



Technical Computing made
Fast, Flexible, and Easy

"The Science is in the Software"

SIAM Conference on
Computational Science and Engineering

An abstract graphic consisting of numerous thin, curved, overlapping lines in various colors including blue, green, yellow, orange, and purple, creating a sense of motion and complexity.

February 27-March 3, 2017
Hilton Atlanta, Atlanta, Georgia, USA

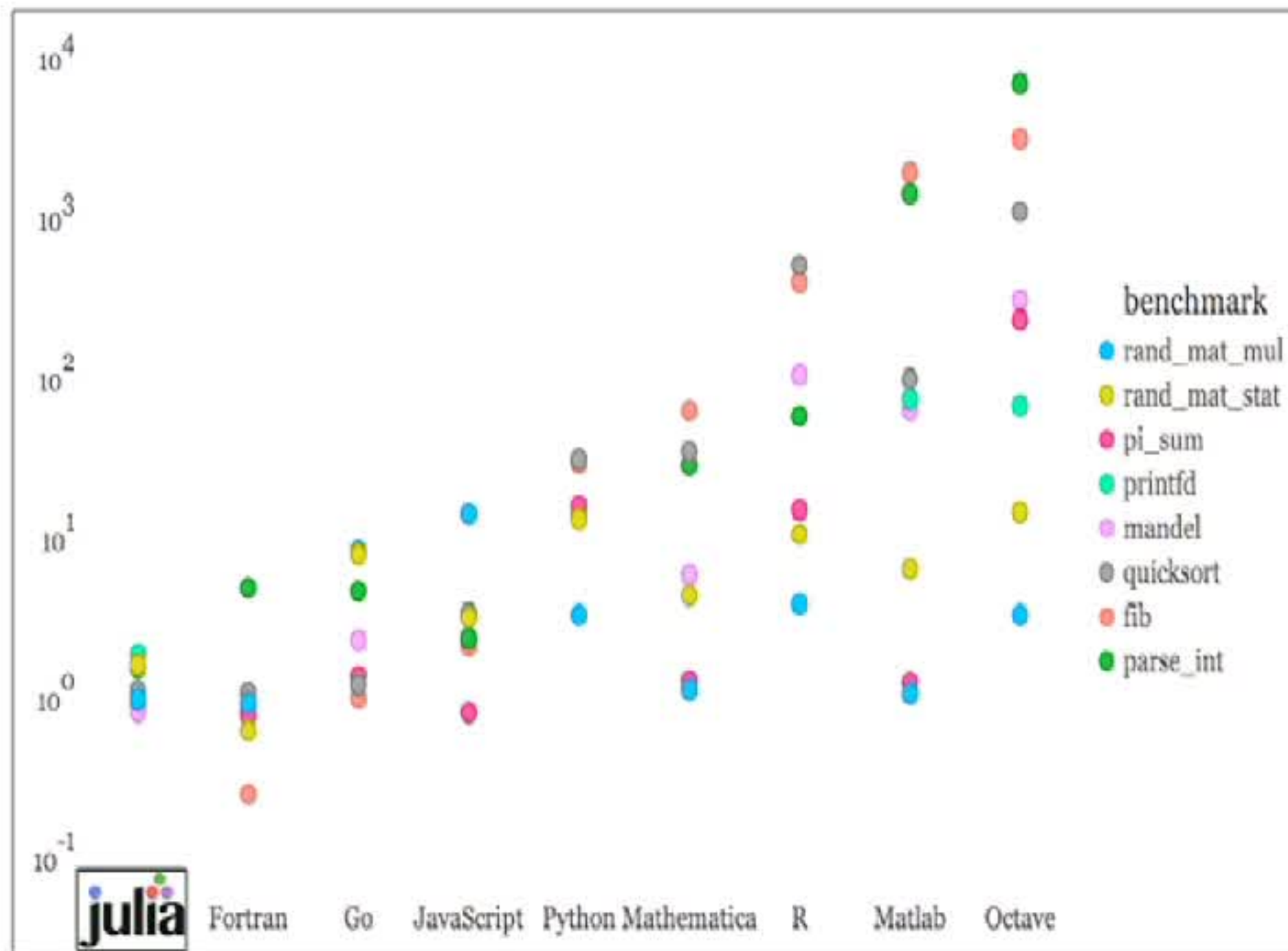
MIT
MATHEMATICS



 **Julia**
computing

Alan Edelman
David Sanders
Andreas Noack
Jarrett Revels

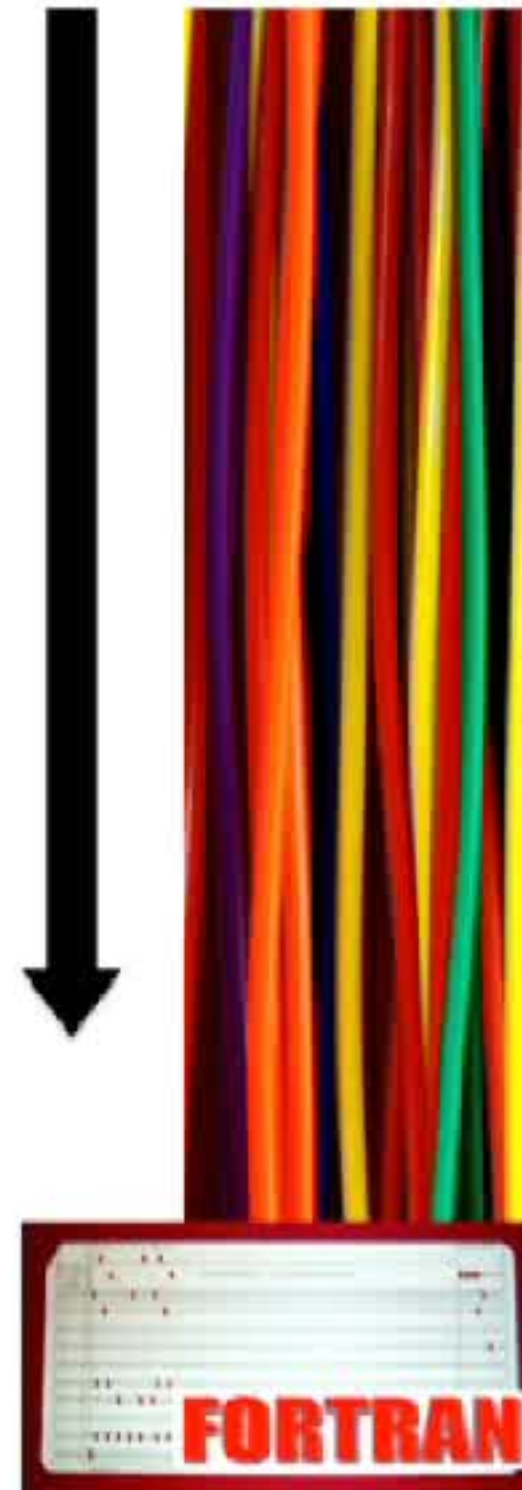
Designed for speed and ease of use



Benchmark times relative to C (smaller is better, C performance = 1.0).



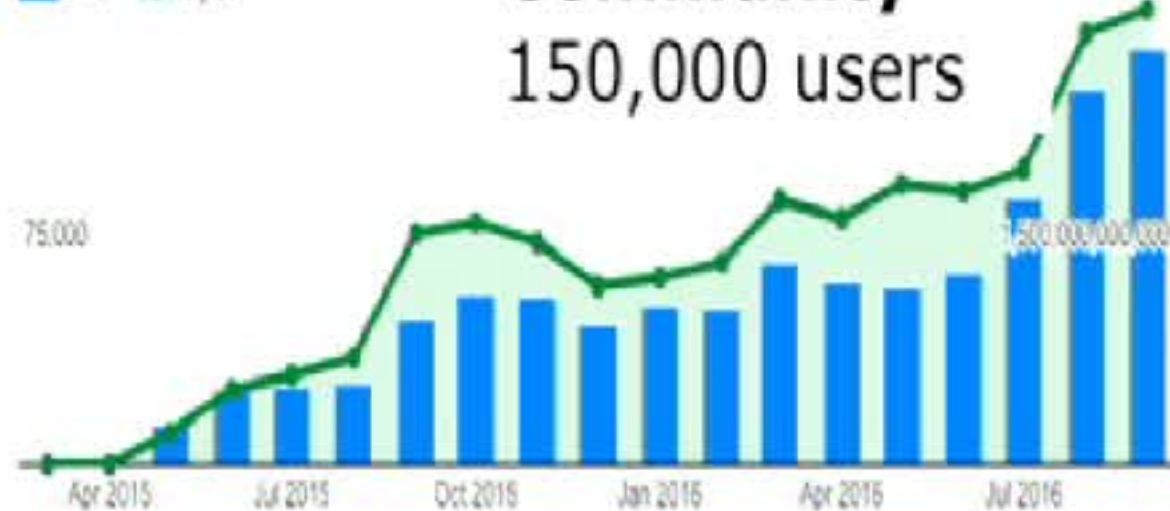
Other Dynamic Languages:



The user base is doubling every 9 months

Monthly Traffic

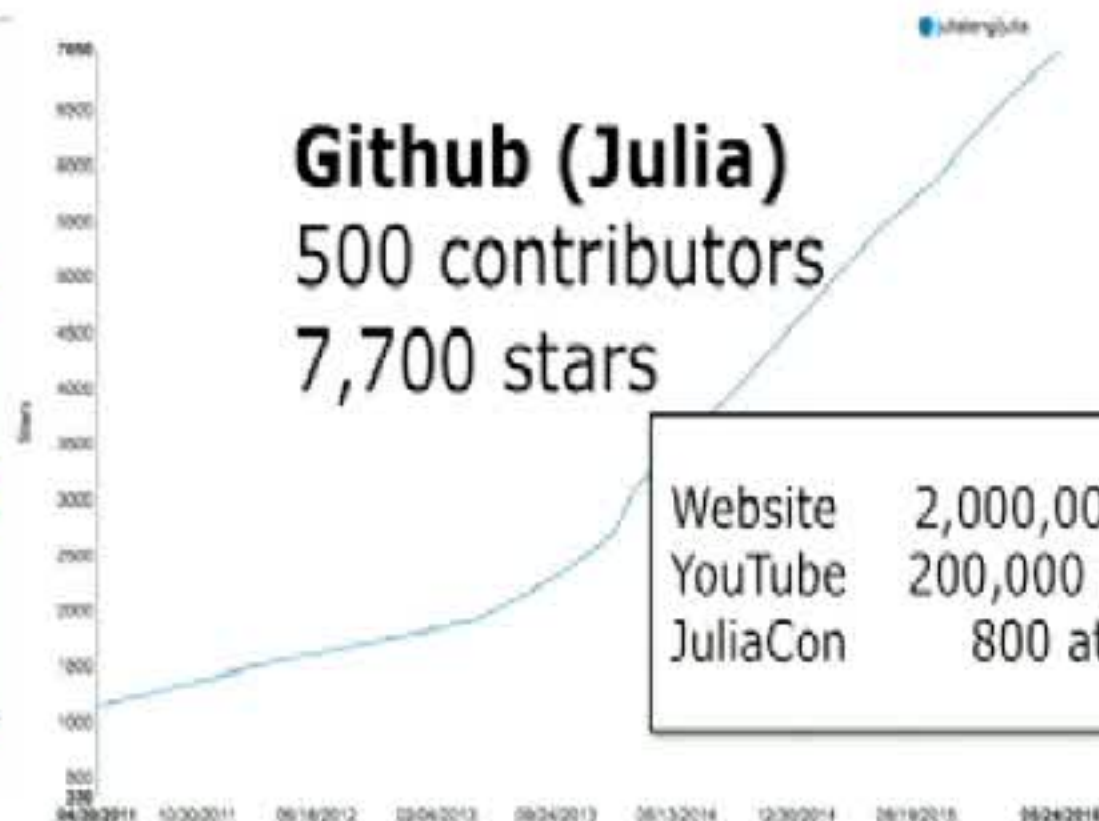
Hits Bytes



Github (Julia)

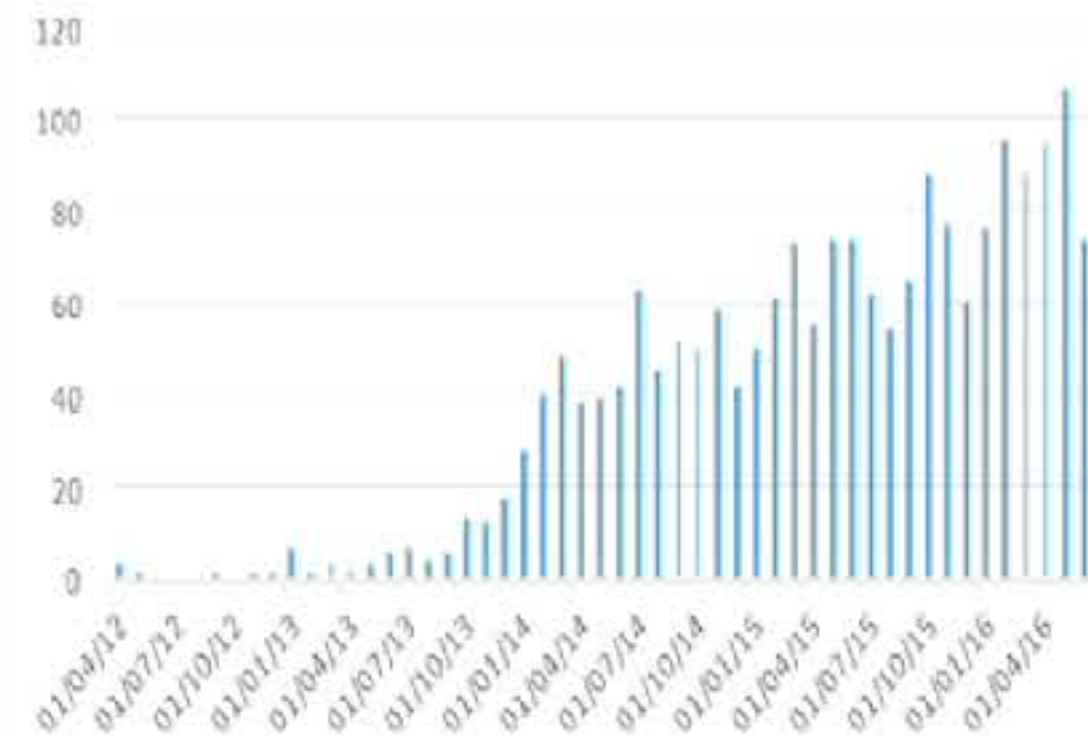
500 contributors

7,700 stars



Website	2,000,000 visitors
YouTube	200,000 views
JuliaCon	800 attendees

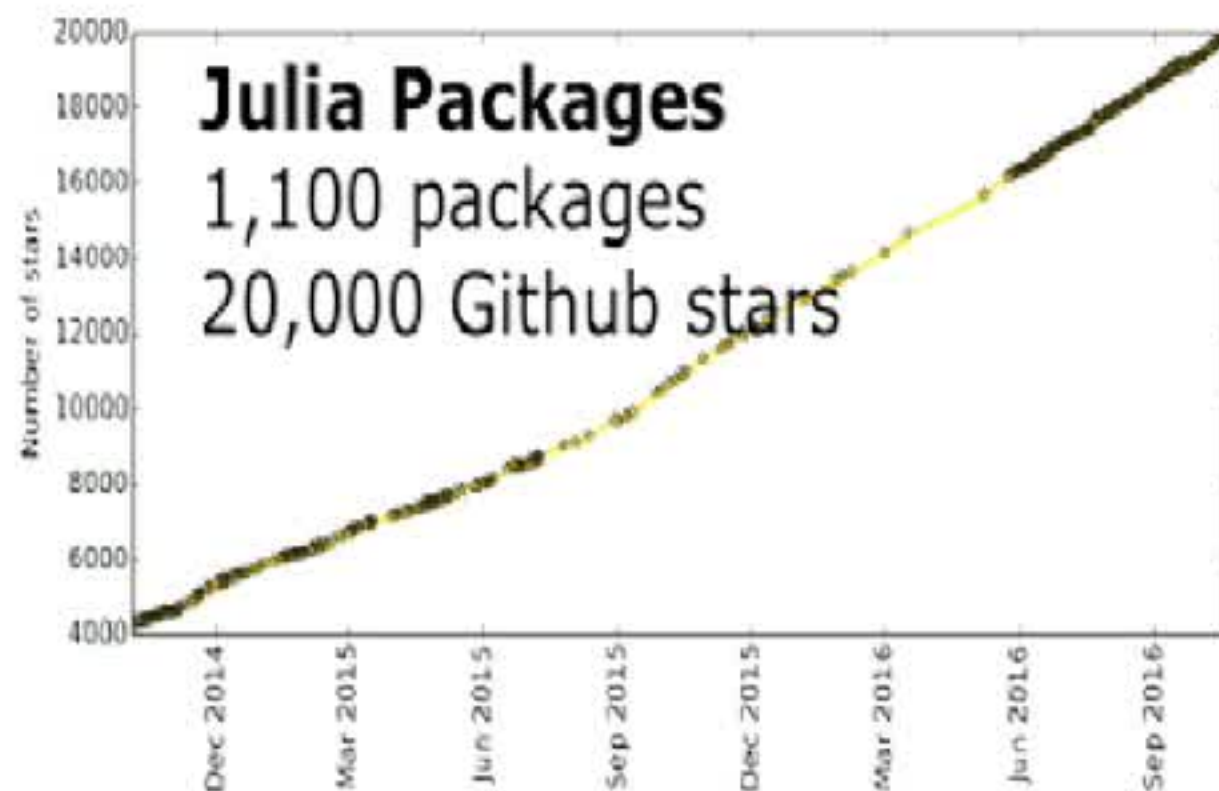
Questions on StackOverflow



Julia Packages

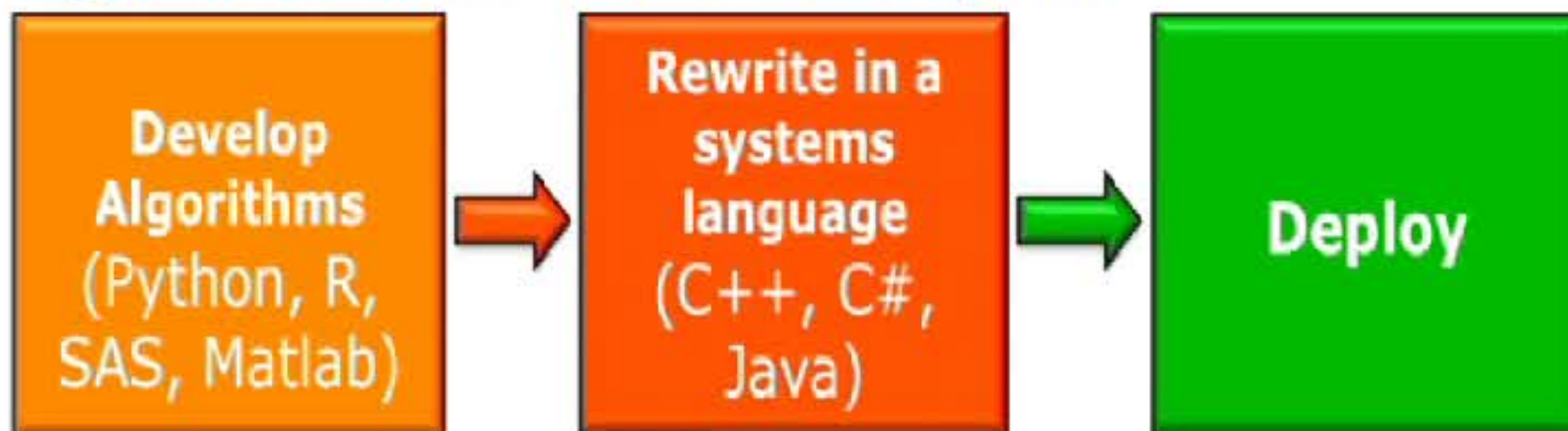
1,100 packages

20,000 Github stars

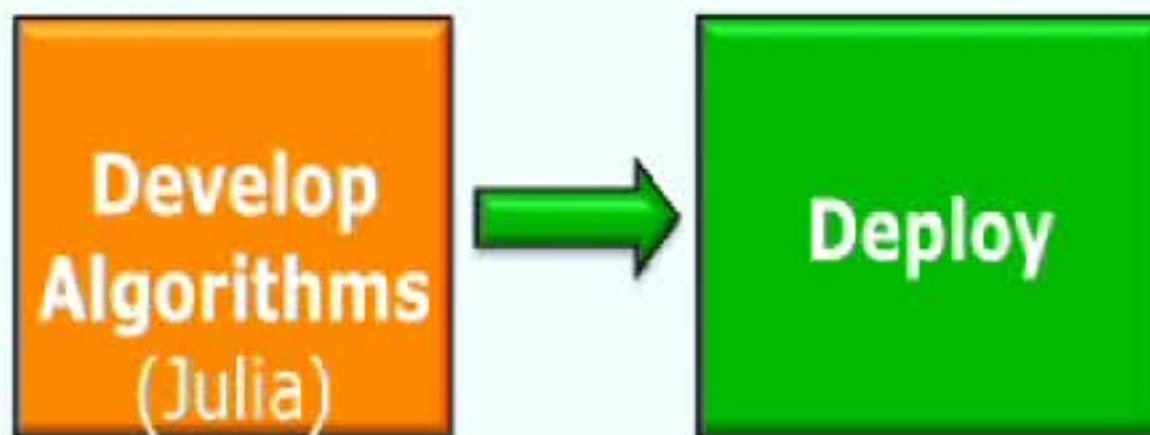


Those who convert ideas to products fastest will win

Typical Workflow in the last 25 years



Compress the innovation cycle with Julia



Celeste: Julia on 8,192 CPUs

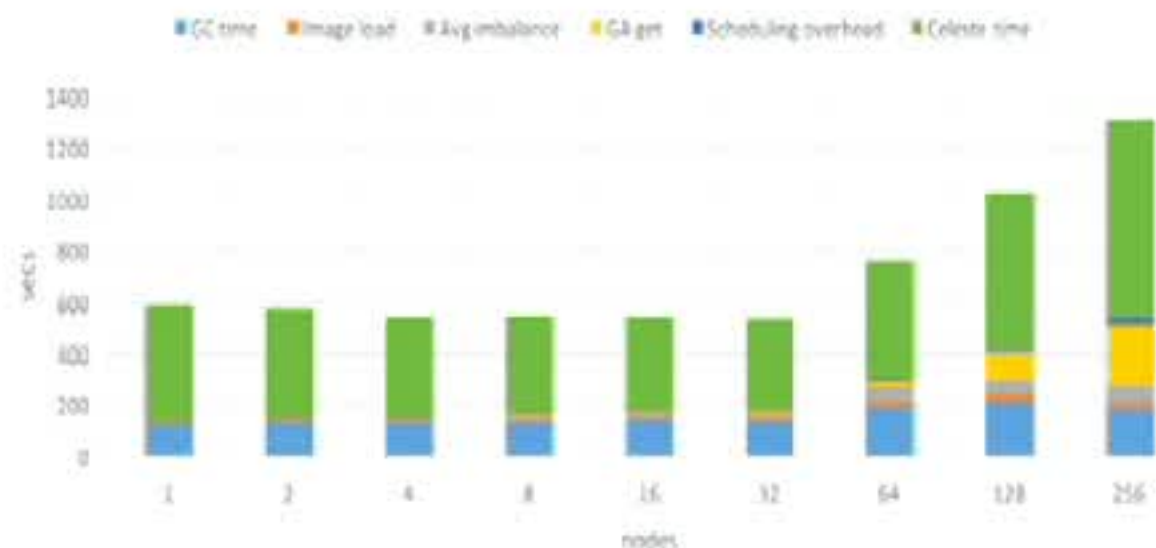


Figure 4: Weak-scaling Celeste. While the total runtime shown is precise, the components of runtime shown are averaged across nodes and thus should be treated as being representative rather than exact.

- Native Julia implementation of variational inference on a complex bayesian model
- Learning an astronomical catalog of the visible universe through scalable Bayesian inference
- Collaboration between LBNL, Julia Computing, Intel and JuliaLabs@MIT
- Huge scaling on the NECSRC Cori Supercomputer on 8192 Intel Xeon® cores
- Uses Julia's multi-threading and multi-processing capabilities

9:23 AM

Format Slide Show

Slide Show

Conference on
Natural Science
and Engineering

27-29 March 2017
Atlanta, Georgia, USA

Alan Edelman
David Sanders
Andreas Noack
Jarrett Revels

Screen Shot
06/04/2016 10:08:42 AM

David Foster

File Edit View Insert Cell Kernel Widgets Help

n 5

Out[67]: 5x5 Array{Int64,2}:

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

In [19]: #double slider!! Put a 3x3 block of i's in an n x n matrix of z

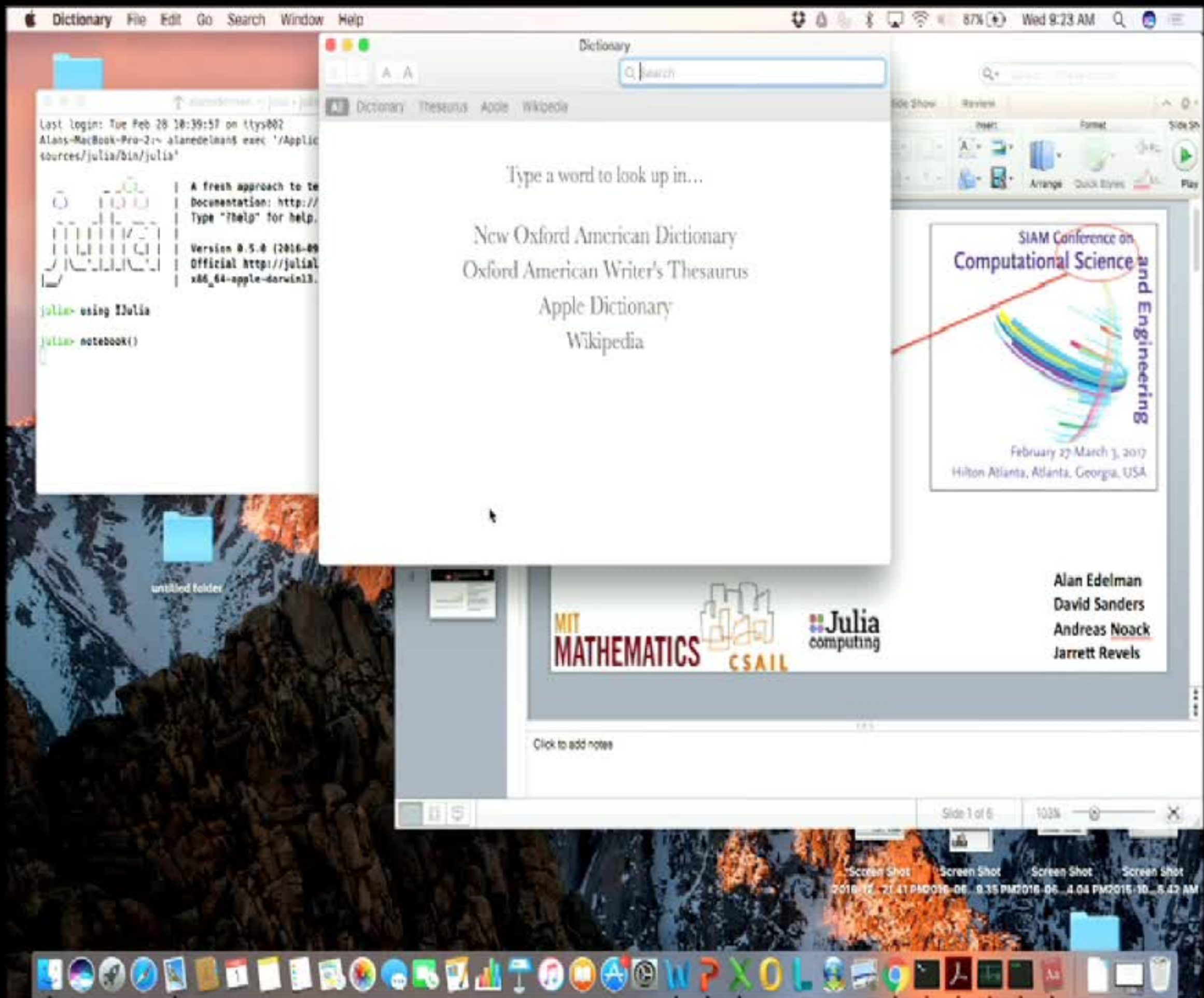
```
@manipulate for n=3:10, i=1:9
    A = fill(0,n,n)
    A[1:3, 1:3] = i
    A
end
```

n 6

i 5

Out[19]: 6x6 Array{Int64,2}:

5	5	5	0	0	0
5	5	5	0	0	0
5	5	5	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0



Dictionary

Search

Dictionary Thesaurus Apple Wikipedia

Type a word to look up in...

New Oxford American Dictionary
Oxford American Writer's Thesaurus
Apple Dictionary
Wikipedia

Last login: Tue Feb 28 18:39:57 on ttys002
Alan@MacBook-Pro-2:~ alan@edelman\$ exec '/Applications/Julia-0.5.0/bin/julia'

A fresh approach to technical computing
Documentation: <http://docs.julialang.org>
Type "julia --help" for help.

Version 0.5.0 (2016-09-01)
Official <http://julialang.org>
x86_64-apple-darwin13.0

julia> using Plots
julia> notebook()

SIAM Conference on Computational Science and Engineering

February 27-March 3, 2017
Hilton Atlanta, Atlanta, Georgia, USA

Alan Edelman
David Sanders
Andreas Noack
Jarrett Revels

Click to add notes

Slide 1 of 6 100%



```
Julia Julia Julia Julia Julia Julia Julia Julia Julia Julia
Julia Julia Julia Julia Julia Julia Julia Julia Julia Julia
Julia Julia Julia Julia Julia Julia Julia Julia Julia Julia
Julia Julia Julia Julia Julia Julia Julia 🐼 🐼 🐼 Julia
Julia Julia Julia Julia Julia Julia Julia 🐼 🐼 🐼 Julia
Julia Julia Julia Julia Julia Julia Julia 🐼 🐼 🐼 Julia
Julia Julia Julia Julia Julia Julia Julia Julia Julia Julia
Julia Julia Julia Julia Julia Julia Julia Julia Julia Julia
```

Colors

```
In [87]: using Colors
```

```
In [43]: distinguishable_colors(12)
```

```
Out[43]:
```



```
In [44]: @manipulate for n = 1:80
           distinguishable_colors(n)
         end
```

Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia
Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia
Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia
Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia
Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia	Julia

Colors

```
In [87]: using Colors
```

```
In [88]: distinguishable_colors(12)
```

```
Out[88]:
```



```
In [44]: @manipulate for n = 1:80
           distinguishable_colors(n)
         end
```

x n 40

```
In [87]: using Colors
```

```
In [88]: distinguishable_colors(12)
```

```
Out[88]:
```



```
In [44]: @manipulate for n = 1:80
           distinguishable_colors(n)
       end
```

× n  40

```
Out[44]:
```



```
In [45]: colors = distinguishable_colors(100)
```

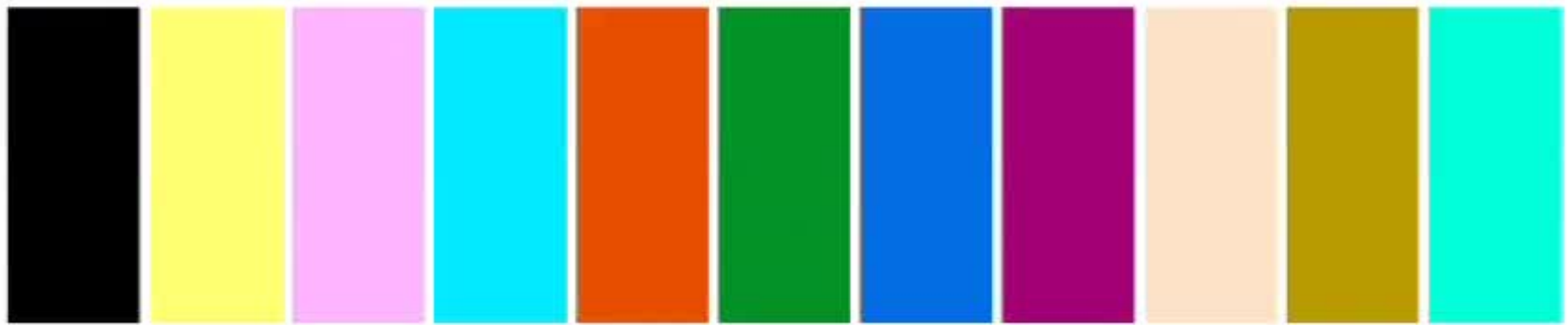
`distinguishable_colors(12)`



```
@manipulate for n = 1:80  
    distinguishable_colors(n)  
end
```

n 40





```
@manipulate for n = 1:80  
  distinguishable_colors(n)  
end
```

n



```
colors = distinguishable_colors(100)
```

```

n = size(A,1)
@manipulate for i in 1:n-2, j in 1:n-2
    insert_block(A, i,j)
end

```

x

i		4
j		4

6]: 10×10 Array{Int64,2}:

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 7 7 7 0 0 0 0
0 0 0 7 7 7 0 0 0 0
0 0 0 7 7 7 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

7]: A = fill{zeros(1,10,10)}

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 7 7 7 0 0 0 0
0 0 0 7 7 7 0 0 0 0
0 0 0 7 7 7 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

```
*]: A = fill(colors[1],10,10)
n =size(A,1)
@manipulate for i in 1:n-2, j in 1:n-2
    insert_block(A, i,j,colors[4])
end
```

```
8]: # Exercise: Create Tetris Pieces, have them fall from the top
```

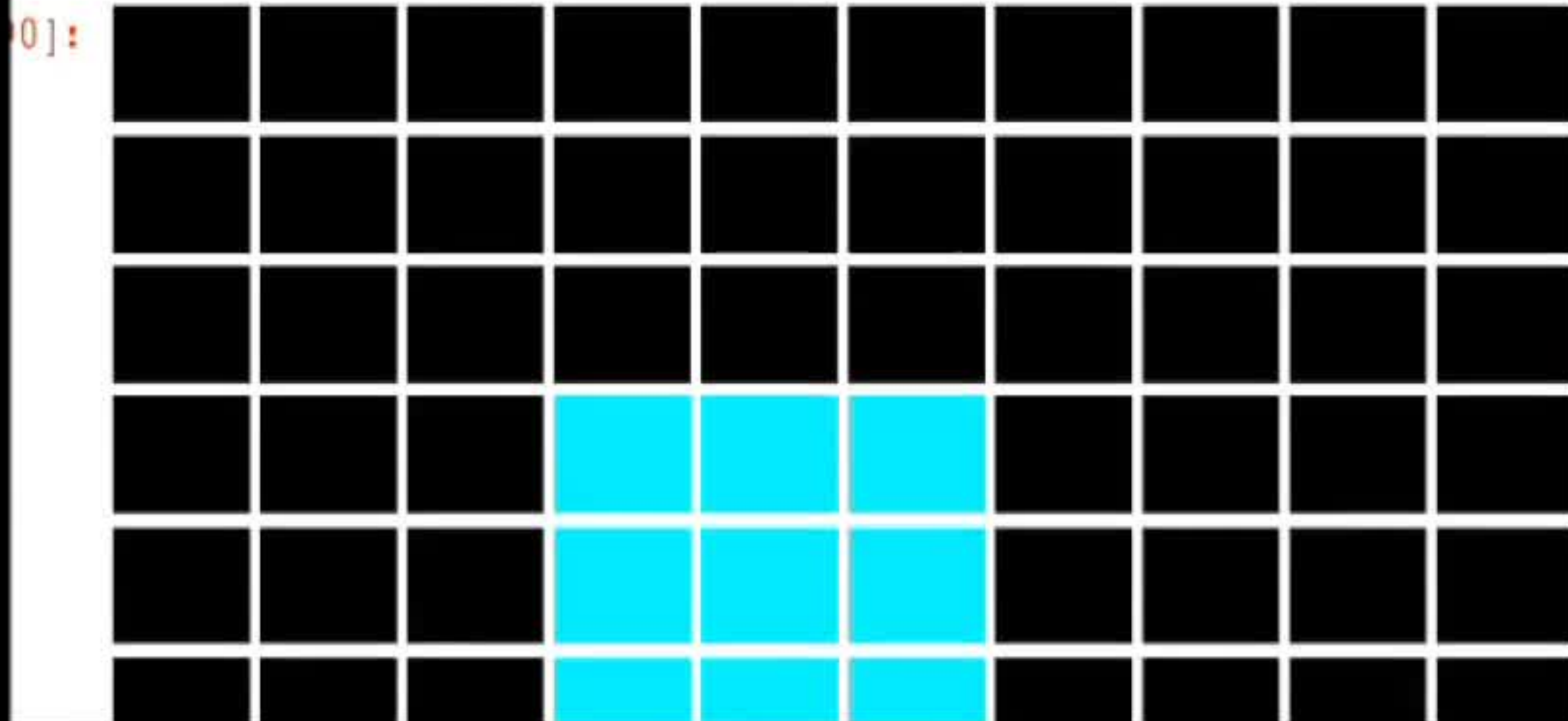
```

0]: A = fill(colors[1],10,10)
    n = size(A,1)
    @manipulate for i in 1:n-2, j in 1:n-2
        insert_block(A, i,j,colors[4])
    end

```

x

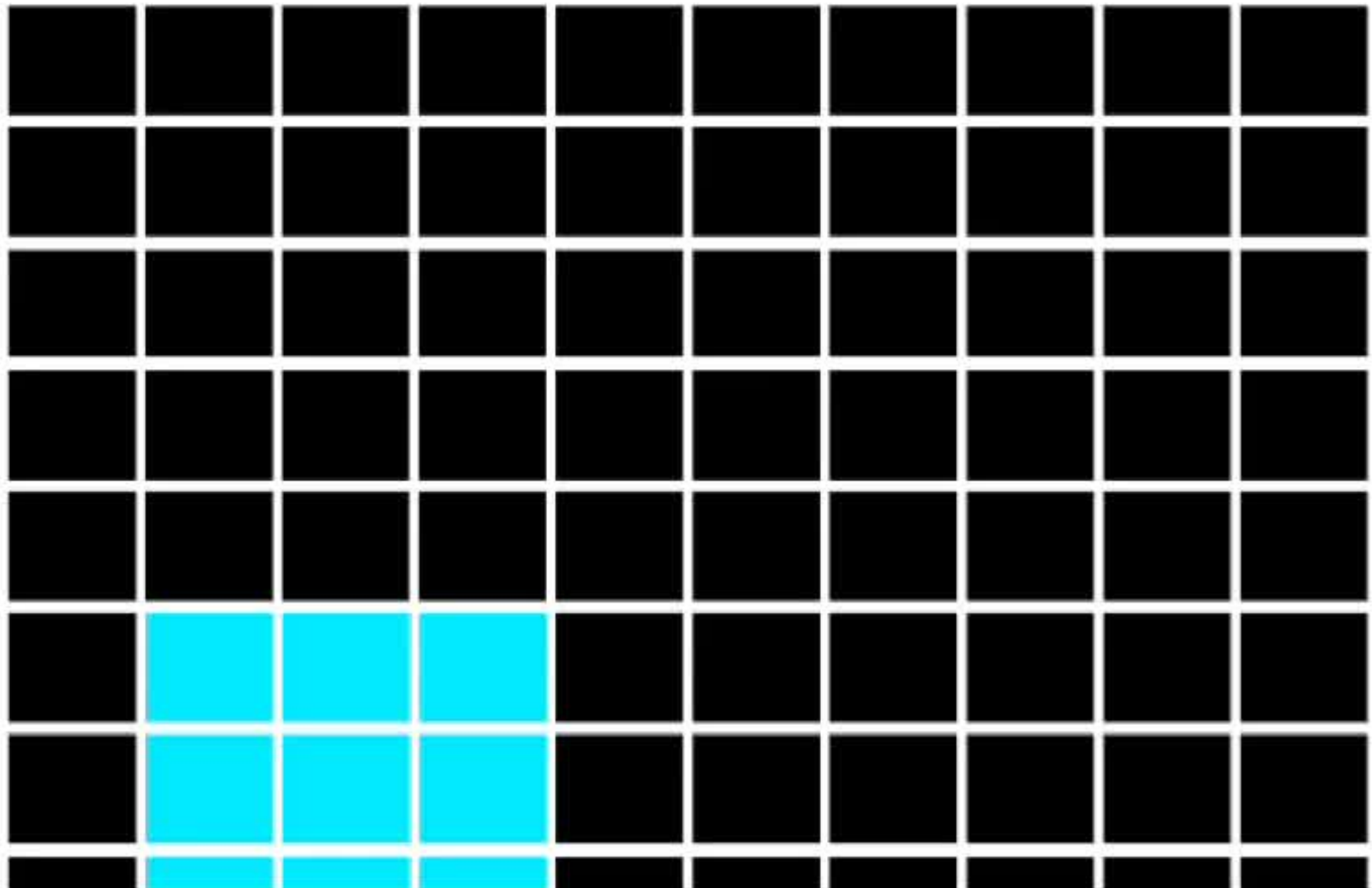
i		4
j		4



j

2

01:



```
In [65]: C_code = """
#include <stddef.h>
double c_sum(size_t n, double *X) {
    double s = 0.0;
    for (size_t i = 0; i < n; ++i) {
        s += X[i];
    }
    return s;
}
"""

const Clib = tempname() # make a temporary file

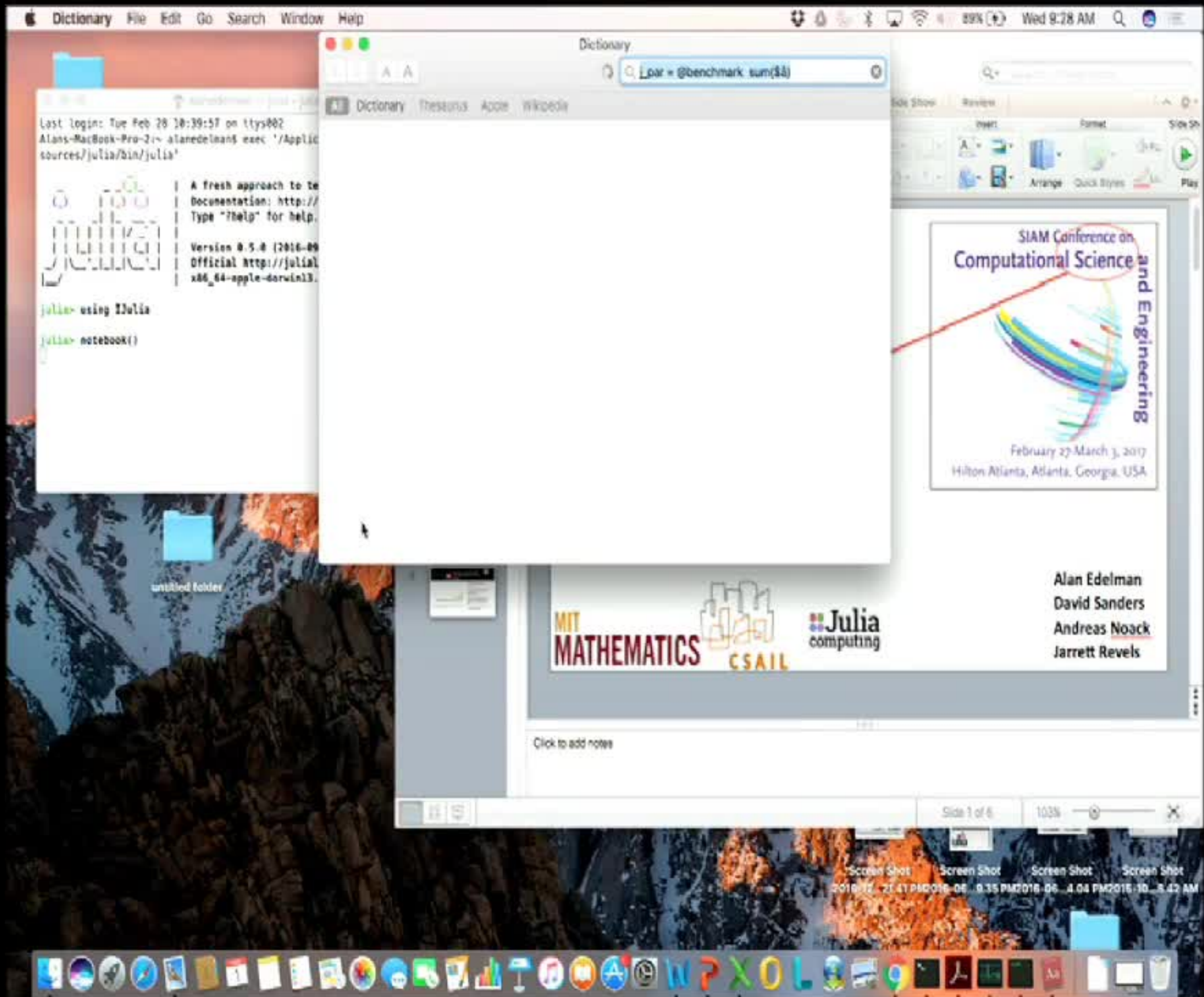
# compile to a shared library by piping C_code to gcc
# (works only if you have gcc installed):

open(`gcc -fPIC -O3 -msse3 -xc -shared -o $(Clib * "." * Libdl.dlext) -`, "w") do f
    print(f, C_code)
end

# define a Julia function that calls the C function:
c_sum(X::Array{Float64}) = ccall(("c_sum", Clib), Float64, (Csize_t, Ptr{Float64}), length(X), X)

WARNING: redefining constant Clib
WARNING: Method definition c_sum(Array{Float64, N{<:Any}}) in module Main at
In[7]:23 overwritten at In[65]:23.
```

Out[65]: c_sum (generic function with 1 method)



```
Chrome File Edit View History Bookmarks People Window Help
2017/SIAM CSE Atlanta) x 1. Getting started, working v... x 2. Julia is Fast x julia/reduce.jl at 3c9d75381: x
GitHub, Inc. [US] https://github.com/JuliaLang/julia/blob/3c9d75381c72d7c32eea75ff187ce77b2d5eff1c8/base/reduce.jl#L229
Apps Presentation Mana... JuliaRun Test your connection receipts - sept Citrix XenApp - Ap... JuliaRun High performance... proposal_guidelines... good Other Bookmarks

225
226 ## sum
227
228 sum(f::Callable, a) = mapreduce(f, +, a)
229 sum(a) = mapreduce(identity, +, a)
230 sum(a::AbstractArray{Bool}) = countnz(a)
231 sumabs(a) = mapreduce(abs, +, a)
232 sumabs2(a) = mapreduce(abs2, +, a)
233
234 # Kahan (compensated) summation: O(1) error growth, at the expense
235 # of a considerable increase in computational expense.
236 function sum_kbn{T<:AbstractFloat}(A::AbstractArray{T})
237     c = r_promote(+, zero(T)::T)
238     if isempty(A)
239         return c
240     end
241     inds = linearindices(A)
242     s = A[first(inds)] + c
243     for i in first(inds)+1:last(inds)
```

j

2

01:

