

Julia: A Fresh Programming Language for Freshmen

大学新入生に Julia で演習する



Julia Tokyo # 8

2018 年 10 月 20 日

菅原 宏治

首都大学東京 システムデザイン学部

機械システム工学科 (2018 年発足)

Hiroharu Sugawara, Ph.D.

Faculty of Systems Design, Tokyo Metropolitan University

hsugawa@tmu.ac.jp

Julia: A Fresh Programming Language for Freshmen

大学新入生に Julia で演習する

- ▶ 経緯
 - ▶ 2018 年度 改組
 - ▶ 秋冬学期 1 限 90 分 × 15 週
 - ▶ 1 年次学生 定員 90 名
 - ▶ 学生にとって最初のプログラミング演習の授業
 - ▶ 講演者にとっても、初めての担当
- ▶ 授業設計・目論見
- ▶ 準備
- ▶ 教材案紹介
- ▶ Tips

授業設計・目論見

◎ 目標

- ▶ 今後の専門学習に役立つ
- ▶ 自ら学習する姿勢を涵養する
(Active Learning)
- ▶ 共通の技法 (Science,
Technology, Engineering,
Art and Mathematics)

◎ 現実：大学1年次

- ▶ PC 未習熟
- ▶ 数学：未習熟 ... 行列

◎ 使用する言語への要求

- ▶ 数学記法との親和性
- ▶ やさしく正しい言語仕様
- ▶ ただちに結果が得られる
- ▶ 導入が容易
- ▶ 無料



× 日本語テキスト

大学院担当授業で試行

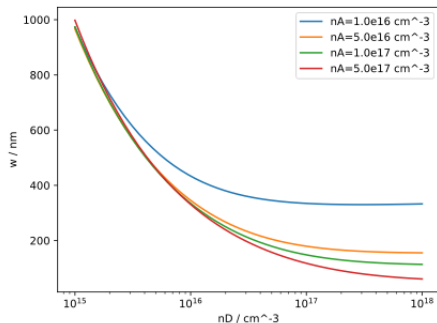
- ▶ 2017 年度開講
数式を全てグラフ化することを目指す
- ▶ ノート PC を教室に持参させた
- ▶ 以下をインストール
 - ▶ Julia v 0.6.4
 - ▶ IJulia パッケージ Jupyter Notebook
 - ▶ PyPlot パッケージ: matplotlib
 - ▶ Unitful
- ▶ Julia 言語のテキストは準備しない
各回授業で、随時説明

大学院授業・グラフ描画例

パラメータ N_D, N_A に対して

$$V_b = \frac{k_B T}{e} \ln \frac{N_A N_D}{n_i^2}$$
$$w = \sqrt{\frac{2\epsilon V_b}{e} \frac{N_D + N_A}{N_D N_A}}$$

k_B, T, n_i, ϵ, e : 定数



```
ee=1.602e-19 # C
kbTe=1.38e-23 * 300 / ee # V
ni=1e10 # cm^-3
vb(nap, ndp) = kbTe * log( nap .*ndp / ni^2) # V
eps=8.854e-12*11.9 * 1e-2 # F/cm
ww(nap, ndp) = sqrt(2*eps/ee .*
    vb.(nap, ndp) .* (nap .+ ndp) / (nap .* ndp )) # cm
using PyPlot
nD=logspace(15,18)
for nA in [1e16, 5e16, 1e17, 5e17] # cm^-3
    plot(nD, ww.(nD, nA) / 1e-7, label="nA="*string(nA)*" cm^-3" )
end
legend()
xscale("log")
xlabel("nD / cm^-3")
ylabel("w / nm")
```

準備・作業日程

- ▶ 授業計画書 (シラバス) : 前年 11 月頃
 - ▶ 「手続き型プログラミングの基本演習」
 - ▶ 使用言語は明記せず
- ▶ 教室 PC 環境作成依頼と検証 : 本年 7~8 月
 - ▶ Julia 0.6.4
 - ▶ Julia パッケージ : PyPlot, IJulia, DataFrames, CSV,
 - ▶ PC 環境復元ツール : ファイル元イメージの作成
 - ▶ Windows10 起動ドライブに「c:\Julia」を作成
 - ▶ 公式バイナリを利用。Julia コマンドラインから Pkg.add
 - ※ anaconda 寸法大きい。4GB 弱。各ユーザ領域には設置不可
- ▶ Github Classroom 申請 : 本年 7 月
- ▶ 演習テキスト : 直前 (現在進行中)
 - ▶ 普段からネタを考える。Jupyter notebook
 - ▶ Documenter.jl の利用
- ▶ 10 月からスタート (3 回実施済み)

教材案：導入

第1回目

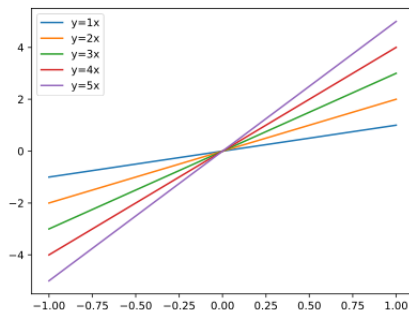
数の四則演算, 変数,
Range 型, plot 関数

```
using PyPlot  
xs=-1:0.1:1  
plot(xs, xs)
```

第2回目

文字列, グラフ凡例, for 文

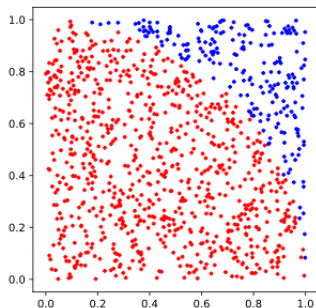
```
using PyPlot  
xs=-1:0.1:1  
for a in 1:5  
    plot(xs, a*xs, label="y="+string(a)+"x")  
end  
legend()
```



教材案 (中盤): モンテカルロ法による図形面積の推定

ランダムに落とした点の総数のうち、
図形の中に入った点の割合で、面積を推定
乱数, if 文 (for ブロック内)

```
using PyPlot
m=10
n=2^m
s=0
for i=1:n
    x=rand()
    y=rand()
    if x*x + y*y <= 1 # 四分円の中
        c="r"
        s += 1
    else
        c="b"
    end
    plot(x,y,".", color=c)
end
@show s/n
@show pi/4
plt[:axes](:)[:set_aspect]("equal")
```



$$s / n = 0.791015625$$

$$\pi / 4 = 0.7853981633974483$$

教材案 (中盤): 数値微分

浮動小数点数 桁落ち・情報落ち

定義に基づく数値微分

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

$f(x) = x^n$ ($n = 1, 2, 3$)

$x = 1$ における微係数。

正解は $f'(x) = n$

using PyPlot

`h=logspace(-18,-8,100)`

for `n=1:3`

`d=(1+h).^n - 1) ./ h`

`plot(h,d, ".", label="y=x^"*string(n))`

end

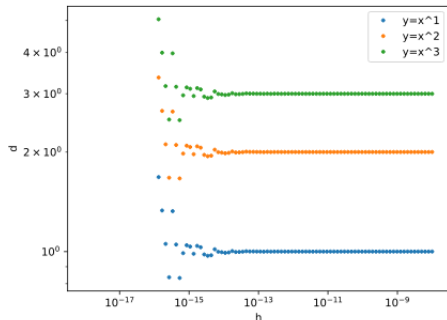
`xlabel("h")`

`ylabel("d")`

`yscale("log")`

`xscale("log")`

`legend()`



教材案 (中盤): 行列を自在に使う

行列の初期化、乗算

```
julia> # 列ベクトルを並べて行列にまとめる
xy=hcat([ [2*cos.(t); sin(t)] for t=0:pi/18:2pi]...)
```

```
2x37 Array{Float64,2}:
```

```
 2.0  1.96962  1.87939  1.73205 ...  1.87939  1.96962  2.0
 0.0  0.173648 0.34202  0.5         -0.34202 -0.173648 -2.44929e-16
```

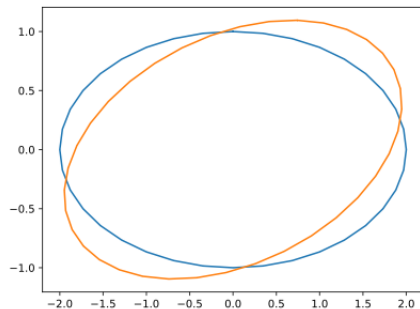
```
julia> using PyPlot
plot(xy[1,:), xy[2,:])
```

```
julia> r15=[ cosd(15) -sind(15);
             sind(15) cosd(15)]
```

```
2x2 Array{Float64,2}:
```

```
0.965926 -0.258819
0.258819 0.965926
```

```
julia> # 回転行列を左から乗ずる
xy = r15 * xy
plot(xy[1,:), xy[2,:])
```



教材案 (中盤): 線形結合・線形写像

ベクトルの初期化、写像

```
julia> u1=[1/2, 1/2]; u2=[1/2, -1/2];
```

```
julia> # 面心平面格子
```

```
    xy1=hcat( [ i * u1 + j*u2
               for i in 0:3, j in 0:3 ]... )
2x16 Array{Float64,2}:
 0.0  0.5  1.0  1.5  0.5  1.0  1.5  2.0  ...  2.0  2.5
 0.0  0.5  1.0  1.5 -0.5  0.0  0.5  1.0  ...  1.0  1.5
```

```
julia> r45=[cosd(45) -sind(45);
           sind(45) cosd(45)]
```

```
julia> # 回転と拡大 →整数の格子点に重なる
```

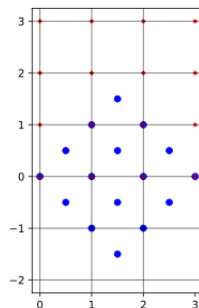
```
    xy2=sqrt(2)*r45*xy1
2x16 Array{Float64,2}:
 0.0  0.0  0.0  1.11022e-16  1.0  1.0
 0.0  1.0  2.0  3.0          0.0  1.0
```

```
julia> using PyPlot
```

```
julia> plot( xy1[1,:], xy1[2,:], "bo")
```

```
julia> plot( xy2[1,:], xy2[2,:], "r.")
```

```
julia> plt[:axes]()[:set_aspect]("equal")
```



評価

- ▶ 説明よりも、実行を優先する方式の演習
- 概ね好評
 - ※ グラフが描ける、すぐ修正して試せる
- △ 不評な受講生も...
 - 一字一句理解してから進めたい
 - 英語文献ばかり。日本語の説明が少ない。
- ▶ Julia は繊細・ガラス細工
 - Predefined がたくさん。警告なく書換え可能。
- ▶ テキスト案。以下で公開。不定期に更新中
 - <https://hsugawa8651.github.io/memojuliav064/>

教室 PC 運用上の Tips

- ▶ anaconda ライブラリは大きい (再掲)
インストール・ディレクトリをシステム側に設置
Windows は `c:\Julia\0.64\`, `c:\Julia\Pkg` を準備
MacOSX は 調査中
- ▶ Jupyter notebook
token 問題
- ▶ github (classroom)
アクセスをずらす。DoS attack にならないように、

解題・まとめに代えて

- ▶ Julia: A Fresh Approach to Numerical Computing

<https://julialang.org/publications/julia-fresh-approach-BEKS.pdf>

We introduce the Julia programming language and its design
— a dance between specialization and abstraction.

- ▶ Julia: A Fresh Programming Language for Freshmen

(New wine in new vessels)

Comprehensive and Powerful

プログラミング導入にもぜひ

... 日本語の文献

