Solving the two language problem in numerical computing

Viral B. Shah, Jeff Bezanson, Stefan Karpinski, and Alan Edelman

James H. Wilkinson Prize for Numerical Software Lecture

NUMFOCUS
OPEN CODE = BETTER SCIENCE


TensorFlow
C++ 47.3%
Python 41.1%
HTML 5.9%
Jupyter Notebook 2.4%
Go 1.3%
Java 0.7%
Other 1.3%
PYTORCH
Python 32.1%
C++ 29.6%
Cuda 18.0%
C 15.6%
CMake 3.4%
Fortran 0.6%
Other 0.7%
Caffe2
C++ 80.1%
Python 9.1%
Cuda 5.9%
CMake 2.8%
Matlab 0.9%
Makefile 0.7%
Shell 0.5%
Microsoft CNTK
C++ 54.7%
Jupyter Notebook 25.9%
Python 11.5%
Cuda 4.1%
C# 1.1%
Shell 0.9%
Other 1.8%
Knet.jl
Julia 84.5%
Cuda 9.8%
C 1.8%
Makefile 1.7%
Matlab 1.7%
Python 0.4%
Other 0.1%
Flux
Julia 100.0%

Demmel, Dongarra et al. on Polymorphism

The Sca/LAPACK software engineering problem is to express the following metaprogram:

```
(1) for all linear algebra problems (linear systems, eigenproblems, ...)
(2)    for all matrix types (general, symmetric, banded, ...)
(3)       for all data types (real, complex, single, double, higher precision)
(4)          for all machine architectures
                and communication topologies
(5)             for all programming interfaces
(6)                provide the best algorithm(s) available
                   best: performance or accuracy
```

Demmel, Dongara et al. Prospectus for the Next LAPACK and ScaLAPACK Libraries
http://www.netlib.org/lapack/lawnspdf/lawn181.pdf

Celeste.jl: Julia at Peta-scale

Cori: 650,000 cores. 1.3M threads. 60 TB of data.

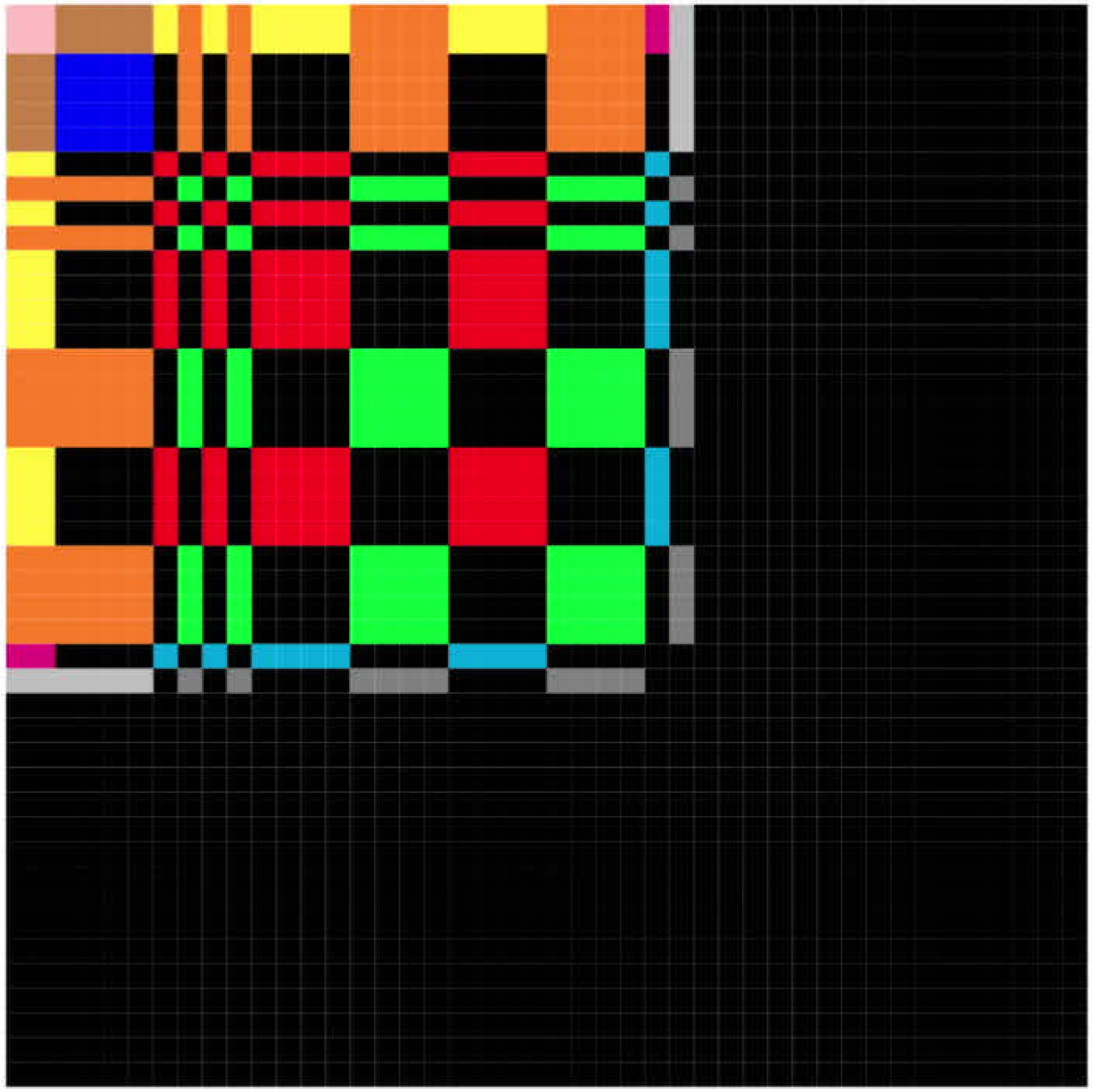
Most light sources are near the detection limit.

Cataloging the Visible Universe through Bayesian Inference at Petascale

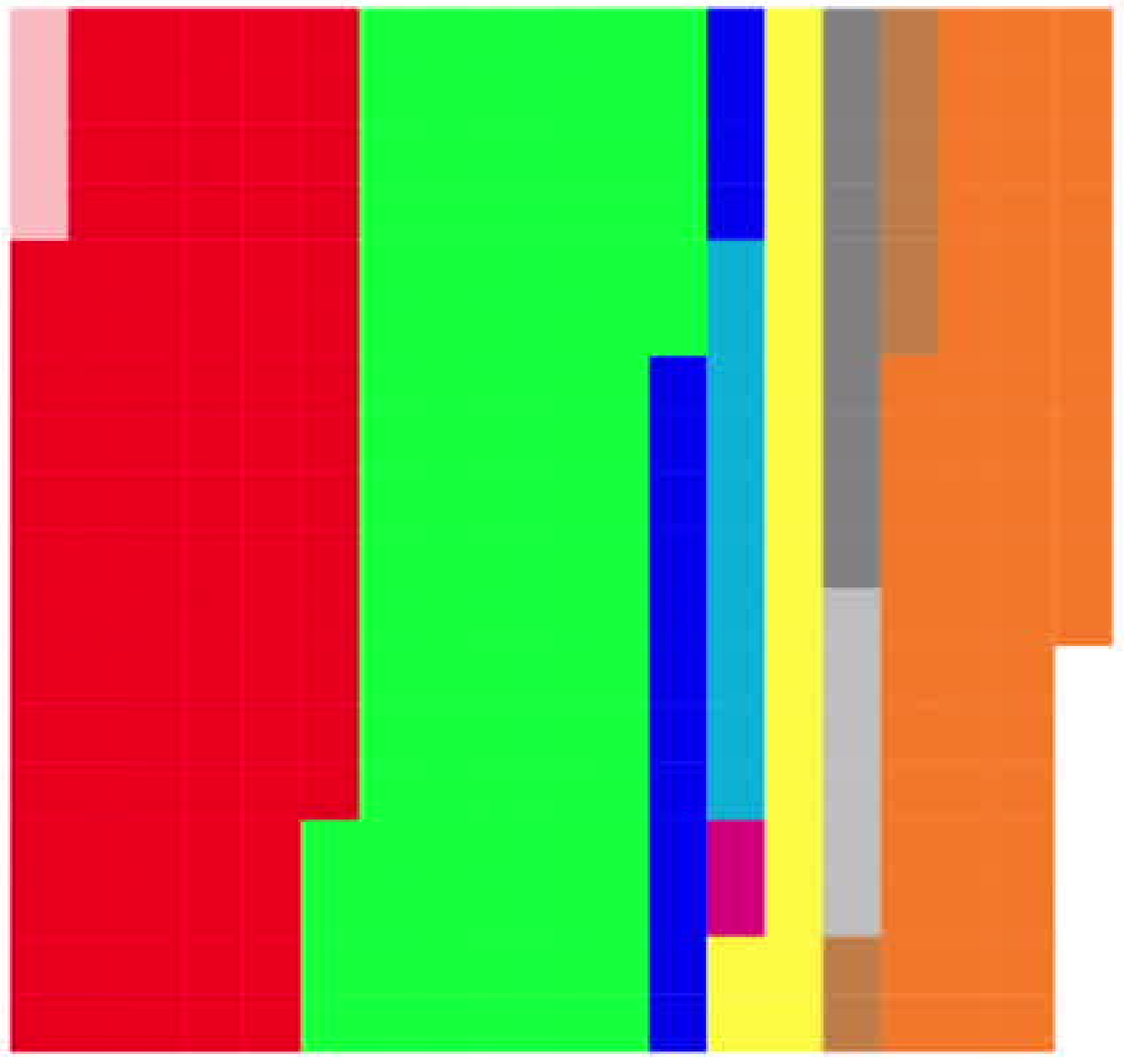
Jeffrey Regier[*], Kiran Pamnany[†], Keno Fischer[‡], Andreas Noack[§], Maximilian Lam[*], Jarrett Revels[§], Steve Howard[¶], Ryan Giordano[¶], David Schlegel[‖], Jon McAuliffe[¶], Rollin Thomas[‖], Prabhat[‖]

[*]*Department of Electrical Engineering and Computer Sciences, University of California, Berkeley*
[†]*Parallel Computing Lab, Intel Corporation*
[‡]*Julia Computing*
[§]*Computer Science and AI Laboratories, Massachusetts Institute of Technology*
[¶]*Department of Statistics, University of California, Berkeley*
[‖]*Lawrence Berkeley National Laboratory*

Celeste: Custom sparsity patterns and storage



Matrix structure

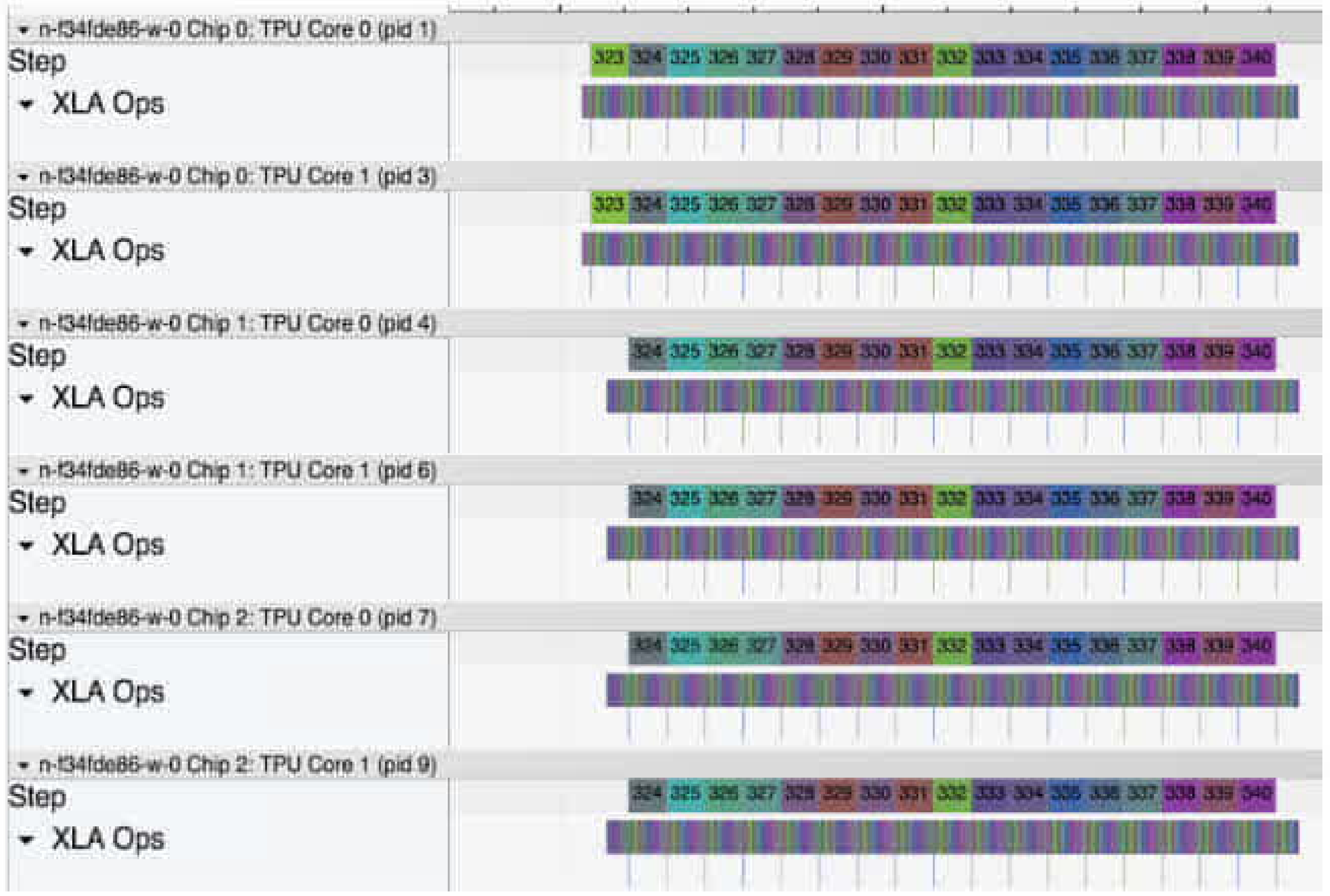


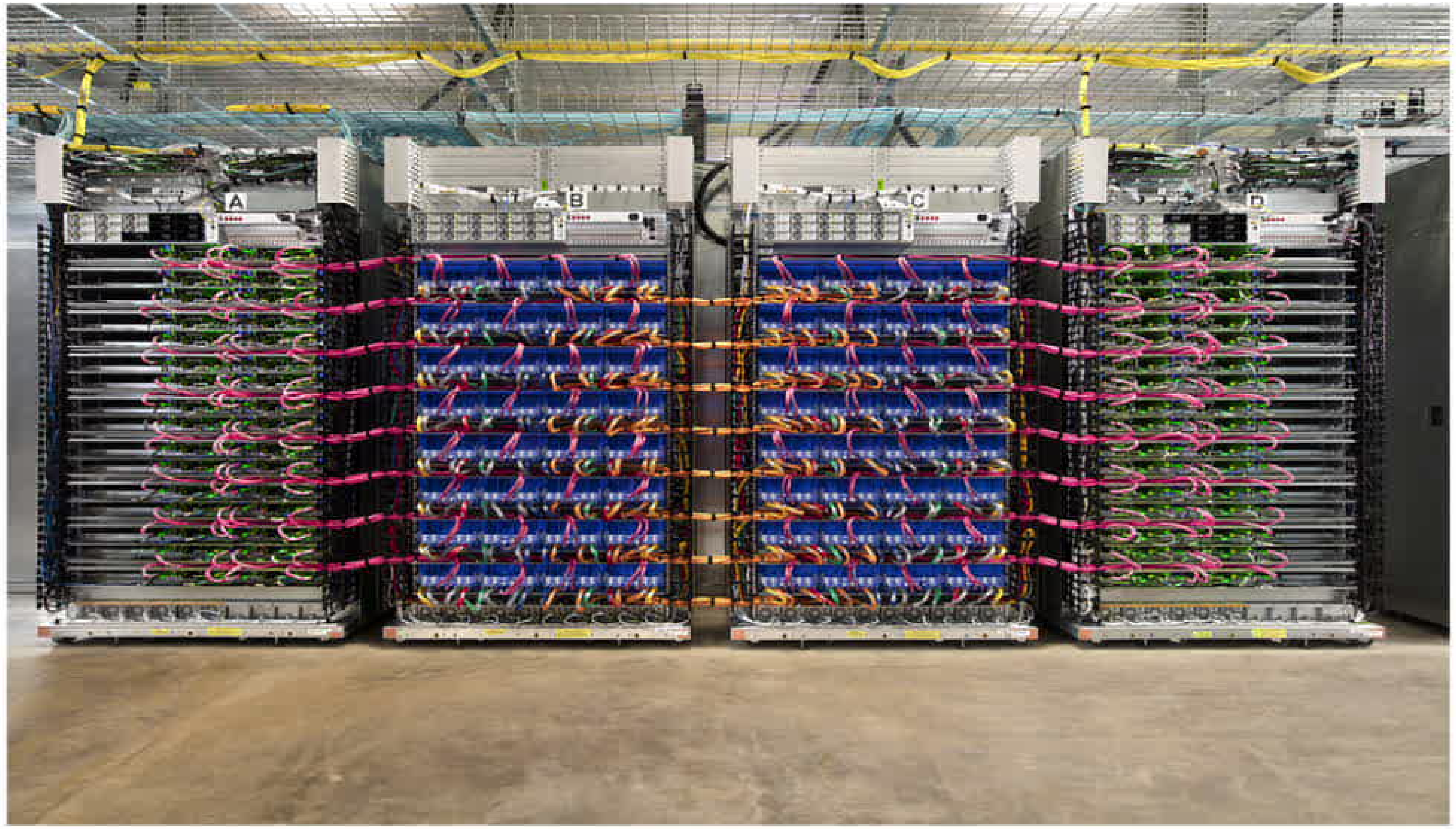
Storage

Julia runs on Google TPUs

Performance on par with TensorFlow on TPUs

Scales to pods (512 TPU cores - 4.3 PF_{16}/s on ResNet50)





Fischer et al. Automatic Full Compilation of Julia Programs and ML Models to Cloud TPUs (*arXiv:1810.09868*)

Zygote.jl: General Purpose Automatic Differentiation

```
function foo(W, Y, x)
    Z = W * Y
    a = Z * x
    b = Y * x
    c = tanh.(b)
    r = a + c
    return r
end
```

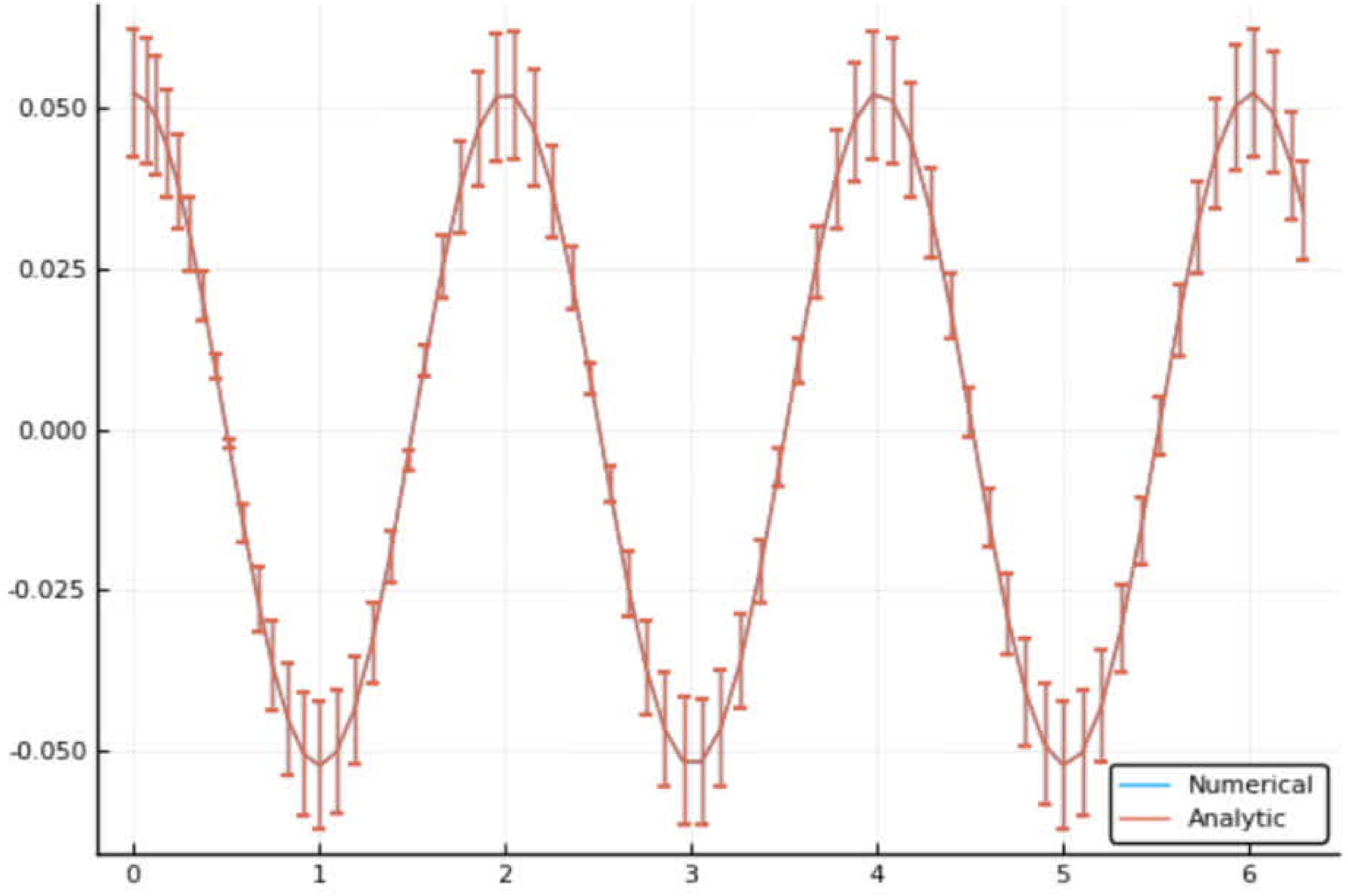
Zygote

```
function ∇foo(W, Y, x)
    Z = W * Y
    a = Z * x
    b = Y * x
    c, 𝒥tanh = ∇tanh.(b)
    a + c, function (Δr)
        Δc = Δr, Δa = Δr
        (Δtanh, Δb) = 𝒥tanh(Δc)
        (ΔY, Δx) = (Δb * x', Y' * Δb)
        (ΔZ = Δa * x', Δx += Z' * Δa)
        (ΔW = ΔZ * Y', ΔY = W' * ΔZ')
        (nothing, ΔW, ΔY, Δx)
    end
end
```

M. Innes. Don't Unroll Adjoint: Differentiating SSA-Form Programs (arXiv:1810.07951)

Composability: DifferentialEquations.jl + Measurements.jl

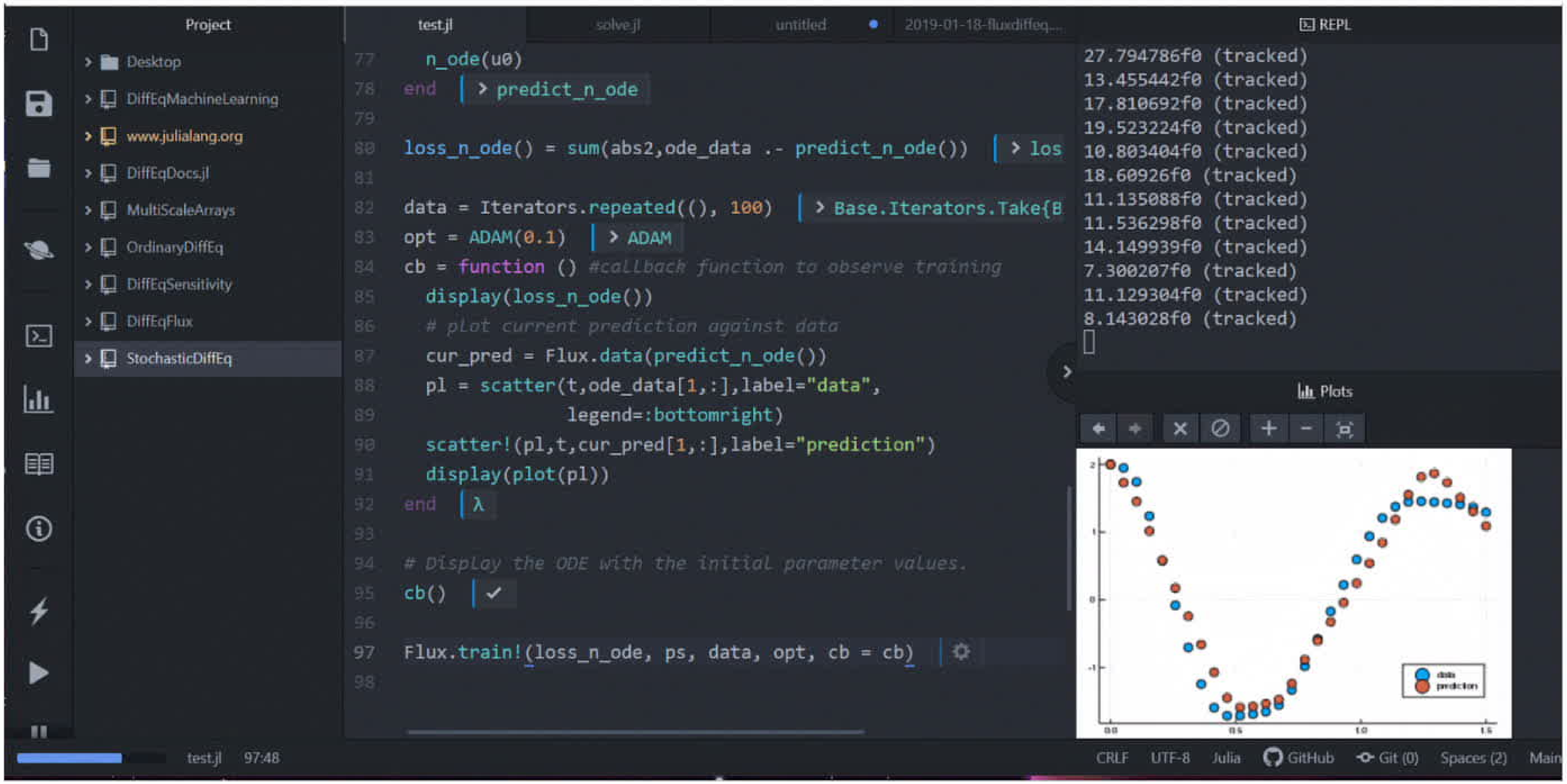
```
g = 9.79 ± 0.02 # Gravitational constants
L = 1.00 ± 0.01 # Length of the pendulum

# Initial speed & angle, time span
u₀ = [0 ± 0, π/60 ± 0.01]
tspan = (0.0, 6.3)

# Define the problem
function pendulum(du, u, p, t)
    θ  = u[1]
    dθ = u[2]
    du[1] = dθ
    du[2] = -(g/L)*θ
end

# Pass to solvers
prop = ODEProblem(pendulum, u₀, tspan)
sol = solve(prob, Tsit5(), reltol = 1e-6)

# Analytic solution
u = u₀[2] .* cos.(sqrt(g/L) .* sol.t)
```



Rackauckas et al. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. 2017. (Journal of Open Research Software)

Giordano. Uncertainty propagation with functionally correlated quantities (arXiv:1610.08716)

Composability: DifferentialEquations.jl + Flux.jl (Neural ODEs)



Rackauckas et al. DiffEqFlux.jl - A Julia Library for Neural Differential Equations ([arXiv:1902.02376](https://arxiv.org/abs/1902.02376))

Other special matrix types

- `Diagonal`
- `UniformScaling`
- `Symmetric`, `Hermitian`
- `LowerTriangular`, `UpperTriangular`
- `Bidiagonal`, `Tridiagonal`, `SymTridiagonal`
- `Adjoint`, `Transpose`

Recent work: type system formalization

Julia Subtyping: a Rational Reconstruction

F. Zappa Nardelli, J. Belyakova, A. Pelenitsyn, B. Chung, J. Bezanson, J. Vitek

OOPSLA 2018

Paper: https://www.di.ens.fr/~zappa/projects/lambdajulia/

$$\frac{}{E \vdash t \;<:\; \texttt{Any} \vdash E}\;[\textsc{top}]$$

$$\frac{}{E \vdash a \;<:\; a \vdash E}\;[\textsc{refl}]$$

$$\frac{E \vdash a_1 \;<:\; a_1' \vdash E_1 \;\;..\;\; E_{n-1} \vdash a_n \;<:\; a_n' \vdash E_n \qquad consistent(E_n)}{E \vdash \texttt{Tuple}\{a_1, .., a_n\} \;<:\; \texttt{Tuple}\{a_1', .., a_n'\} \vdash E_n}\;[\textsc{tuple}]$$

$$\frac{t' = lift\_union(\texttt{Tuple}\{a_1, .., a_n\}) \qquad E \vdash t' \;<:\; t \vdash E'}{E \vdash \texttt{Tuple}\{a_1, .., a_n\} \;<:\; t \vdash E'}\;[\textsc{tuple\_lift\_union}]$$

$$\frac{t' = unlift\_union(\texttt{Union}\{t_1, .., t_n\}) \qquad E \vdash t \;<:\; t' \vdash E'}{E \vdash t \;<:\; \texttt{Union}\{t_1, .., t_n\} \vdash E'}\;[\textsc{tuple\_unlift\_union}]$$

$$\frac{E \vdash t_1 \;<:\; t \vdash E_1 \;\;..\;\; reset\_occ_E(E_{n-1}) \vdash t_n \;<:\; t \vdash E_n}{E \vdash \texttt{Union}\{t_1, .., t_n\} \;<:\; t \vdash max\_occ_{E_1..E_n}(E_n)}\;[\textsc{union\_left}]$$

$$\frac{\exists j.\, E \vdash t \;<:\; t_j \vdash E'}{E \vdash t \;<:\; \texttt{Union}\{t_1, .., t_n\} \vdash E'}\;[\textsc{union\_right}]$$

$$\frac{n \leq m \quad E_0 = add(\text{Barrier}, E) \qquad \forall\, 0 < i \leq n.\;\; E_{i-1} \vdash a_i \;<:\; a_i' \vdash E_i' \;\wedge\; E_i' \vdash a_i' \;<:\; a_i \vdash E_i}{E \vdash name\{a_1, .., a_m\} \;<:\; name\{a_1', .., a_n'\} \vdash del(\text{Barrier}, E_n)}\;[\textsc{app\_inv}]$$

$$\frac{name\{\top_1, .., \top_m, ..\} \;<:\; t'' \in tds \qquad E \vdash t''[a_1/\top_1 .. a_m/\top_m] \;<:\; t' \vdash E'}{E \vdash name\{a_1, .., a_m\} \;<:\; t' \vdash E'}\;[\textsc{app\_super}]$$

$$\frac{add({}^{L}\top_{t_1}^{t_2}, E) \vdash t \;<:\; t' \vdash E'}{E \vdash t\;\textbf{where}\;t_1 <: \top <: t_2 \;<:\; t' \vdash del(\top, E')}\;[\textsc{L\_intro}]$$

$$\frac{add({}^{R}\top_{t_1}^{t_2}, E) \vdash t \;<:\; t' \vdash E' \qquad consistent(\top, E')}{E \vdash t \;<:\; t'\;\textbf{where}\;t_1 <: \top <: t_2 \vdash del(\top, E')}\;[\textsc{R\_intro}]$$

$$\frac{search(\top, E) = {}^{L}\top^{u}_{l} \quad E \vdash u \;<:\; t \vdash E'}{E \vdash \top \;<:\; t \vdash E'}\;[\text{L\_LEFT}]$$

$$\frac{search(\top, E) = {}^{L}\top^{u}_{l} \quad E \vdash t \;<:\; l \vdash E'}{E \vdash t \;<:\; \top \vdash E'}\;[\text{L\_RIGHT}]$$

$$\frac{search(\top_1, E) = {}^{R}\top_{1\,l_1}^{\;u_1} \quad search(\top_2, E) = {}^{L}\top_{2\,l_2}^{\;u_2} \quad outside(\top_1, \top_2, E) \Rightarrow E \vdash u_2 \;<:\; l_2 \vdash E' \quad E \vdash u_1 \;<:\; l_2 \vdash E''}{E \vdash \top_1 \;<:\; \top_2 \vdash upd({}^{R}\top_{1\,\texttt{Union}\{\top_1, l_1\}}^{\;u_1}, E')}\;[\text{R\_L}]$$

$$\frac{search(\top, E) = {}^{R}\top^{u}_{l} \quad E \vdash l \;<:\; t \vdash E'}{E \vdash \top \;<:\; t \vdash upd({}^{R}\top^{t}_{l}, E')}\;[\text{R\_LEFT}]$$

$$\frac{search(\top, E) = {}^{R}\top^{u}_{l} \quad (is\_var(t) \land search(t, E) = {}^{L}\mathsf{S}^{u_1}_{l_1}) \Rightarrow \neg outside(\top, \mathsf{S}, E) \quad E \vdash t \;<:\; u \vdash E'}{E \vdash t \;<:\; \top \vdash upd({}^{R}\top^{u}_{\texttt{Union}\{l, t\}}, E')}\;[\text{R\_RIGHT}]$$

$$\frac{\neg is\_var(a_1) \quad E \vdash typeof(a_1) \;<:\; t_2 \vdash E'}{E \vdash \texttt{Type}\{a_1\} \;<:\; t_2 \vdash E'}\;[\text{TYPE\_LEFT}]$$

$$\frac{is\_kind(t_1) \quad is\_var(t_2) \quad E \vdash \texttt{Type}\{\top\}\;\textbf{where}\;\top \;<:\; \texttt{Type}\{t_2\} \vdash E'}{E \vdash t_1 \;<:\; \texttt{Type}\{t_2\} \vdash E'}\;[\text{TYPE\_RIGHT}]$$

$$\frac{add(\text{Barrier}, E) \vdash a_1 \;<:\; a_2 \vdash E' \quad E' \vdash a_2 \;<:\; a_1 \vdash E''}{E \vdash \texttt{Type}\{a_1\} \;<:\; \texttt{Type}\{a_2\} \vdash del(\text{Barrier}, E'')}\;[\text{TYPE\_TYPE}]$$

Reproducibility by default: `Manifest.toml` files

```toml
# This file is machine-generated - editing it directly is not advised

[[UnicodePlots]]
deps = ["Dates", "Random", "SparseArrays", "StatsBase", "Test"]
git-tree-sha1 = "b8e58d4390ccebfa4f3bf502a45e08066eec3bf9"
uuid = "b8865327-cd53-5732-bb35-84acbb429228"
version = "1.1.0"

[[OrderedCollections]]
deps = ["Random", "Serialization", "Test"]
git-tree-sha1 = "85619a3f3e17bb4761fe1b1fd47f0e979f964d5b"
uuid = "bac558e1-5e72-5ebc-8fee-abe8a469f55d"
version = "1.0.2"

[[SortingAlgorithms]]
deps = ["DataStructures", "Random", "Test"]
git-tree-sha1 = "03f5898c9959f8115e30bc7226ada7d0df554ddd"
uuid = "a2af1166-a08f-5f64-846c-94a0d3cef48c"
version = "0.3.1"
```

Fix deprecations #111

Merged ararslan merged 2 commits into `master` from `fbot/deps` on Sep 25, 2018

Conversation 4 | Commits 2 | Checks 1 | Files changed 1

femtocleaner bot commented on Aug 8, 2018 — Contributor

I fixed a number of deprecations for you

femtocleaner bot commented on Sep 14, 2018 — Author — Contributor

My code has been updated. I now view the world differently.
Am I still the same bot I was before?
In any case, I've updated this PR to reflect my new knowledge. I hope you like it.

femtocleaner bot force-pushed the `fbot/deps` branch from **879a084** to **44c9b89** on Sep 14, 2018

Reviewers

ararslan

Assignees

No one—assign

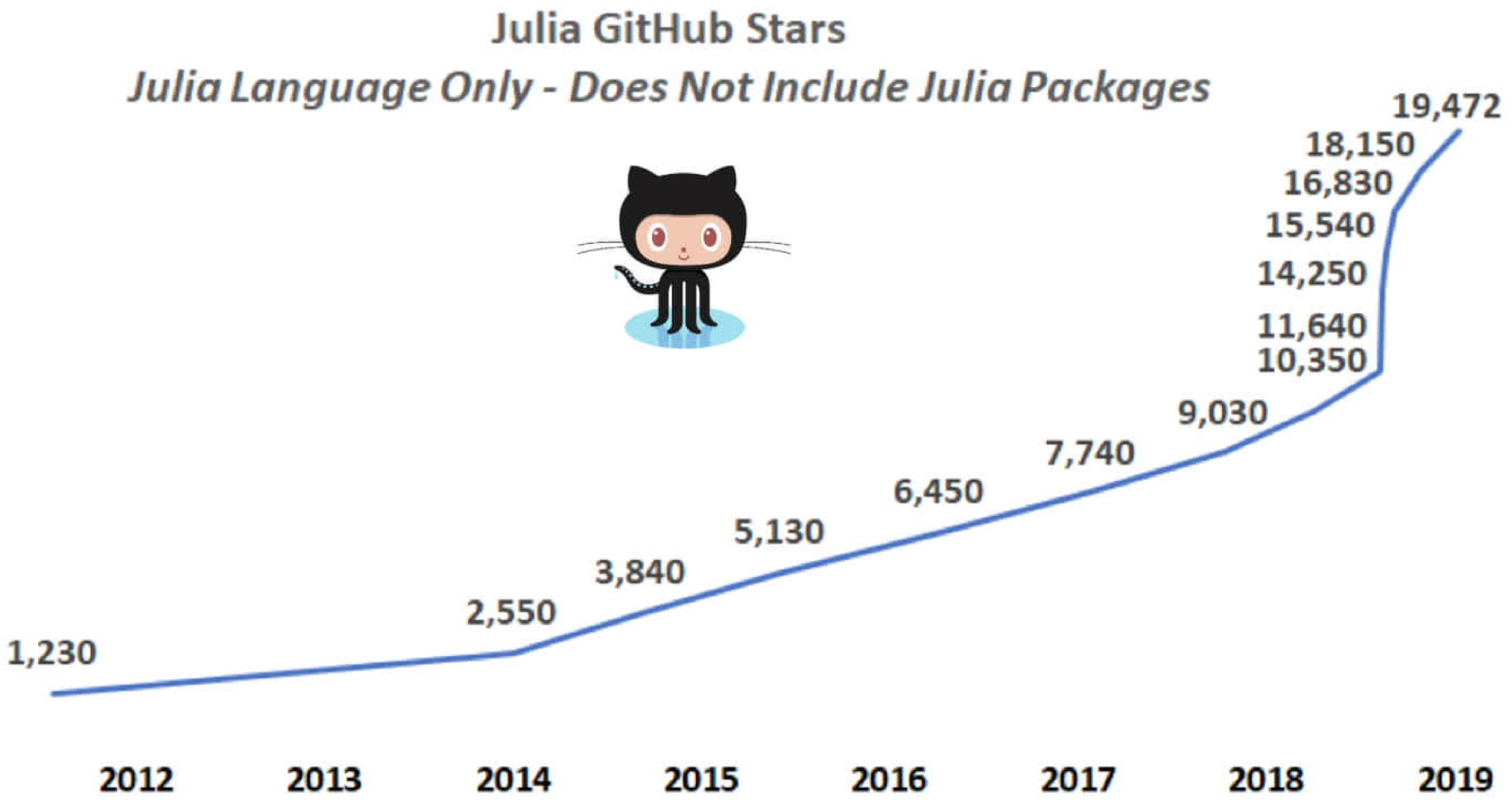
Labels

None yet

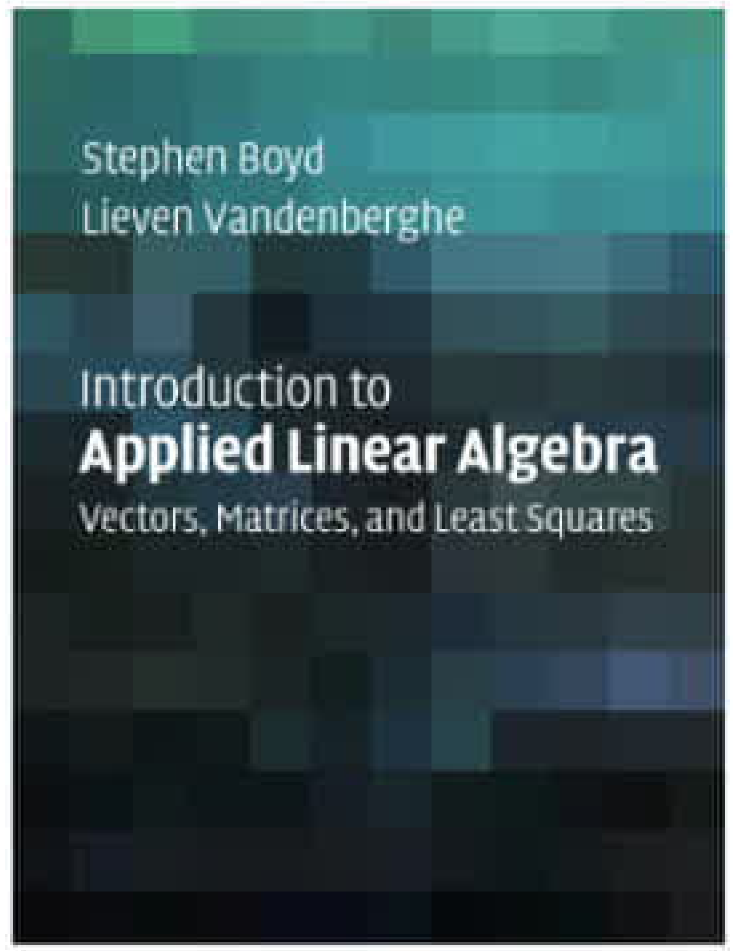
Projects

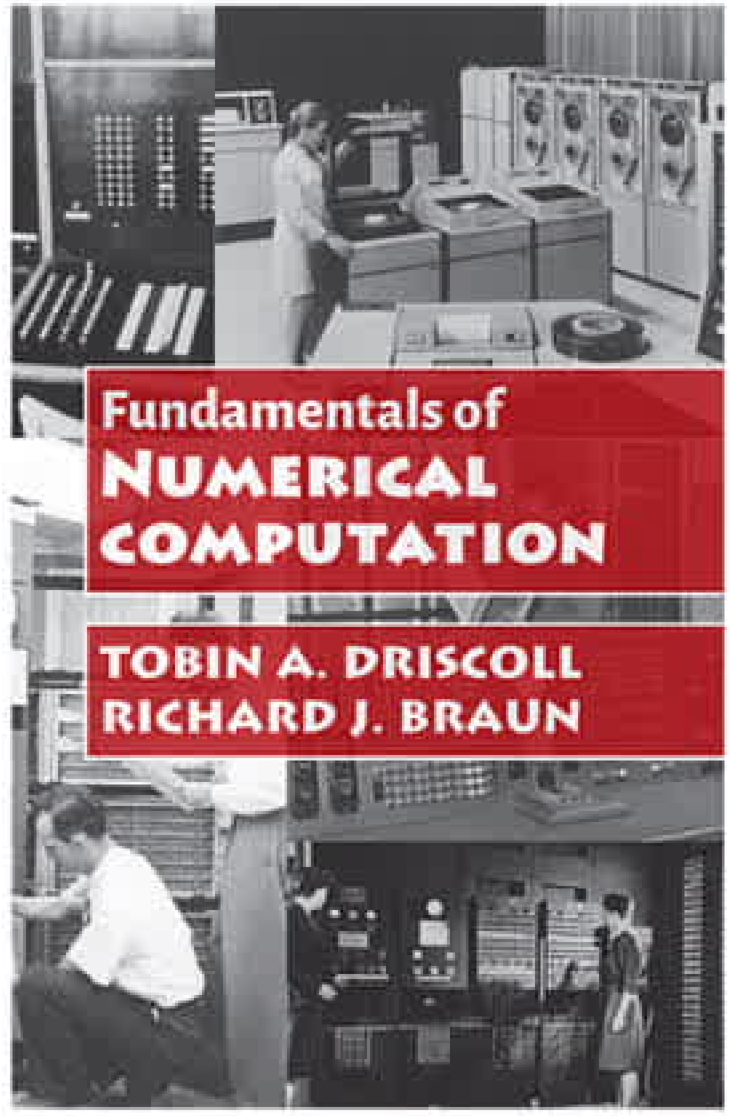
None yet

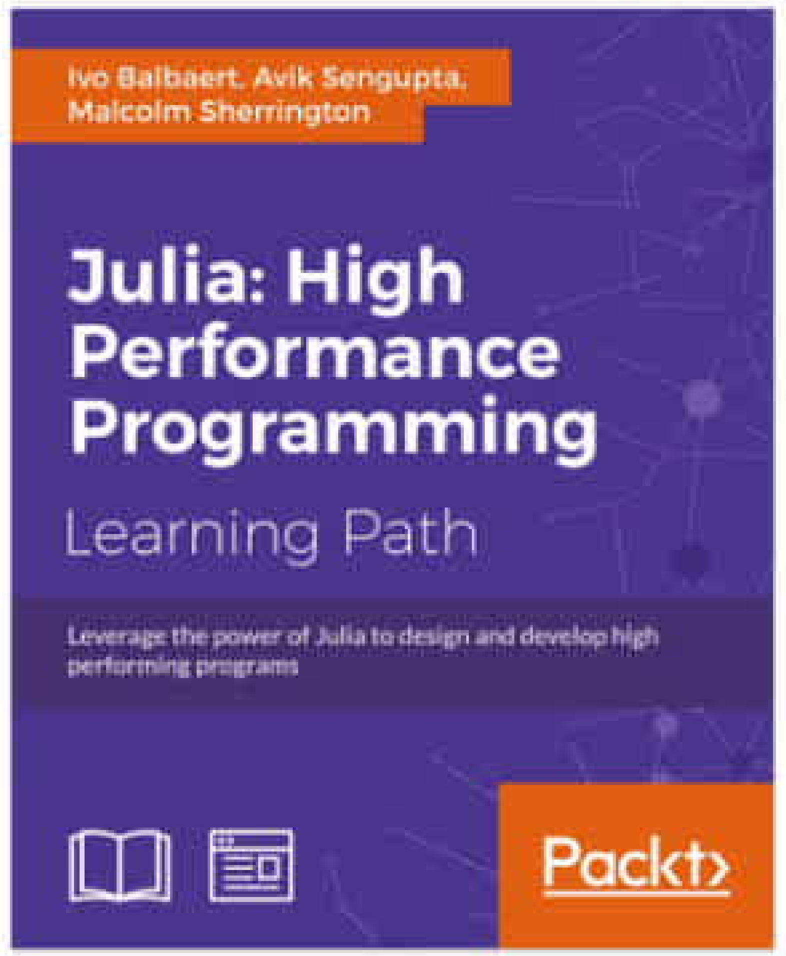
A Growing Community

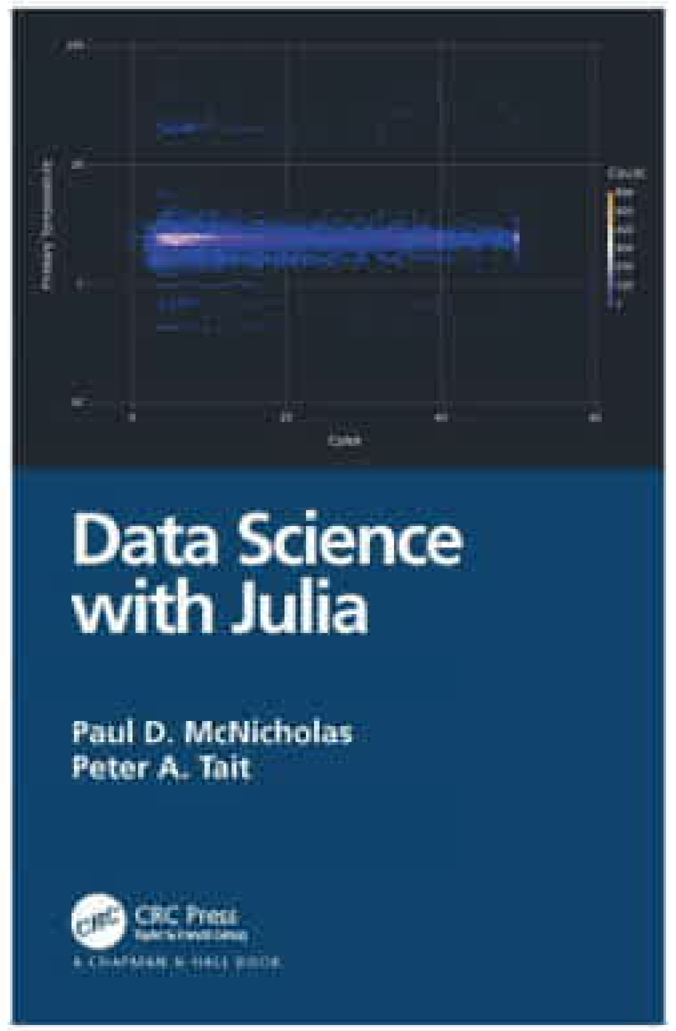


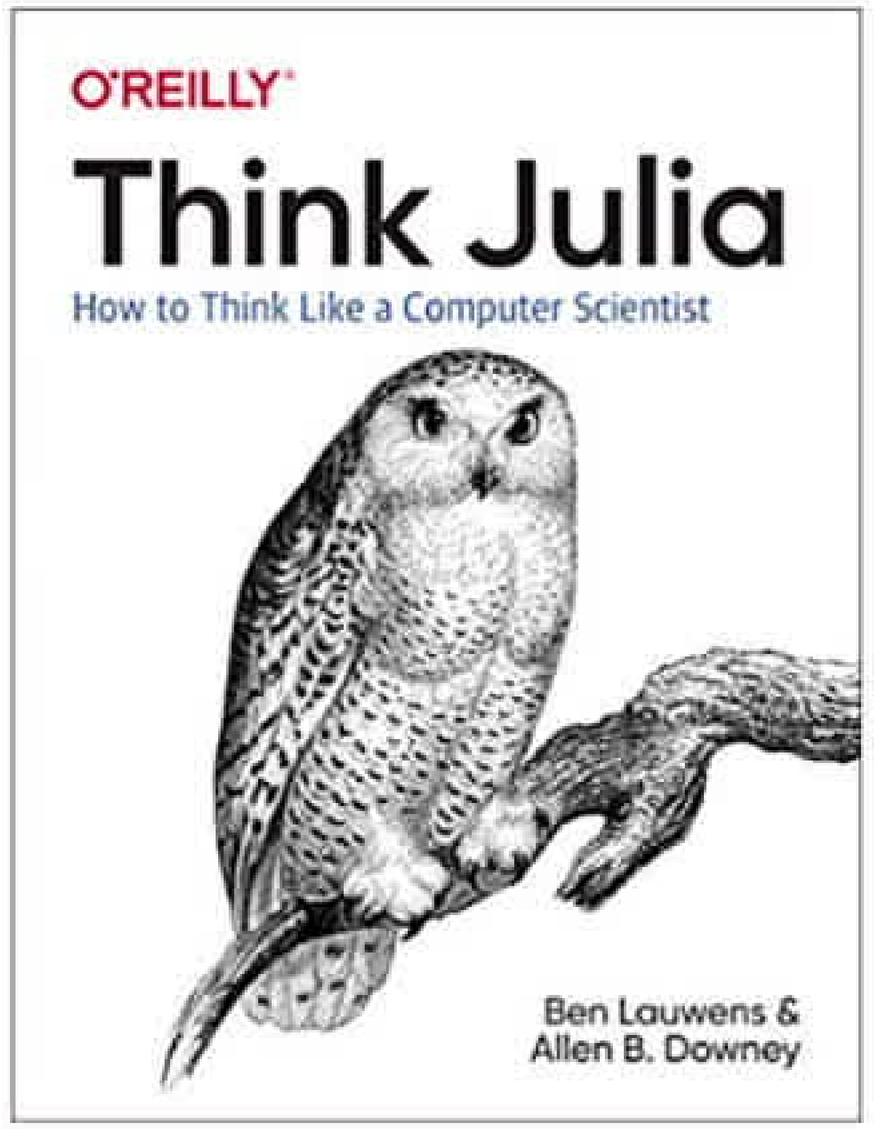
Books

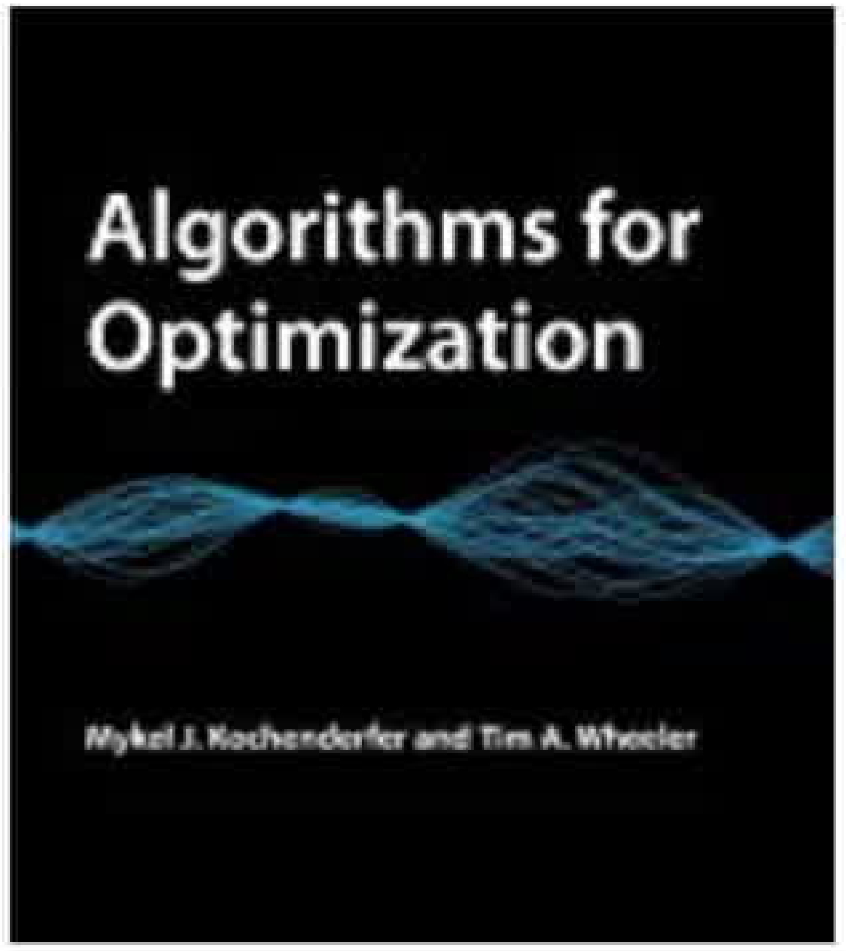

Stephen Boyd
Lieven Vandenberghe
Introduction to
Applied Linear Algebra
Vectors, Matrices, and Least Squares


Fundamentals of
Numerical
Computation
Tobin A. Driscoll
Richard J. Braun

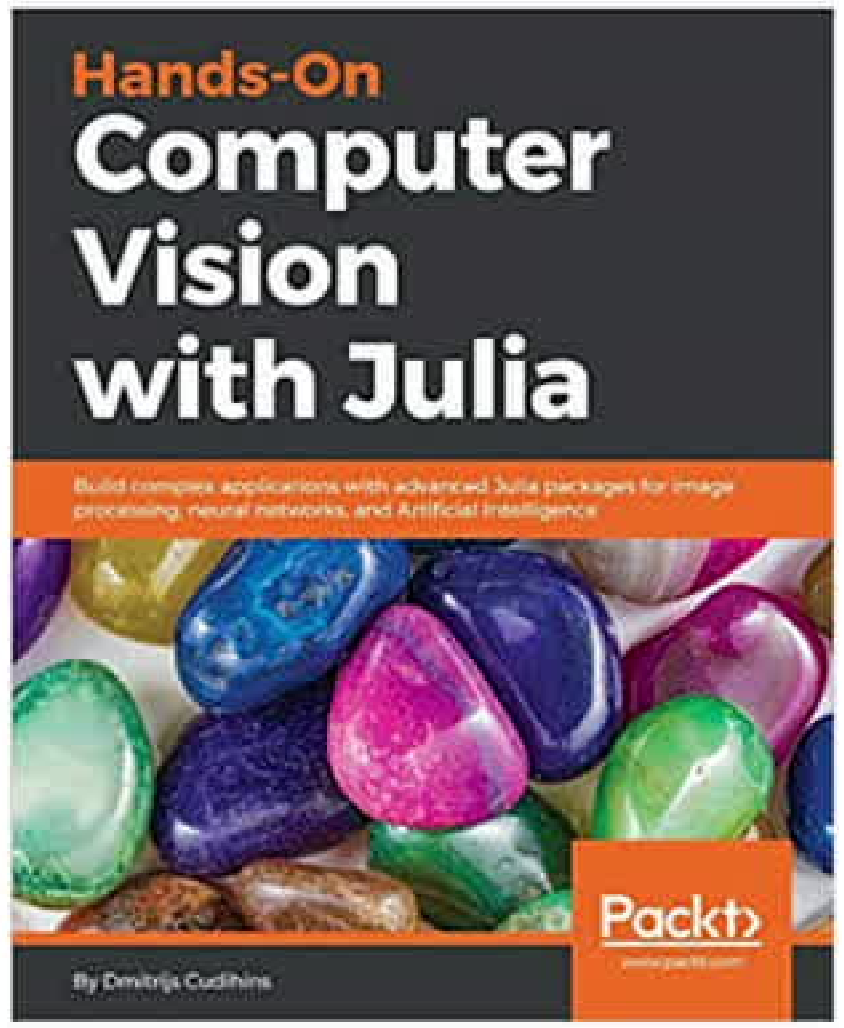

Ivo Balbaert, Avik Sengupta,
Malcolm Sherrington
Julia: High
Performance
Programming
Learning Path
Leverage the power of Julia to design and develop high
performing programs
Packt>


Data Science
with Julia
Paul D. McNicholas
Peter A. Tait
CRC Press

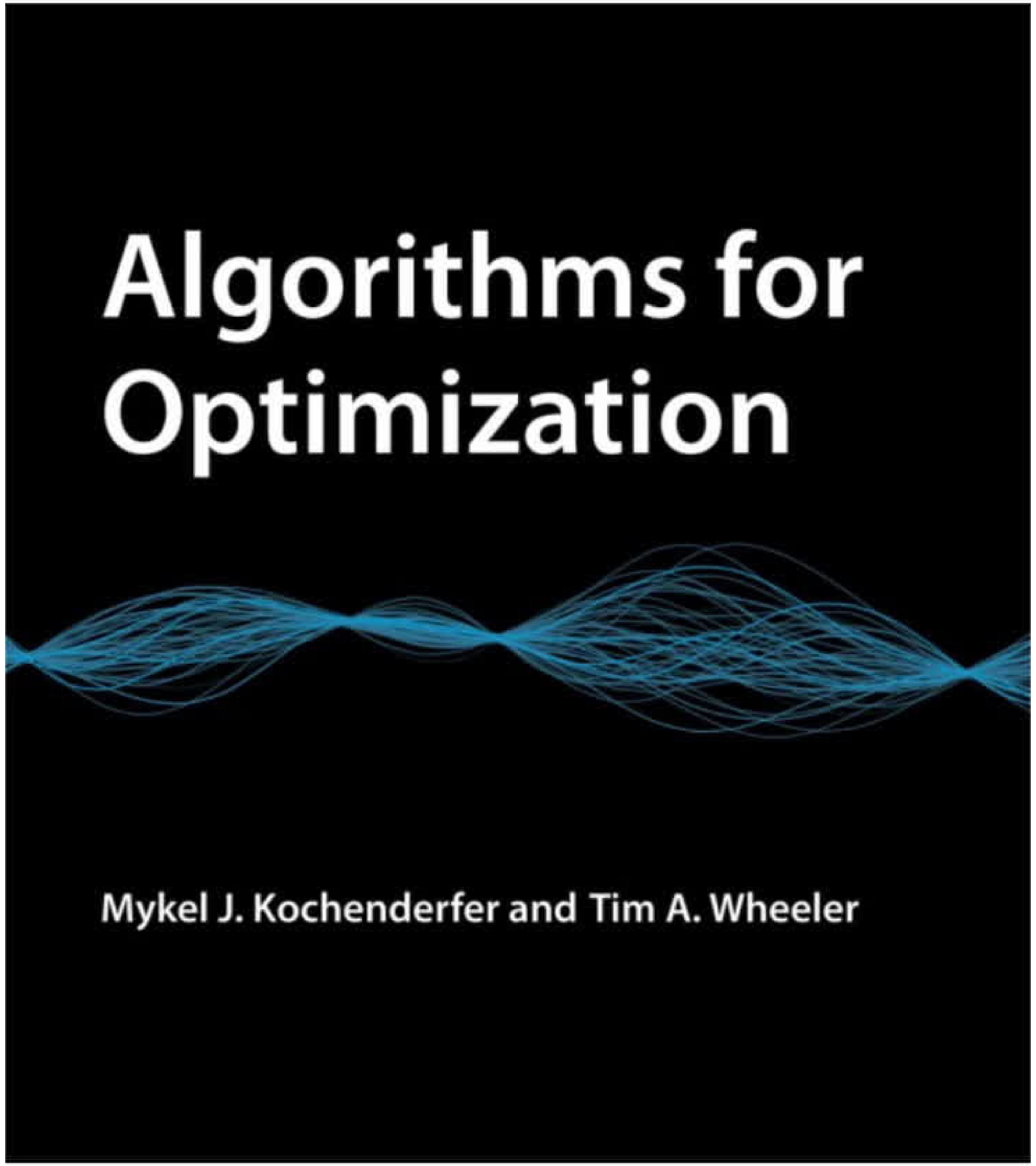

O'REILLY
Think Julia
How to Think Like a Computer Scientist
Ben Lauwens &
Allen B. Downey


Algorithms for
Optimization
Mykel J. Kochenderfer and Tim A. Wheeler


JULIA
PROGRAMMING
for
OPERATIONS RESEARCH
A Primer on Computing
Changhyun Kwon


Hands-On
Computer
Vision
with Julia
Packt>

Books


Algorithms for Optimization
Mykel J. Kochenderfer and Tim A. Wheeler

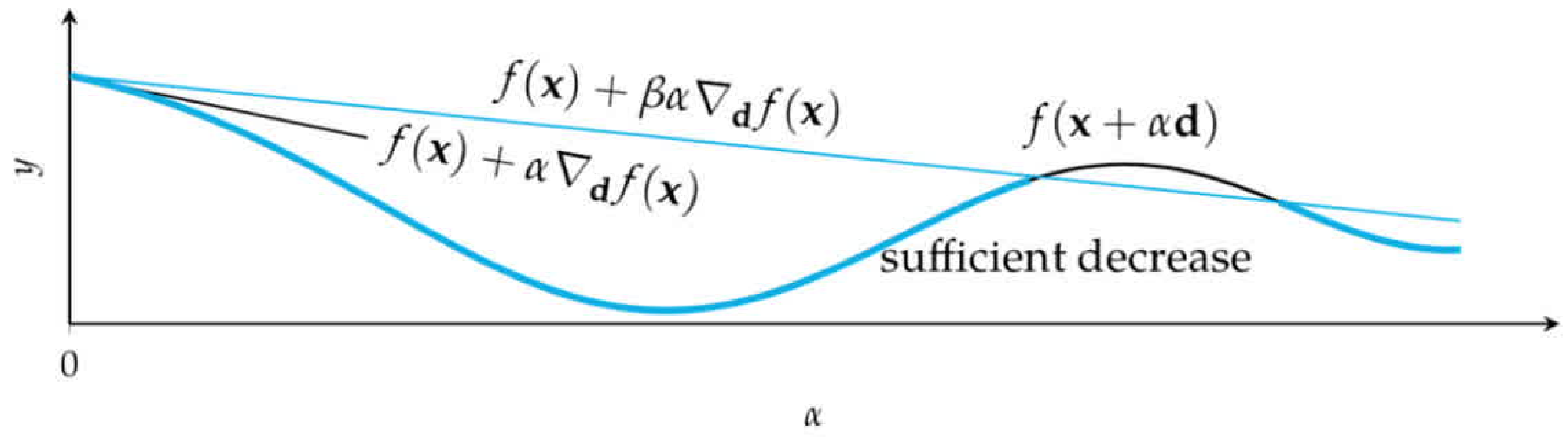


Figure 4.1. The sufficient decrease condition, the first Wolfe condition, can always be satisfied by a sufficiently small step size along a descent direction.

If $\mathbf{d}$ is a valid descent direction, then there must exist a sufficiently small step size that satisfies the sufficient decrease condition. We can thus start with a large step size and decrease it by a constant reduction factor until the sufficient decrease condition is satisfied. This algorithm is known as *backtracking line search*[6] because of how it backtracks along the descent direction. Backtracking line search is shown in figure 4.2 and implemented in algorithm 4.2. We walk through the procedure in example 4.2.

[6] Also known as *Armijo line search*, L. Armijo, "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

```
function backtracking_line_search(f, ∇f, x, d, α; p=0.5, β=1e-4)
    y, g = f(x), ∇f(x)
    while f(x + α*d) > y + β*α*(g⋅d)
        α *= p
    end
    α
end
```

Algorithm 4.2. The backtracking line search algorithm, which takes objective function `f`, its gradient `∇f`, the current design point `x`, a descent direction `d`, and the maximum step size `α`. We can optionally specify the reduction factor `p` and the first Wolfe condition parameter `β`.

```
function direct(f, a, b, ϵ, k_max)
    g = reparameterize_to_unit_hypercube(f, a, b)
    intervals = Intervals()
    n = length(a)
    c = fill(0.5, n)
    interval = Interval(c, g(c), fill(0, n))
    add_interval!(intervals, interval)
    c_best, y_best = copy(interval.c), interval.y

    for k in 1 : k_max
        S = get_opt_intervals(intervals, ϵ, y_best)
        to_add = Interval[]
        for interval in S
            append!(to_add, divide(g, interval))
            dequeue!(intervals[min_depth(interval)])
        end
        for interval in to_add
            add_interval!(intervals, interval)
            if interval.y < y_best
                c_best, y_best = copy(interval.c), interval.y
            end
        end
    end

    return rev_unit_hypercube_parameterization(c_best, a, b)
end
```

Algorithm 7.8. DIRECT, which takes the multidimensional objective function `f`, vector of lower bounds `a`, vector of upper bounds `b`, tolerance parameter `ϵ`, and number of iterations `k_max`. It returns the best coordinate.

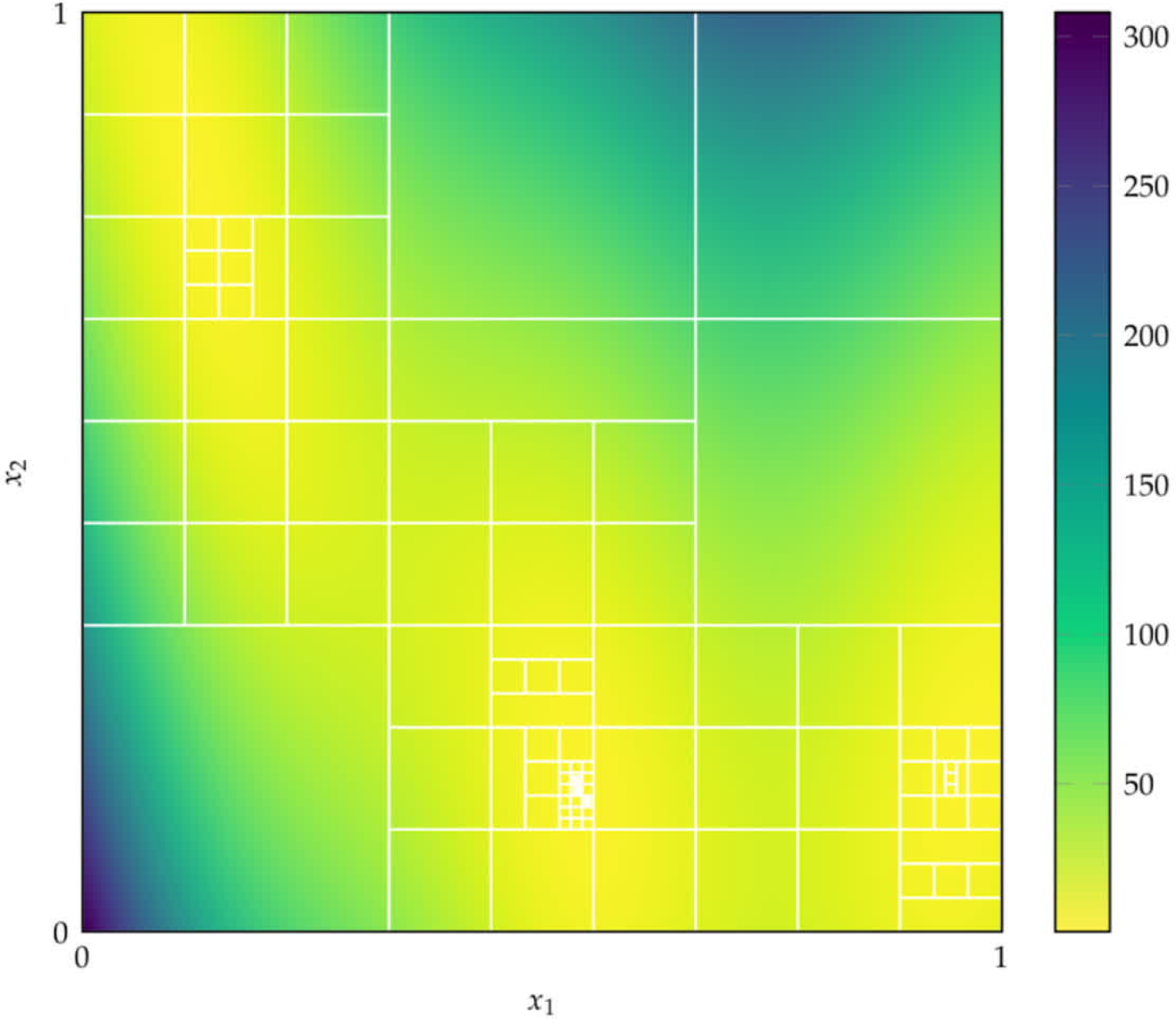


Figure 7.20. The DIRECT method after 16 iterations on the Branin function, appendix B.3. Each cell is bordered by white lines. The cells are much denser around the minima of the Branin function, as the DIRECT method procedurally increases its resolution in those regions.

Some of the Universities Teaching Julia

BROWN UNIVERSITY
CORNELL UNIVERSITY FOUNDED A.D. 1865
EMORY UNIVERSITY
NORTHEASTERN UNIVERSITY BOSTON, MASSACHUSETTS
PSU PENN STATE UNIVERSITY
PURDUE
SZKOŁA GŁÓWNA HANDLOWA W WARSZAWIE
Stanford University
THE UNIVERSITY OF EDINBURGH
UNIVERSITY of GLASGOW
UNIVERSITY of SOUTH FLORIDA 1956
UNIVERSITE PAUL SABATIER TOULOUSE III
UNIVERSITY OF CONNECTICUT 1881
FIDE SPLENDET ET SCIENTIA
THE UNIVERSITY OF CALIFORNIA BERKELEY 1868
Western UNIVERSITY · CANADA
ARIZONA STATE UNIVERSITY 1885
MŰEGYETEM 1782
ILLINOIS STATE UNIVERSITY 1857
THE CITY UNIVERSITY OF NEW YORK
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
TOKYO METROPOLITAN UNIVERSITY 首都大学東京
Ucla
AGH

“ *A main goal in designing a language should be to plan for growth. The language must start small, and the language must grow as the set of users grows.*

Guy Steele, “Growing a language”, 1998