[PolyMath]

version 1.0 and beyond

Serge Stinckwich, Oleksandr Zaytsev
Hemal Varambhia, Konrad Hinsen
Atharva Khare, Nikhil Pinnaparaju
Werner Kassens

UMMISCO
Unite Mixte Internationale

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS
DEPUIS 1257

IRD
Institut de Recherche pour le Développement
FRANCE

I’m not a mathematician !

Current Status of PolyMath

- A numeric framework for Pharo (written in pure Pharo)

- Ordinary differential equations, Random Number Generators, Linear Algebra, Matrices, Complex Numbers, FFT, Polynomials, Probability distributions, …

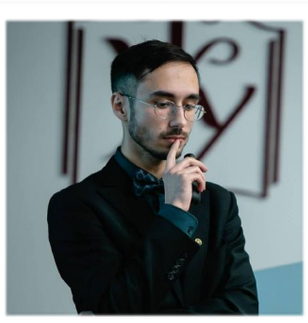
- DataFrames, data analysis on tabular data

- MIT Licence

- PolyMath 1.0 release

- https://github.com/PolyMathOrg/PolyMath

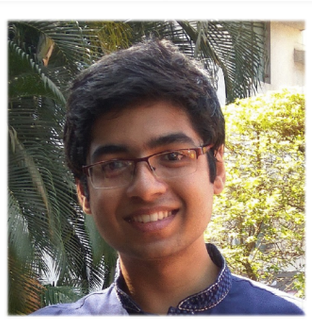
Contributors



Oleksandr Zaytsev

Konrad Hinsen

Hemal Varambhia



Atharva Khare

Nikhil Pinnaparaju

- Thank you all !
- All contributions are valuable (code&documentation)
- Regular meeting on Discord (#polymath channel)

Since last year

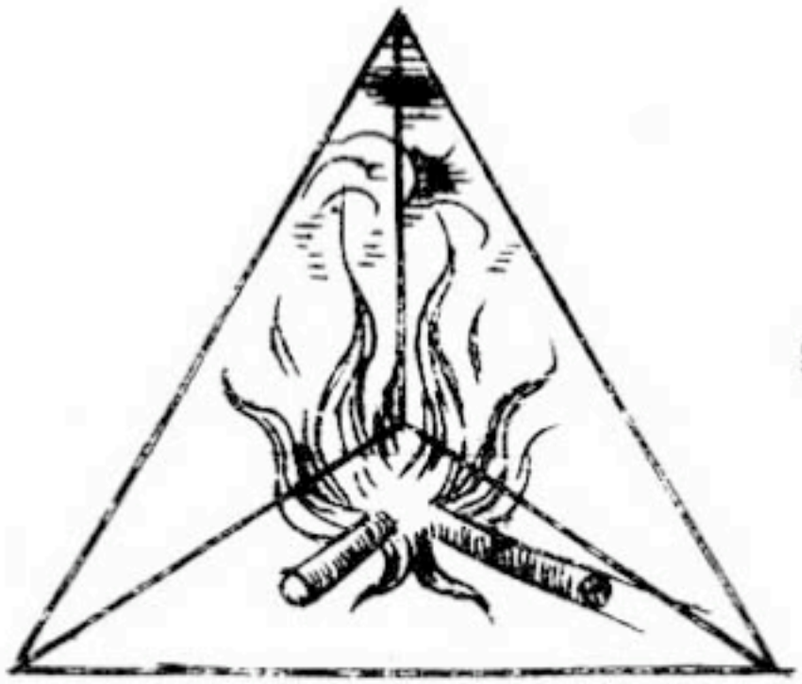
- Release 1.0 version + new logo
- Clean the code (All PM prefix classes), add more unit tests
- PCA and t-SNE, DataFrames (Atharva, Nikhil, Oleks)
- Random Numbers cleaning (Hemal)
- Automatic Differentiation package cleaning (Serge)
- Data Transformer hierarchy (PMStandardizationScaler) like in scikit-learn
- Trunk-based development since 1.0 (all code is committed to master branch + one release branch)
- Add a contributing guide and code of conduct

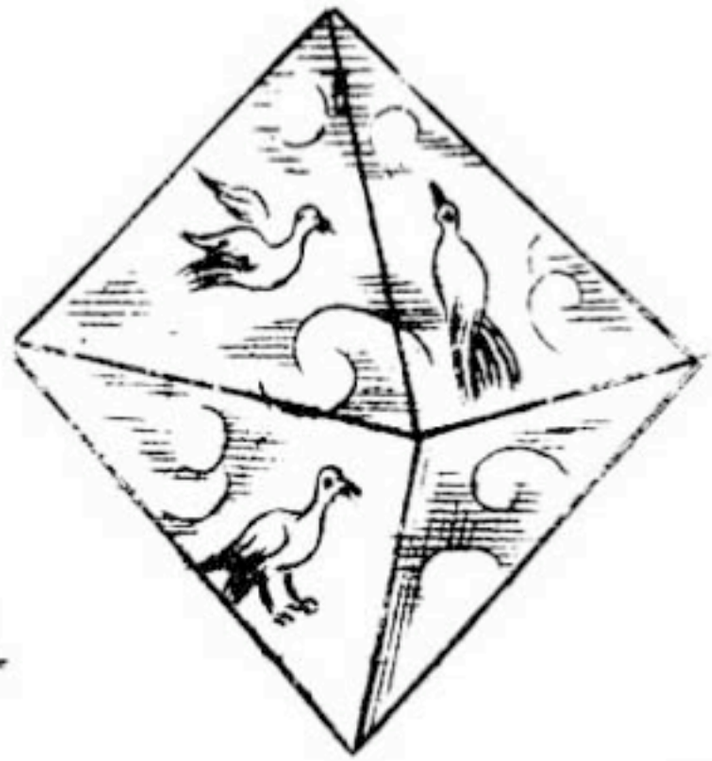
Metrics

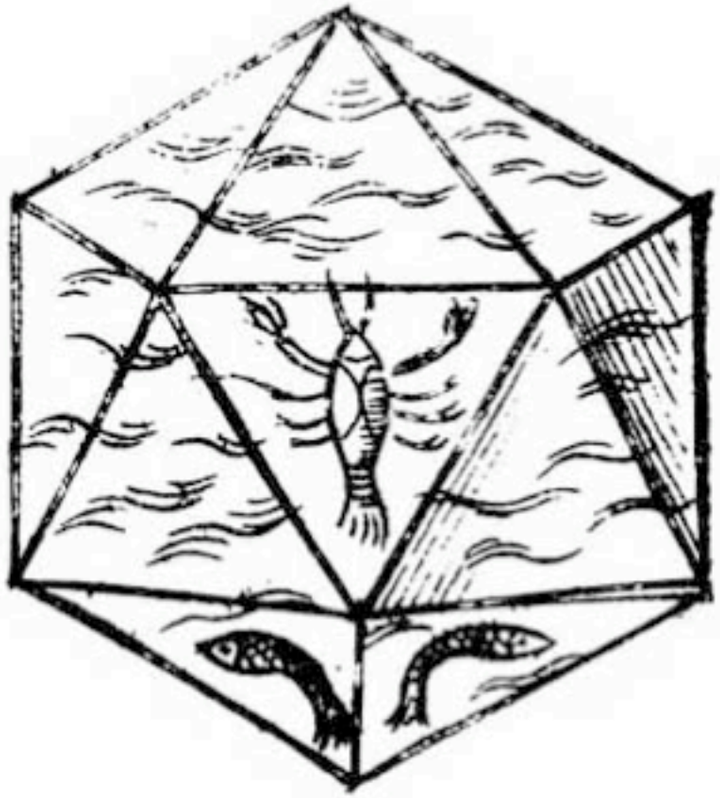
- 300 classes, 59 packages, 24K LOC

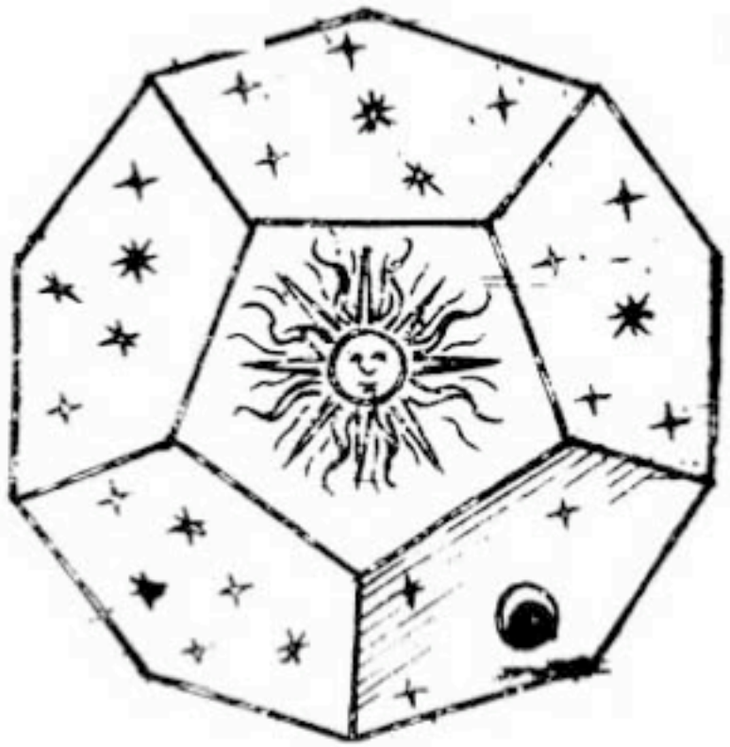
- 806 green unit tests, 1034 commits, 100 stars on github (9th Pharo project on github)

Pharo









[PolyMath]

designed by Lusy Dolia

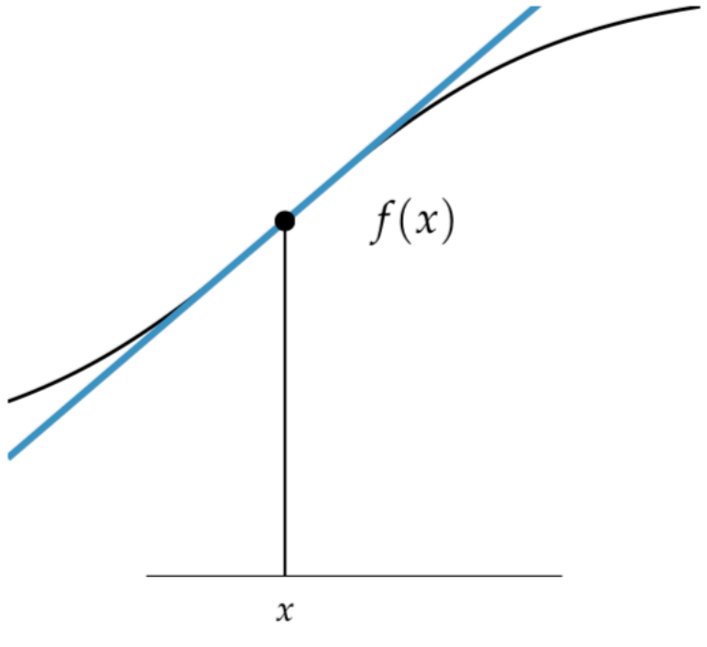
DEMO

Differentiation

Derivatives

- The derivatives f’(x) of a function of a single variable x is the rate at which the value of f changes at x.

- How to compute derivatives numerically ?



$$f'(x) = \frac{\Delta f(x)}{\Delta x}$$

Numerical differentiation by using finite difference

$$f'(x) \approx \underbrace{\frac{f(x+h)-f(x)}{h}}_{\text{forward difference}} \approx \underbrace{\frac{f(x+h/2)-f(x-h/2)}{h}}_{\text{central difference}} \approx \underbrace{\frac{f(x)-f(x-h)}{h}}_{\text{backward difference}}$$

- Theoretically, the smaller the h is, the better the derivative estimates

- Practically very small value of h can result in numerical cancellation errors …

```
:= [ :x| x sin ].
 := 1/2.
iffForward := [:h| (((f value: (x+h)) - (f value: x)) / h)].
rror := [:h| ((diffForward value: h) - x cos) abs].

rror value: 0.1
```

Numerical differentiation by the Complex Step Method

- The Complex Step Method bypass the subtractive errors, by using only one single function evaluation (but on complex numbers).

- Taylor expansion for an imaginary step is:

$$f(x+ih) = f(x) + ihf'(x) - h^2\frac{f''(x)}{2!} - ih^3\frac{f'''(x)}{3!} + \cdots$$

$$\mathrm{Im}(f(x+ih)) = hf'(x) - h^3\frac{f'''(x)}{3!} + \cdots$$

$$\Rightarrow f'(x) = \frac{\mathrm{Im}(f(x+ih))}{h} + h^2\frac{f'''(x)}{3!} - \cdots$$

$$= \frac{\mathrm{Im}(f(x+ih))}{h} + O(h^2) \text{ as } h \to 0$$

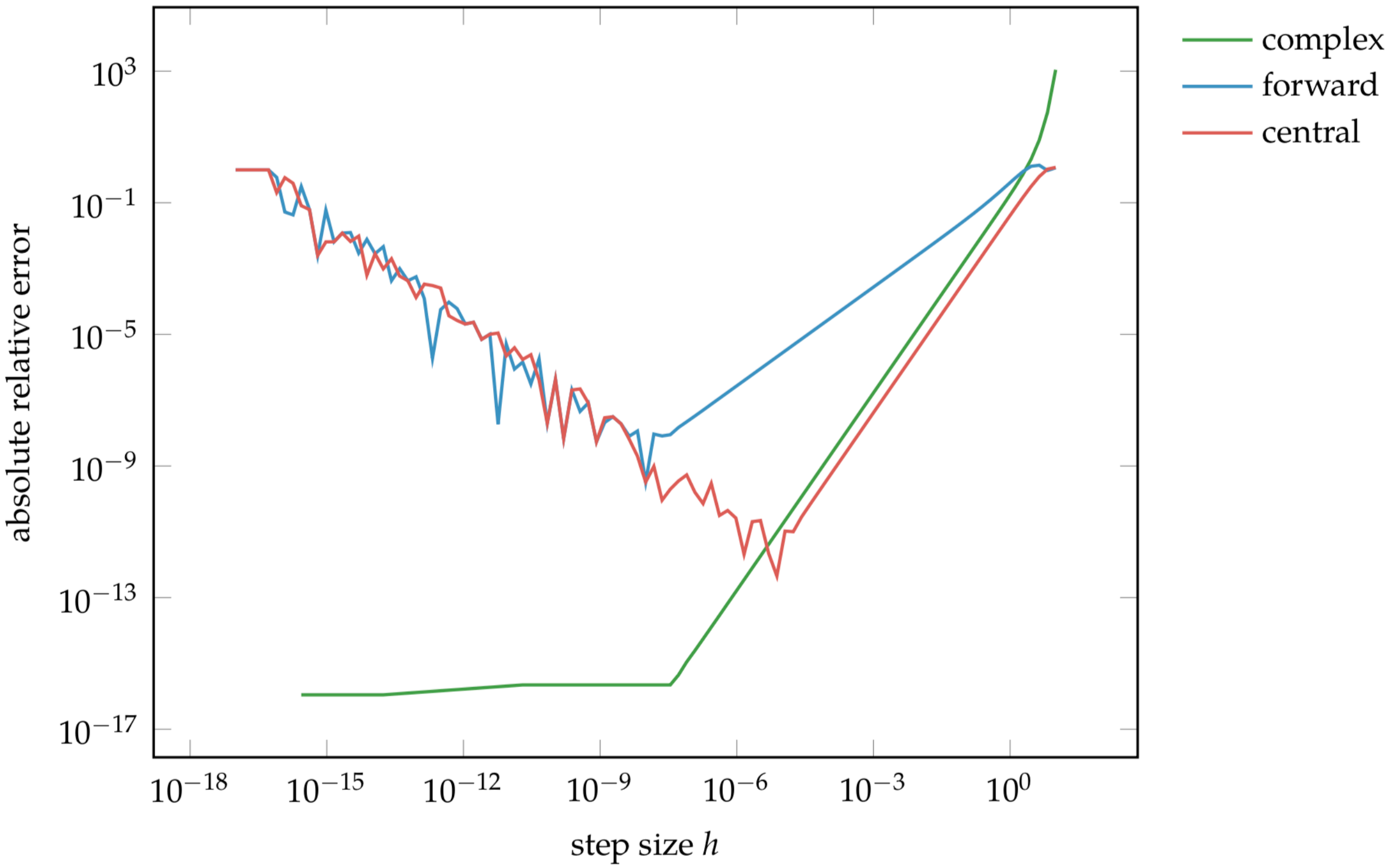
Consider $f(x) = \sin(x^2)$. The function value at $x = \pi/2$ is approximately 0.624266 and the derivative is $\pi \cos(\pi^2/4) \approx -2.45425$. We can arrive at this using the complex step method:

```
f := [ :x| (x*x) sin ].

pi := Float pi.
x := pi /2.
h := 0.001.

((f value: (h*1i +x )) imaginary) / h.

pi * ((pi*pi/4) cos)
```


complex
forward
central
absolute relative error
10^{3}
10^{-1}
10^{-5}
10^{-9}
10^{-13}
10^{-17}
10^{-18}
10^{-15}
10^{-12}
10^{-9}
10^{-6}
10^{-3}
10^{0}
step size h

Automatic Differentiation

- Automatic Differentiation is a generalisation of derivatives on general programs.

Dual numbers can be expressed mathematically by including the abstract quantity ϵ, where ϵ^2 is defined to be 0. Like a complex number, a dual number is written $a + b\epsilon$ where a and b are both real values. We have:

$$(a + b\epsilon) + (c + d\epsilon) = (a + c) + (b + d)\epsilon \tag{2.32}$$

$$(a + b\epsilon) \times (c + d\epsilon) = (ac) + (ad + bc)\epsilon \tag{2.33}$$

Lets say we want to calc the first derivative at several points of this function:

$$f(x) = e^x / (\sin(x)^3 + \cos(x)^3)^{1/2}$$

Lets put that into a block:

```
f:=[:x|x exp / ((x sin raisedToInteger: 3) +(x cos raisedToInteger: 3))sqrt].
```

a dual number can consist of a value and its derivative. if we want to know the value of f' at x = 1.5 we construct a `PMDualNumber` this way, as the derivative of the identity function is 1:

```
anX:= PMDualNumber value: 1.5 eps:1.
f value:anX.
```

Hence f(1.5) is 4.497780053946163, lets check that:

```
f value:1.5."--> 4.497780053946163"
"and the value of f'(1.5) is 4.053427893898621"
```

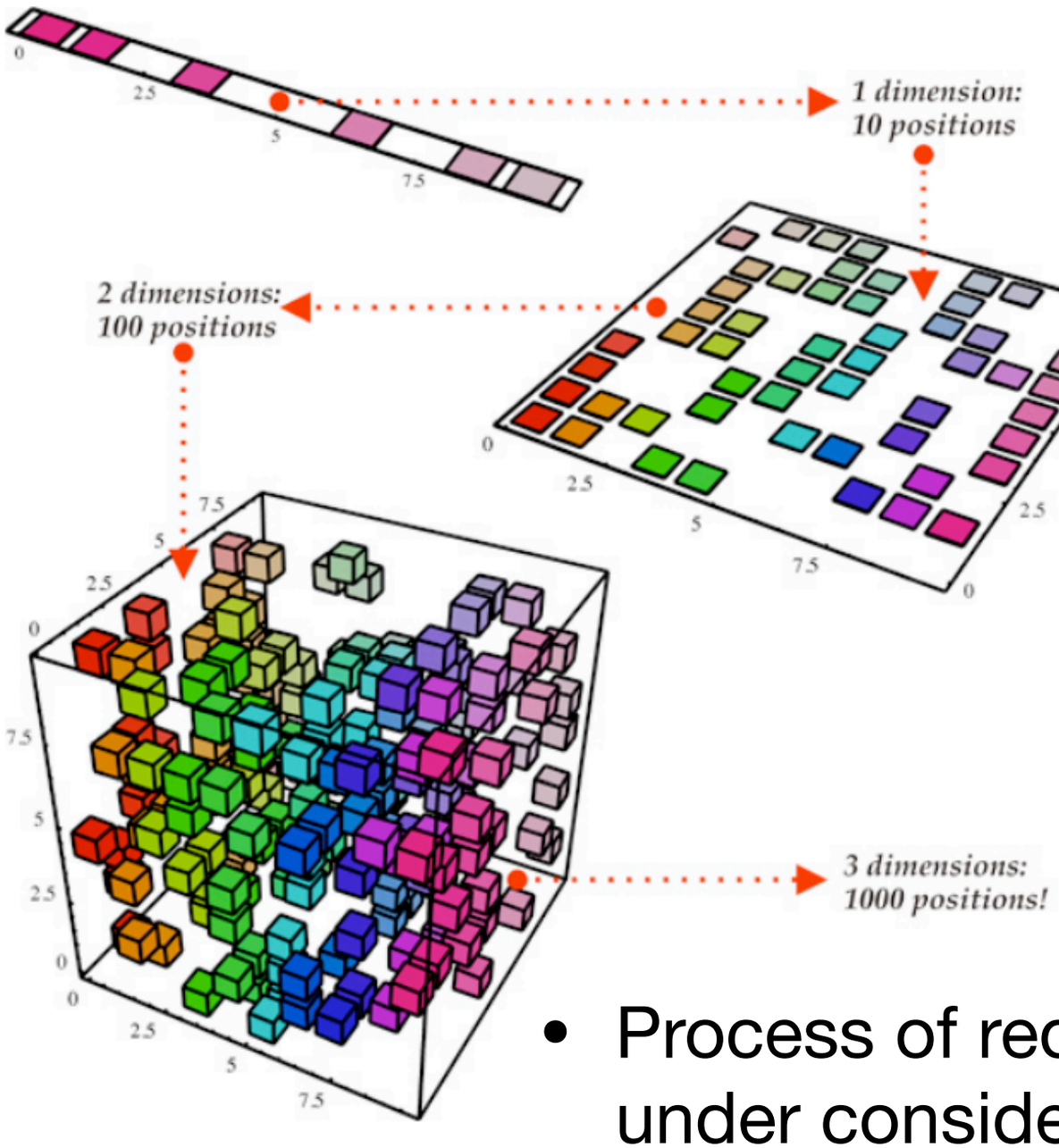
just to check this result, it is:

$$f'(x) = e^x(3 \cos(x) + 5 \cos(3 x) + 9 \sin(x) + \sin(3 x)) / (8 (\sin(x)^3 + \cos(x)^3)^{3/2})$$

```
1.5 exp*(1.5 cos *3 +((3*1.5)cos *5)+(1.5 sin *9)+(3*1.5)sin)/
 ((((1.5 sin raisedToInteger: 3) + (1.5 cos raisedToInteger: 3))
 raisedTo: (3/2))*8).
"--> 4.053427893898622"
```

Dimensionality reduction

What is dimensionality reduction ?

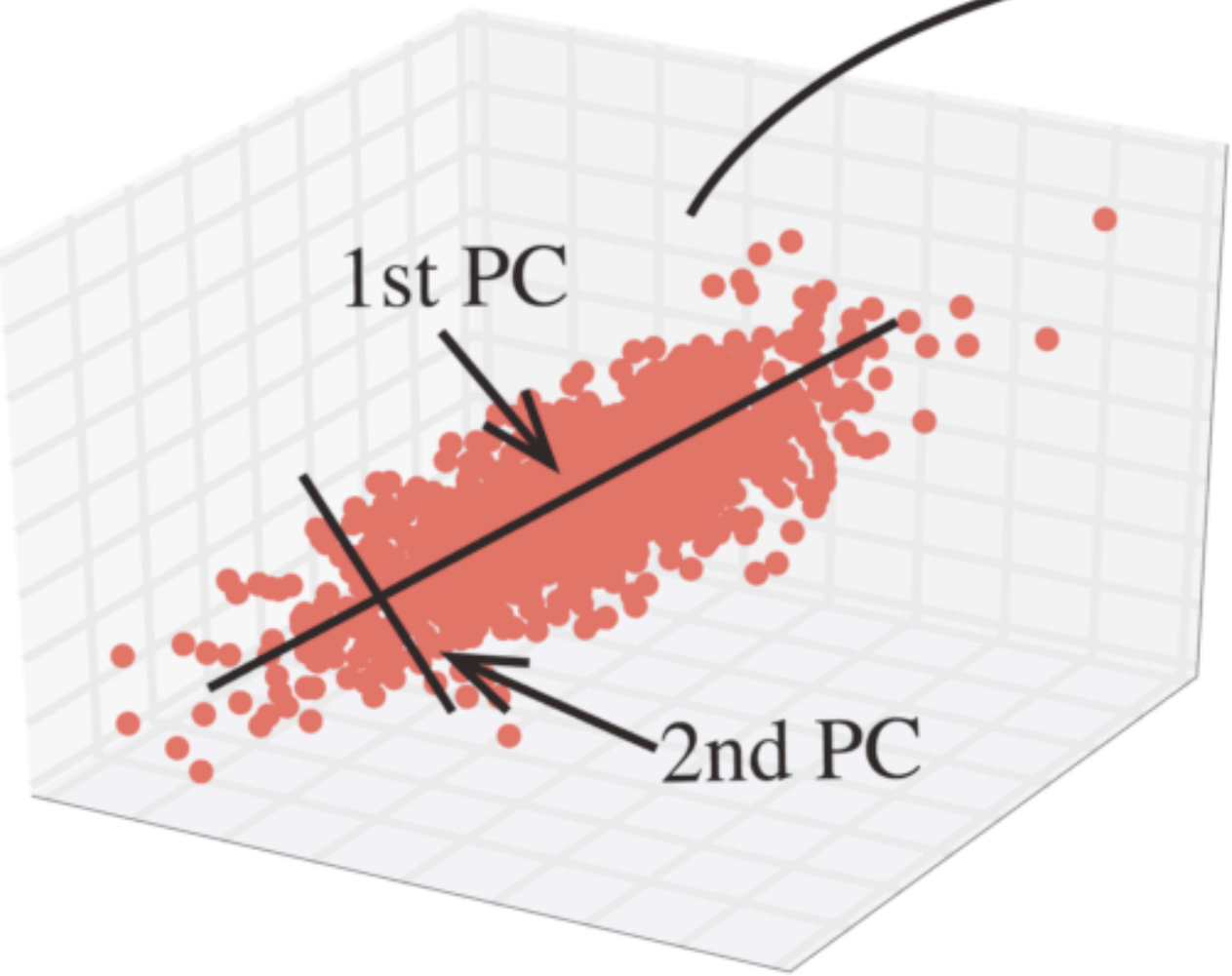


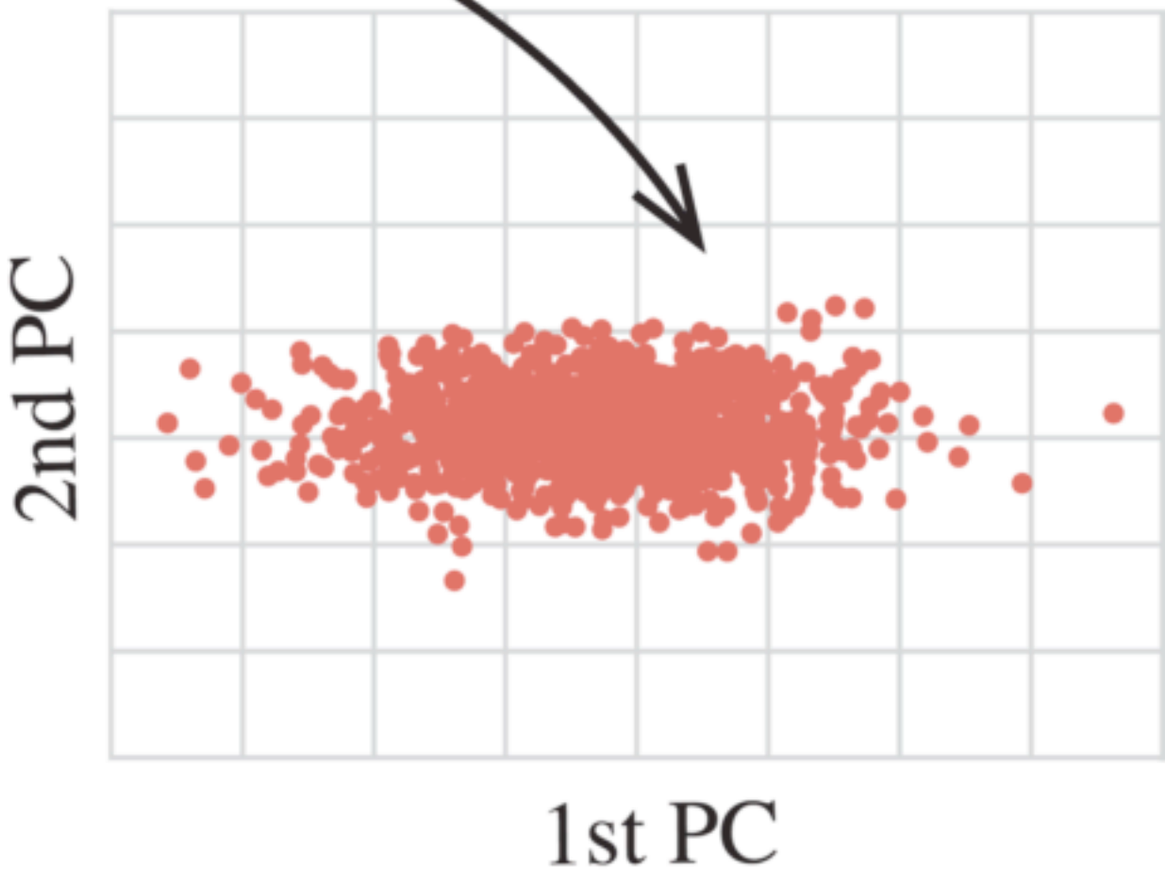
- Process of reducing the number of variables under consideration by obtaining a set of principal variables

Principal Component Analysis (PCA)

- Used in machine learning, statistics to solve the dimensionality curse.

- Dimension reductionality approach that perform a linear mapping of the data to a lower-dimensional space, in such a way that the variance of the data in low-dimensional representation is maximised.


1st PC
2nd PC


2nd PC
1st PC

t-SNE

- t-SNE short story: The goal is to take a set of points in a high-dimensional space and find a faithful representation of those points in a lower-dimensional space, typically the 2D plane.

- Probabilistic non-linear algorithm

- Demo of Atharva Kharve

Future of PolyMath

ActivePapers Pharo

- Work of Konrad Hinsen
- Pharo + GToolkit + ActivePapers + PolyMath = Scientific Workbench
- Presented at IWST 2019 (Tuesday)

ActivePapers Demo

Inspector

an InfluenzaLikeIllnessIncidenceInFrance class (InfluenzaLikeIllnessIncidenceInFrance)

Main page | Pages | Workflow | Scripts | Comment | Definition | Methods | InstVars | Exar

Incidence of influenza-like illness in France

This dataset on the incidence of influenza-like illness (French: syndrôme grippal) in France has been made available by the "Réseau Sentinelles" (`https://www.sentiweb.fr/`) and provides estimates extrapolated from the diagnoses made by general practitioners all over the country. The dataset contains cumulative weekly data, starting in October 1984. Unfortunately, the Web site does not provide the raw data (number of diagnoses), nor an explanation of the statistical methods used for generating the estimates.

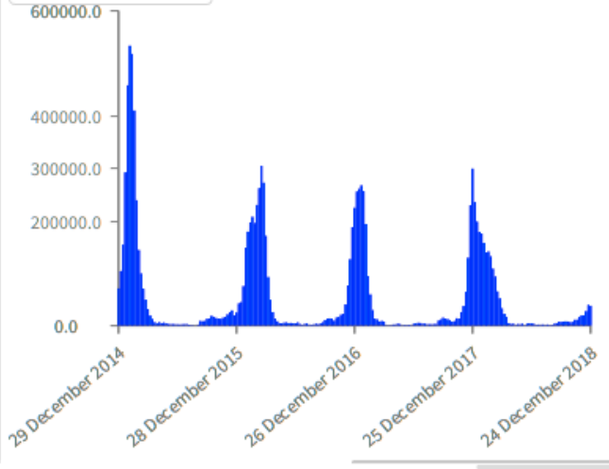
For each week, an incidence estimation is provided together with a 95% confidence interval. A population-relative incidence estimatation (cases per 100.000 inhabitants) is provided as well, again with a 95% confidence interval. The Web site does not say where the population data has been taken from.

See `DataProcessing` ▸ for an explanation of how the datasets in this document were obtained from the downloaded tables, and `Verification` ▸ for consistency and validity checks.

A plot of a three-year period shows the seasonal character of the incidence:

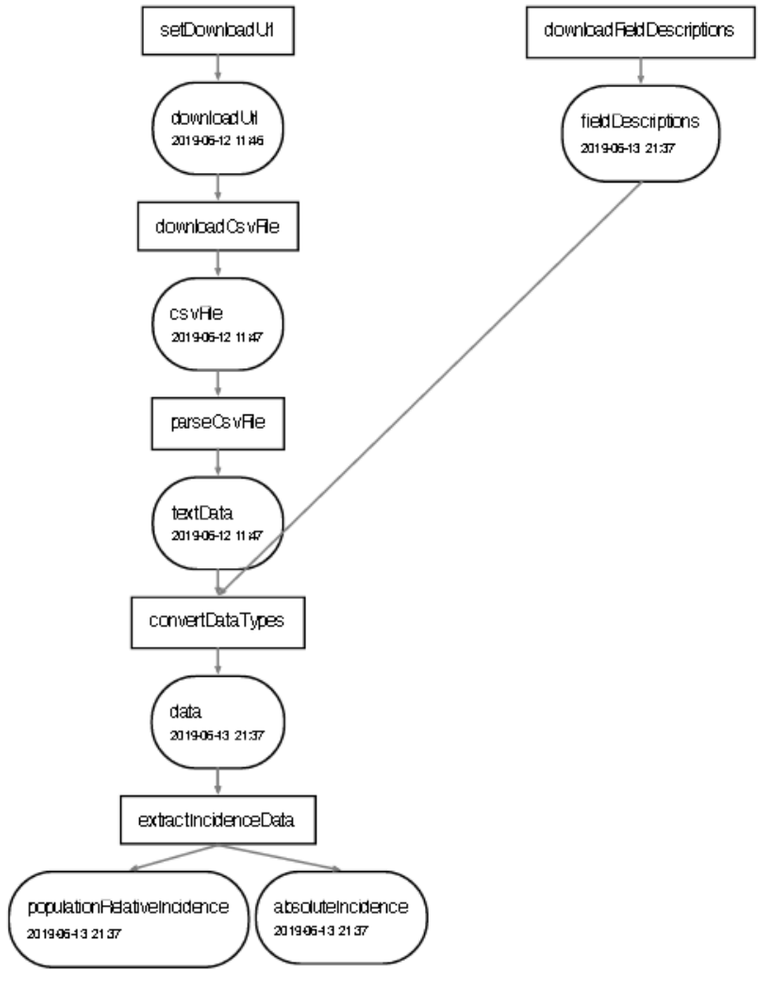
```
| threeYears |
threeYears := Timespan starting: (DateAndTime fromString: '2015-01-01') duration: 4 years.
APGraphics show:
   (self incidencePlotFor:
      (self selectTimespan: threeYears
            forIncidenceData: self absoluteIncidence)•)
```

Inspect in Morphic



an InfluenzaLikeIllnessIncidenceInFrance class (InfluenzaLikeIllnessIncidenceInFrance)

Main page | Pages | Workflow | Scripts | Comment | Definition | Methods | Inst



Port of Domains to Pharo

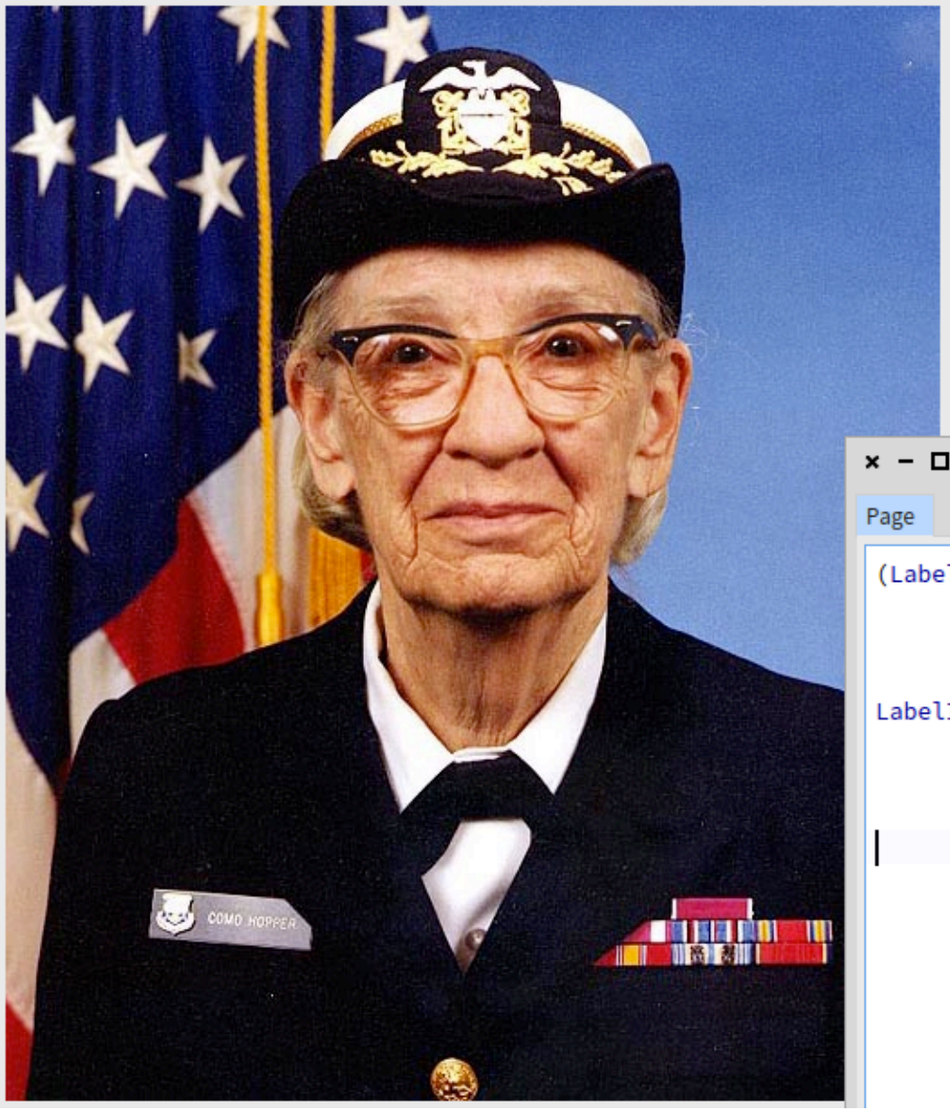
- Work of Luciano Notarfrancesco
- Computational algebra system in Smalltalk (Group, Morphism, Monoids, Elliptic curves, …)
- Ongoing port to Pharo : https://github.com/PolyMathOrg/Domains

Pharo

Playground

a FileReference (/tmp/grace_hopper.jpg)

Picture | Contents | Raw | Meta



LabelImage>>initialize

BaselineOfLibTensorFlowPha
LibTensorFlow-Core
LibTensorFlow-Examples

ExampleOLSPlan !
LabelImage
MNIST3LayersNNExamplePlan
MNIST3LayersNNSigmoid
MNISTFile !
MNISTImageFile !
MNISTLabelFile !
MNISTSoftMaxExamplePlan !
ManifestLibTensorFlowExampl

instance side
accessing
initialization
preparation
run
overrides

graphFile
graphFile:
imageFile
imageFile:
imageSize
imageSize:
initialize
inputMean
inputMean:
inputStddev

tensor

Filter...

All Packages | Scoped View | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementors | Senders

Comment | LabelImage | initialize | Inst. side method

```
initialize
    imageSize := 224.
    inputMean := 128.0.
    inputStddev := 127.0.
    graphFile := '/tmp/mobilenet_v1_1.0_224_quant_frozen.pb'.
    imageFile := '/tmp/grace_hopper.jpg'.
    labelsFile := '/tmp/labels.txt'
```

Playground

Page

```
(LabelImage new prepareImageInput;
        prepareSession;
        predict) first: 5.

LabelImage new prepareImageInput;
        prepareSession;
        predict;
         yourself
```

an Array [5 items] (#('653:military uniform' 0.811956524848938) #('668:mc...

Items | Raw | Meta

Index	Item
1	an Array [2 items] ('653:military uniform' 0.811956524848938)
2	an Array [2 items] ('668:mortarboard' 0.04821934178471565)
3	an Array [2 items] ('668:mortarboard' 0.04821934178471565)
4	an Array [2 items] ('820:stage' 0.012768302112817764)
5	an Array [2 items] ('820:stage' 0.012768302112817764)

Support PolyMath

- You can contribute code and documentation. Pull requests are welcome

- Please contribute fundings for PolyMath.

- BountySource: https://www.bountysource.com/teams/polymath

- Ask to be funded by NumFocus (Numpy, Matplotlib, pandas …)