The Julia Language

With a performance comparison in Kalman filtering

Simo Särkkä

Aalto University

November 10, 2015

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

$\Rightarrow$ *Demo*

What is Julia?

- Open-source programming language for technical computing.
- Very Matlab-like syntax.
- Interpreted language with a builtin JIT-compiler.
- Fast program execution – closer to C than Matlab (they say).
- Modern programming concepts found in Python and Lisp, but not in Matlab.
- Multiple dispatching (kind of polymorphism).
- Operator overloading (operators and functions).

  $\Rightarrow$ *Demo*

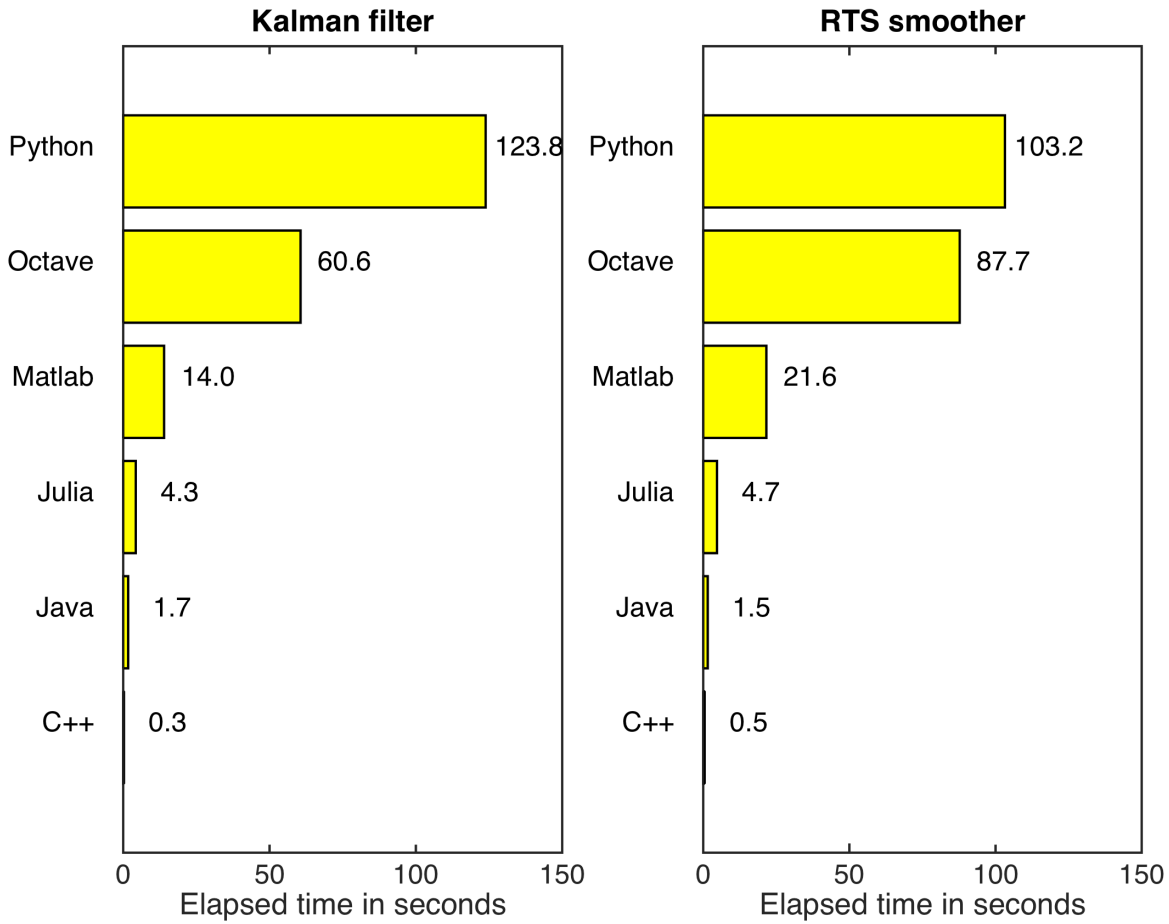
Performance I

Performance comparison from `http://julialang.org/`:

	Fortran gcc 5.1.1	Julia 0.4.0	Python 3.4.3	R 3.2.2	Matlab R2015b	Octave 4.0.0	Mathematica 10.2.0	JavaScript V8 3.28.71.19	Go go1.5	LuaJIT gsl-shell 2.3.1	Java 1.8.0_45
fib	0.70	2.11	77.76	533.52	26.89	9324.35	118.53	3.36	1.86	1.71	1.21
parse_int	5.05	1.45	17.02	45.73	802.52	9581.44	15.02	6.06	1.20	5.77	3.35
quicksort	1.31	1.15	32.89	264.54	4.92	1866.01	43.23	2.70	1.29	2.03	2.60
mandel	0.81	0.79	15.32	53.16	7.58	451.81	5.13	0.66	1.11	0.67	1.35
pi_sum	1.00	1.00	21.99	9.56	1.00	299.31	1.69	1.01	1.00	1.00	1.00
rand_mat_stat	1.45	1.66	17.93	14.56	14.52	30.93	5.95	2.30	2.96	3.27	3.92
rand_mat_mul	3.48	1.02	1.14	1.57	1.12	1.12	1.30	15.07	1.42	1.16	2.36

Figure: benchmark times relative to C (smaller is better, C performance = 1.0).

Performance II



Relative computation time (filter + smoother):

C++	Java	Julia	Matlab	Octave	Python
1	4	12	48	199	304

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

  $\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

$\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

  $\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

$\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

$\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

$\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

$\Rightarrow$ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.

  ⇒ *Should I switch to it?* – Not for now.

Quick practical experiences

- ✓ Transliteration from Matlab is very easy (with performance catches).
- ✓ Faster than Matlab, but loses to C++ and Java.
- ✓ Much faster than Python or Octave (no surprise).
- ✓ Some good programming features (proper lists, functional programming, multiple dispaching, . . .).
- ✓ No objects or classes!
- ✓ Programming environment quite clumsy to get working.
- ✓ Plotting features are just awful.
- ✓ No proper IDEs.
  - ⇒ *Should I switch to it?* – Not for now.