



alternatywa Matlabu i Fortranu

Wojciech Dębski

IGF PAN, 7.03.2016

Obliczenia numeryczne - języki programowania

- ◆ proste arkusze kalkulacyjne a'la Excell (Basic)
- ◆ języki uniwersalne: Fortran, C, C++, Algol, Lisp
- ◆ języki wysokiego poziomu:

Matlab, Mathematica, R, Python, Julia

- ◆ języki skryptowe: Bash, Csh, PHP
- ◆ obliczenia równoległe (HPC)

Obliczenia numeryczne: FORTRAN 1950 ...



Programowanie statyczne i dynamiczne

♦ języki programowania statycznego (Fortran, C)

- ★ stworzenie kodu źródłowego
- ★ kompilacja
- ★ debugowanie

.....

♦ języki dynamiczne (Matlab, R, Python, Julia)

- ★ stworzenie kodu źródłowego - interpretacja + debugowanie
- ★ wyniki (jeśli program pracuje)

Jakiego języka używać

- ◆ Zagadnienia bardzo proste (np. dopasowanie prostej metodą LSQR)
Excell, R, Matlab, Julia, Mathematica
- ◆ zagadnienia bardziej złożone (np. optymalizacja i algebra liniowa)
Matlab, Julia, R, Mathematica, Excell
- ◆ zagadnienia nieliniowe, równania różniczkowe zwyczajne
Matlab, R, Julia, Mathematica
- ◆ szeregi czasowe, statystyka
R, Matlab, Julia
- ◆ przetwarzanie danych obserwacyjnych, “prosta” inwersja
Julia, Matlab oraz C, Fortran + języki skryptowe
- ◆ intensywne obliczenia, tomografia, modelowanie
Fortran, C, + zrównoleglenie (MPI, OpenMP), **Julia!!!**

Jakiego języka używać - porównanie wydajności

Test	Fortran	Julia	Python	R	Matlab	Mathematica
fib	0.70	2.11	77.76	533.52	26.89	118.53
parse_int	5.05	1.45	17.02	45.73	802.52	15.02
quicksort	1.31	1.15	32.89	264.54	4.92	43.23
mandel	0.81	0.79	15.32	53.16	7.58	5.13
pi_sum	1.00	1.00	21.99	9.56	1.00	1.69
rand_mat_stat	1.45	1.66	17.93	14.56	14.52	5.95
rand_mat_mul	3.48	1.02	1.14	1.57	1.12	1.30

Julia - typy zmiennych

- ◆ liczby całkowite (różne formaty)
- ◆ liczby zmiennie przecinkowe (różne formaty)
- ◆ liczby zespolone ($x + y im$)
- ◆ liczby ułamkowe ($x // y$)
- ◆ tekstowe ("text") + UTF-8
- ◆ *Arbitrary Precision Arithmetic*

Julia - statystyka

- ◆ podstawowe estymatory (mean, var, std, ,...)
- ◆ histogramy (w tym 2D
- ◆ kwantyle
- ◆ kowariancja, korelacje
- ◆ testy ???

Julia - przetwarzanie sygnałów

- ◆ FFT, IFFT (w tym wielowymiarowe)
- ◆ discrete cosine transformation
- ◆ convolution, correlation (w tym 2D)
- ◆ transformacje Hartley +
- ◆ filtry (pakiety zewnętrzne)

Przykład

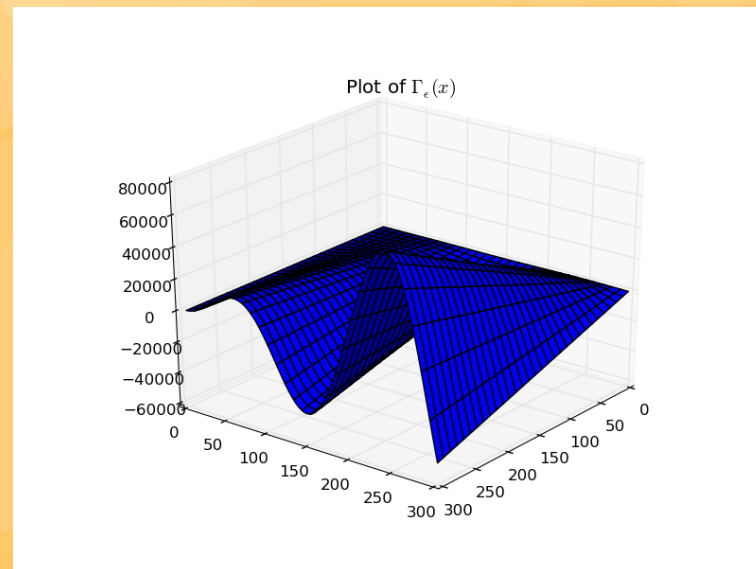
```
function randmatstat(t)
    n = 5
    v = zeros(t)
    w = zeros(t)
    for i = 1:t
        a = randn(n,n)
        b = randn(n,n)
        c = randn(n,n)
        d = randn(n,n)
        P = [a b c d]
        Q = [a b; c d]
        v[i] = trace((P.'*P)^4)
        w[i] = trace((Q.'*Q)^4)
    end
    std(v)/mean(v), std(w)/mean(w)
end
```

Przykład - zrównoleglenie

```
function randmatstat(t)
    n = 5
    v = zeros(t)
    w = zeros(t)
    @parallele for i = 1:t
        a = randn(n,n)
        b = randn(n,n)
        c = randn(n,n)
        d = randn(n,n)
        P = [a b c d]
        Q = [a b; c d]
        v[i] = trace((P.'*P)^4)
        w[i] = trace((Q.'*Q)^4)
    end
    std(v)/mean(v), std(w)/mean(w)
end
```

Przykład - grafika (PyPlot)

```
using PyPlot
x=[1:300];
y= x*x';
z=y.*sin(y[1,:]/30);
surf(z)
title(L"Plot of  $\Gamma_\epsilon(x)$ ")
```



Pakiety (<http://pkg.julialang.org/>)

- Anasol - Analytical solutions for groundwater contaminant transport
- ApproxFun - Julia package for function approximation
- Regression - Algorithms for regression analysis (e.g. linear regression and logistic regression)
- PyCall - Package to call Python functions from the Julia language
- LinearLSR - Least squares solver in Julia
- Lora - MCMC inference in Julia
- MAT - Julia module for reading MATLAB files
- TimeMdels - Handling time series data in Julia
- V-statistic - Implements the v-statistic, a measure that compares the estimation accuracy of the ordinary least squares estimator against a random benchmark.

Czy Julia zastąpi Matlaba, R, Pythona....

Prawdopodobnie nie, ale wraz z tworzeniem nowych pakietów, dopracowaniu grafiki, będzie na pewno stanowiła duże wyzwanie dla bardzo drogich (Matlab) lub wolnych (R, Python) języków programowania.

Wydaje się, że może być alternatywą dla Fortranu/C w obliczeniach HPC

Dużym atutem Julii jest możliwość bezpośredniego korzystania z pakietów Pythona, R, Matlaba i bibliotek Fortranu/C

**Reasumując Julia ma dużą szansę na wprowadzenie
“HPC pod strzechy”**