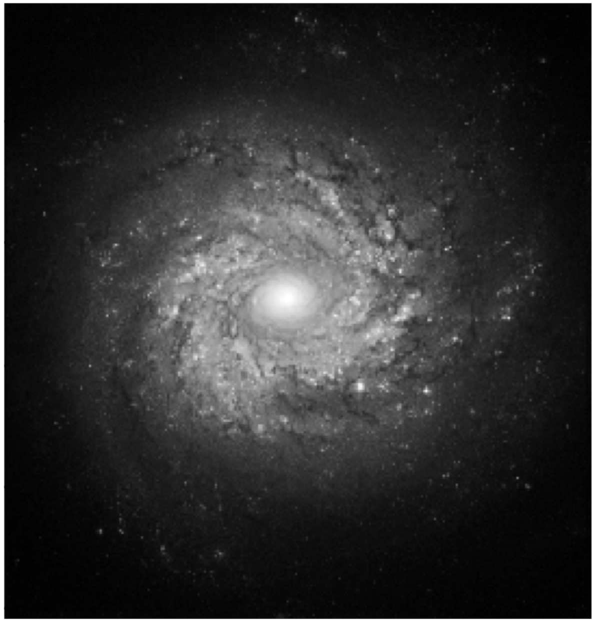
Finite-Difference Filter for Numerical Differentiation

HEAd '20

Applications-Driven Labs



Deconvolution



Deblur Hubble Space Telescope Image

Teach Computational Skills

```
In [46]: lab1_helper.get_speedup(np.convolve, numpyless_conv)

Without NumPy: 0.5981 sec
With NumPy:    0.0001 sec
NumPy gives a 6334.9323x speedup
```

Use Python libraries to accelerate computation 6,000x

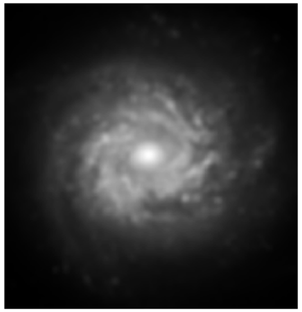
Teach Computational Skills

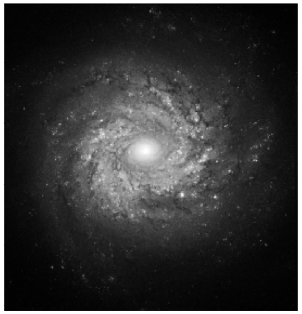
```
(Current Best) RMSE: 12.303, Sigma: 0.204, Kernel Size: 1
```

```
(Current Best) RMSE: 1.962, Sigma: 5.102, Kernel Size: 15

Original RMSE: 12.303
Best RMSE:     1.962
Parametric sweep gives a 6.27x better result.
```

Blurry

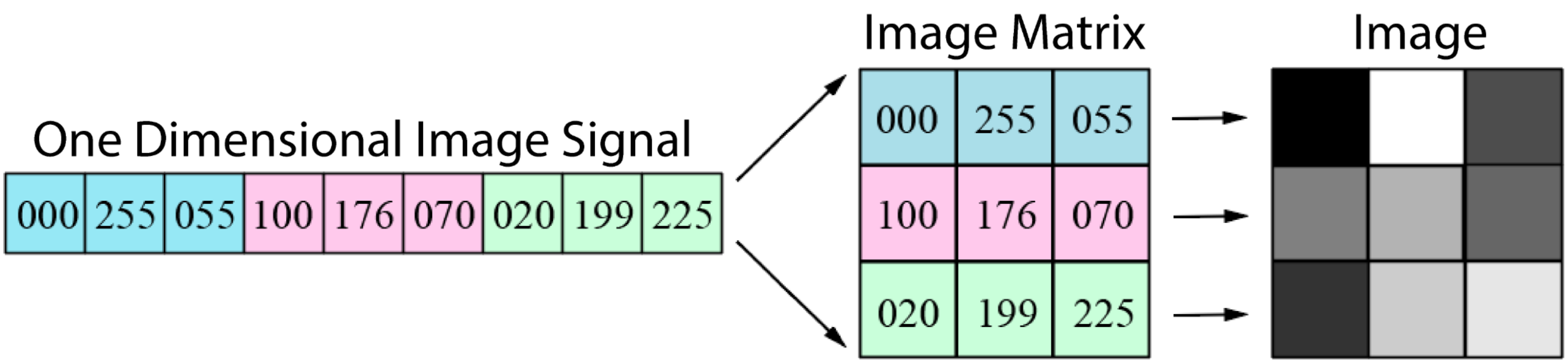




Deblurred

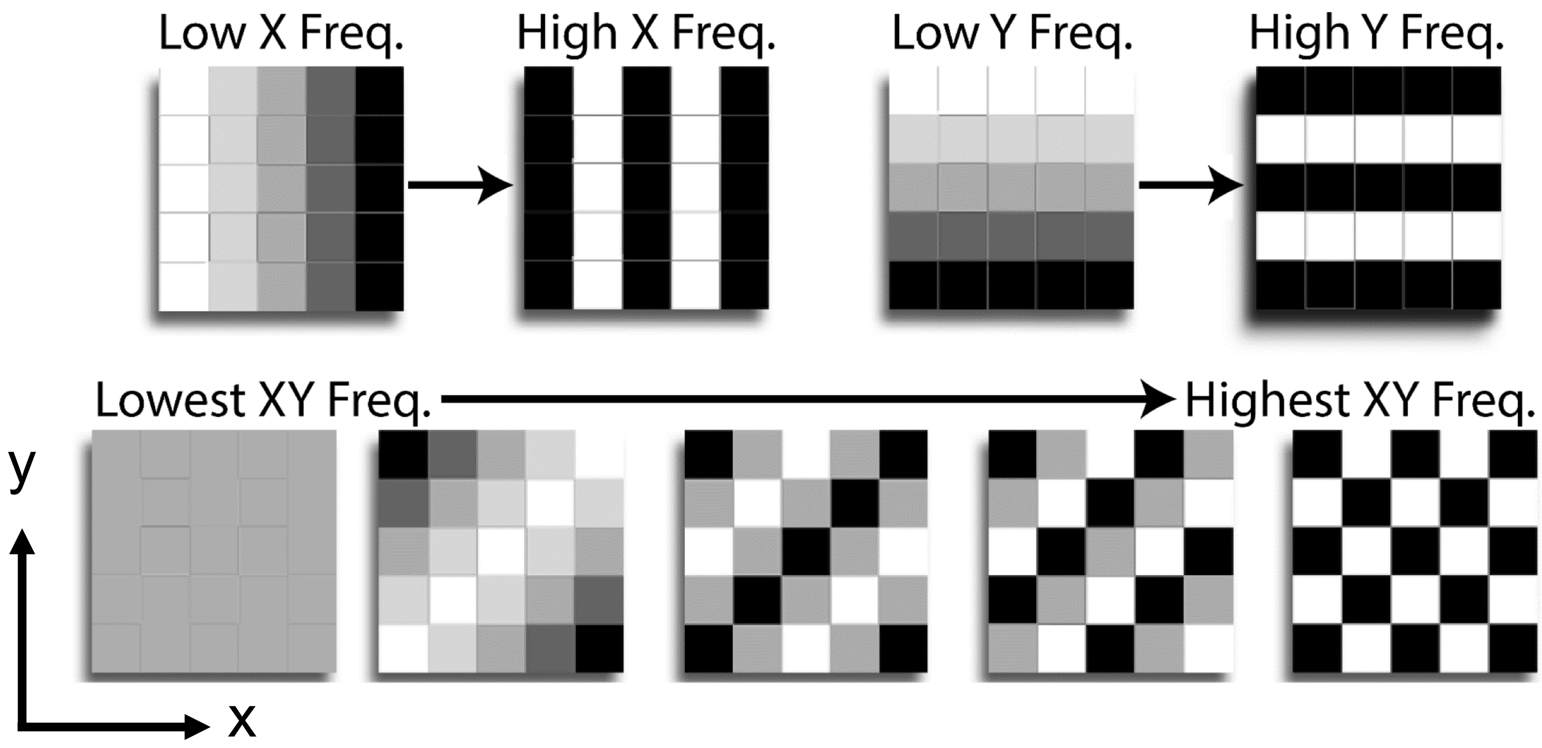
Use a parametric sweep for numerical optimization

Self-Contained Labs



Generalize 1D signals to 2D signals

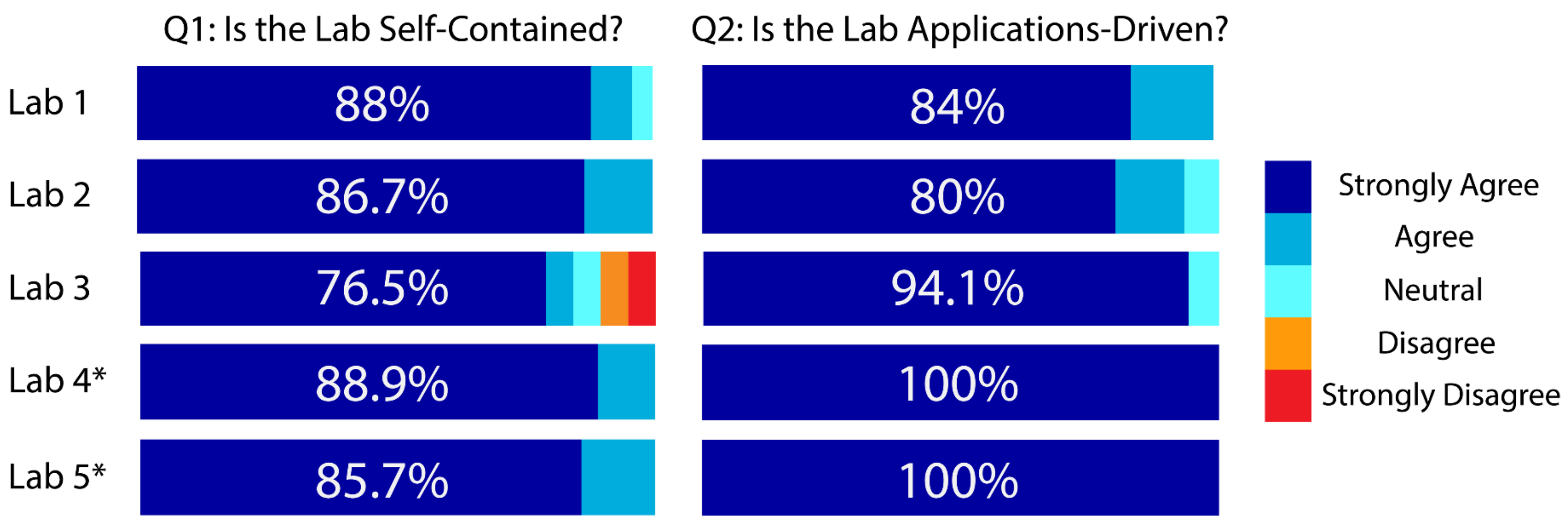
Self-Contained Labs



Generalize 1D frequencies to 2D frequencies

HEAd '20

Student Feedback



*lower response rate due to COVID-19

Spring 2020 student feedback