Title

What Happens to a Dream Deferred? Chasing Language-Based Parallel Programming for HPC and AI

Author
Rouson, Damian

Publication Date
2025-03-05

DOI
10.25344/S47S36

BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

What Happens to a Dream Deferred?
Chasing Language-Based Parallel Programming for HPC and AI

Damian Rouson
Computer Languages and Systems Software (CLaSS) Group

SIAM Conference on Computational Science and Engineering, 5 March 2025

Overview

From Software Archaeology to Software Modernity

01
A Walking Tour

02
The Dream

03
HPC

04
AI

05
Background Matters

06
Conclusions

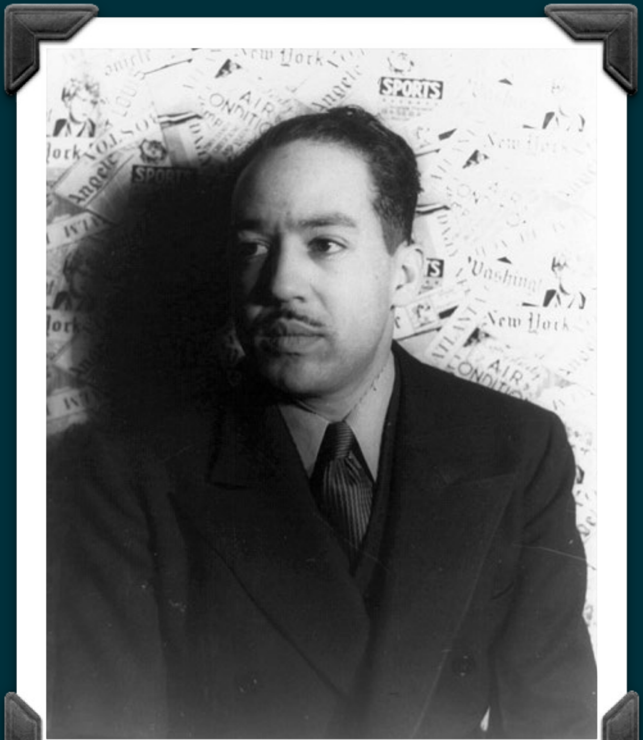
“Harlem”

By Langston Hughes, 1951

What happens to a dream deferred?
Does it dry up
like a raisin in the sun?
Or fester like a sore—
And then run?
Does it stink like rotten meat?
Or crust and sugar over—
like a syrupy sweet?

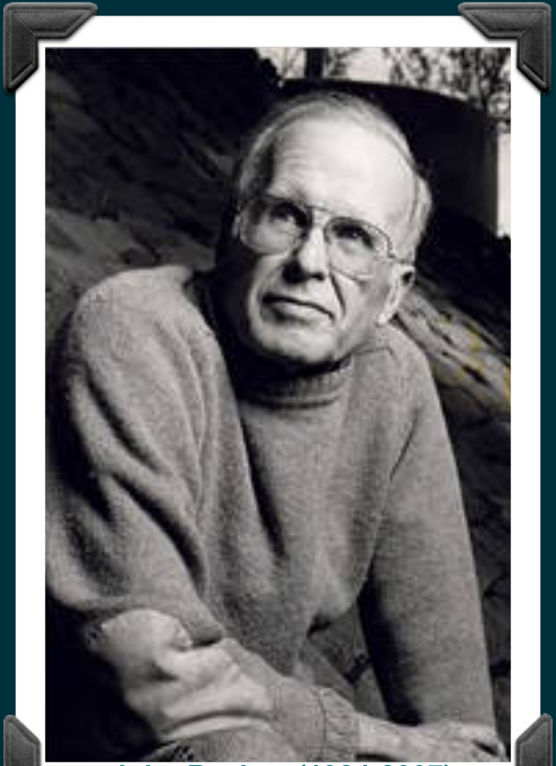
Maybe it just sags
like a heavy load.

Or does it explode?



Langston Hughes (1901-1967)

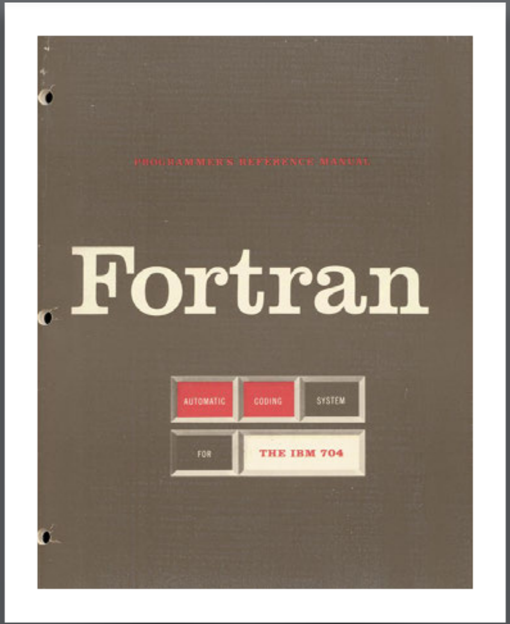
Portrait by Carl Van Vechten, 1936. Public Domain.
Library of Congress Prints and Photographs Division Washington, D.C. 20540
http://hdl.loc.gov/loc.pnp/cph.3b38891



John Backus (1924-2007)

Pioneers in Science and Technology Series: John Backus, 1984
© City of Oak Ridge, Oak Ridge, TN 3783 (Public Domain)
https://cdm16107.contentdm.oclc.org/digital/collection/p15388coll1/id/526

1956



The Fortran Automatic Coding System fort he IBM 704,
the first programmer’s reference manual for Fortran
(Public Domain)

https://cdm16107.contentdm.oclc.org/digital/collection/p15388coll1/id/526

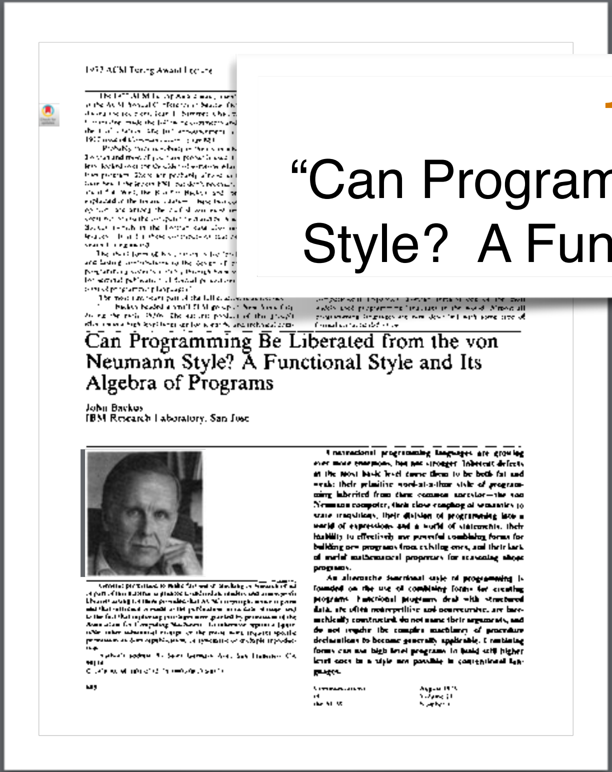
1961

"Fortran is a new and exciting language used by programmers to communicate with computers. It is exciting as it is the wave of the future."

Character of Dorothy Vaughan,
a NASA mathematician and programmer,
as played by Octavia Spencer in
Hidden Figures (20th Century Fox, 2016).

BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science


Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose

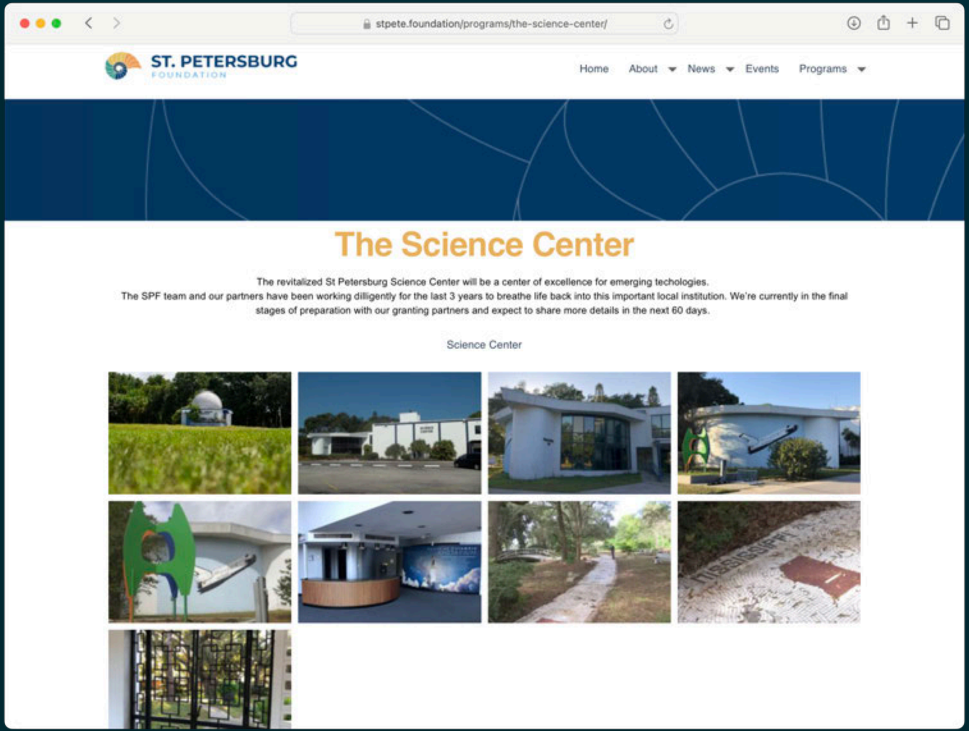
Backus, J., *Communications of the ACM*, August 1978, 21:8

1977 Turing Award Lecture:

"Can Programming be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs"

BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science


stpete.foundation/programs/the-science-center/

ST. PETERSBURG FOUNDATION

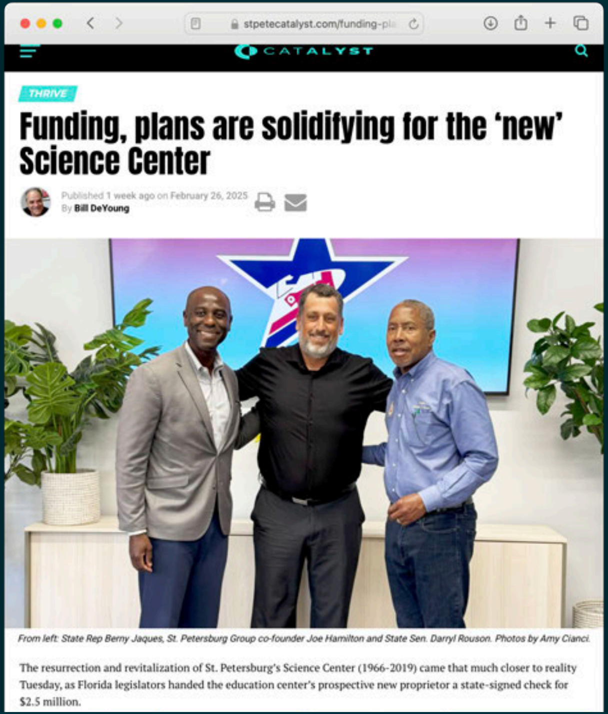
Home About News Events Programs

The Science Center

The revitalized St Petersburg Science Center will be a center of excellence for emerging techologies.
The SPF team and our partners have been working diligently for the last 3 years to breathe life back into this important local institution. We're currently in the final stages of preparation with our granting partners and expect to share more details in the next 60 days.

Science Center

1979


stpetecatalyst.com/funding-pla

CATALYST

THRIVE

Funding, plans are solidifying for the 'new' Science Center

Published 1 week ago on February 26, 2025
By Bill DeYoung

From left: State Rep Berny Jaques, St. Petersburg Group co-founder Joe Hamilton and State Sen. Darryl Rouson. Photos by Amy Cianci.

The resurrection and revitalization of St. Petersburg's Science Center (1966-2019) came that much closer to reality Tuesday, as Florida legislators handed the education center's prospective new proprietor a state-signed check for $2.5 million.

2025

Berkeley Lab
Bringing Science Solutions to the World

U.S. Department of Energy
Office of Science

Rumors of Fortran's Demise…

Retire Fortran? A Debate Rekindled

David Cann
Computing Research Group, L-306
Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA 94550
cann@lll-crg.llnl.gov

Abstract

In the May 1984 issue of Physics Today, Jim McGraw debated David Kuck and Michael Wolfe on the question of retiring FORTRAN. They addressed such questions as: Is FORTRAN the best tool for decomposing

cause of today's software crisis. We bel
Graw in 1984, that increased producti
utility, portability, and performance
ble if programmers avoid the constra
tive languages and adopt a higher leve
We must escape the morass of imper

July 24, 1991

Lawrence Livermore National Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

2 Programming Alternatives

In 1984, McGraw noted that by all indications future supercomputers would be multiprocessors. Today, most supercomputer users and vendors agree. But can programmers take advantage of the horse-

model to the imperative model of FORTRAN. To begin, we list the desired characteristics of a true parallel programming language [1]:

1. The language must insulate the programmer from the underlying machine. Deriving and expressing a parallel algorithm is hard enough; one should not have to reprogram it for each new machine.

2. Parallelism must be implicit in the semantics of the language. The compilation system should not have to unravel the behavior of the computation.

3. When a programmer desires determinancy, the language should guarantee it. Regardless of the conditions of execution, a program that realizes a determinate algorithm should yield the same results for the same data.

Of the three items, the last is an issue only when automatic parallelizing compilers are not available and the programmer is responsible for expressing and managing parallelism. Programmers will make mistakes, and these mistakes may remain hidden until system activity changes the rate of execution. This is all we will say about determinancy, as most parallel machines support automatic parallelizing compilers.

Regarding the first two items, however, imperative languages fail to meet the requirements. Remember that languages like FORTRAN were designed to exploit von Neumann machines. As such their computational model assumes that a single program counter will step

For example, consider the following FORTRAN excerpt:

```
A = Foo(X)
B = Goo(Y)
```

Determining if these statements can execute in parallel requires a full understanding of both functions. Because of COMMON blocks, they might share data. Further, because of aliasing, some combination of X, Y, A, or B might represent the same memory cell. Hence the parallelism in this excerpt is not immediately obvious, and its discovery requires interprocedural analysis or function expansion.

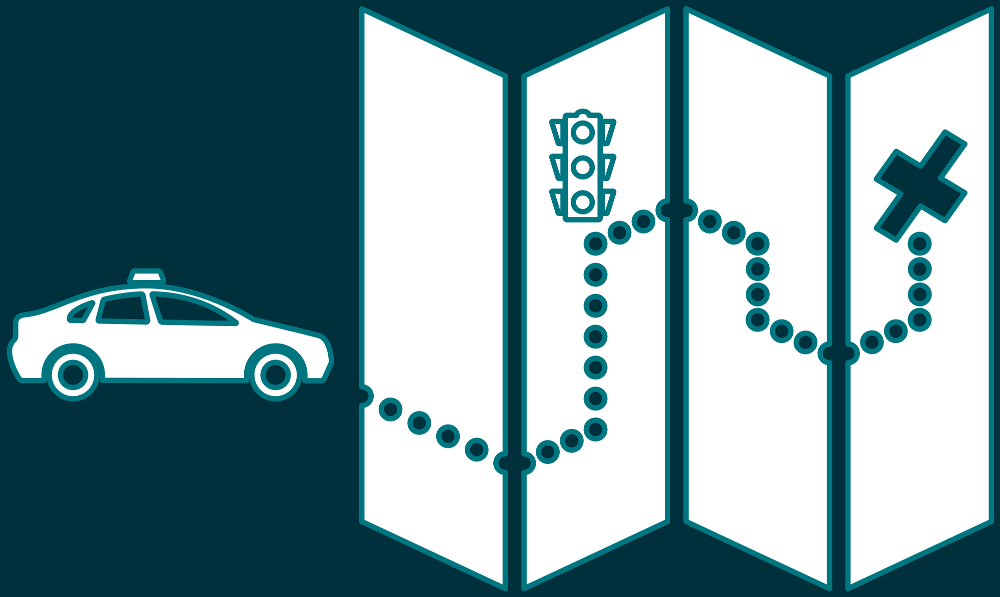
Functional languages, on the other hand, meet all the requirements listed above and do not require analysis for the discovery of parallelism [1,11,13,14]. A functional program is a collection of mathematically sound expressions comprised of both intrinsic and user defined functions. These functions are *well defined* and *determinate*. That is, they define a unique mapping between their domain and their range. A function passed the same set of values will yield the same results regardless of the environment of invocation. This establishes *referential transparency*, which implies that the evaluation of an expression, or the sharing of its subexpressions, does not change the value it denotes. Consequently, expressions are *side effect free*. The concept of a FORTRAN COMMON block does not exist. In the absence of side effects, programmers cannot see the target machine; the concept of *data* replaces memory, and the concept of *creation* replaces update. Further, in the absence of side effects, programs are implicitly parallel.

1984 – 1991

BERKELEY LAB
Bringing Science Solutions to the World

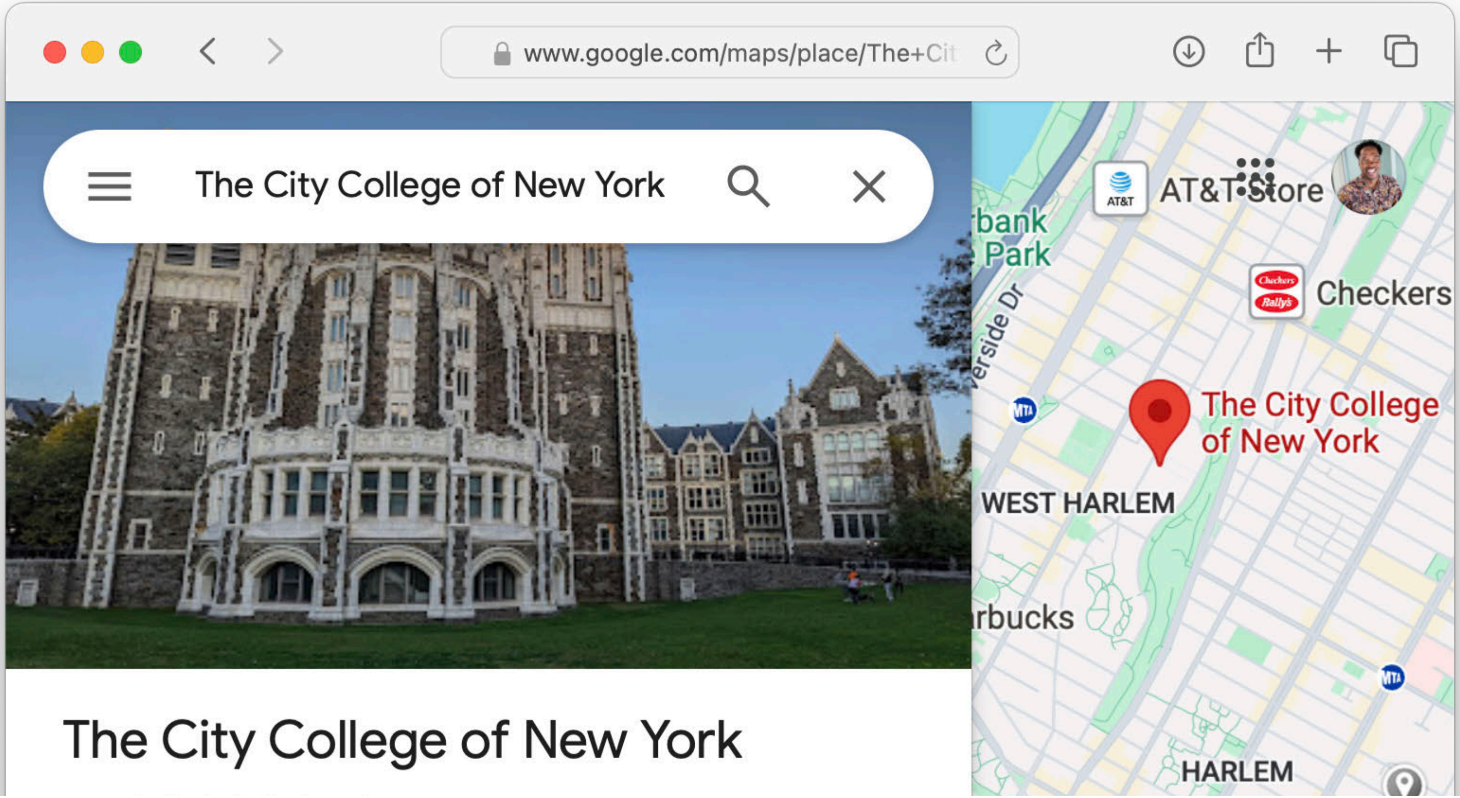
U.S. DEPARTMENT OF ENERGY
Office of Science

Or a Roadmap for Fortran’s Future?



Overview

From Software Archaeology to Software Modernity


www.google.com/maps/place/The+Cit
The City College of New York
bank
Park
erside Dr
AT&T
AT&T Store
Checkers
The City College
of New York
WEST HARLEM
rbucks
HARLEM
The City College of New York

1999

The Dream

To Make Software Representations of Partial Differential Equations More Closely Resemble Their Textbook Counterparts…

by applying differential operators to continuous mathematical abstractions supported by discrete approximations executing in parallel.

Overview

From Software Archaeology to Software Modernity

01
A Walking Tour

02
The Dream

03
HPC

Δ

04
AI

05
Background Matters

06
Conclusions

Abstract Calculus Pattern

```
u_t = -(.grad.p)/rho              &
          + nu * (.laplacian.u)   &
          -(u .dot.(.grad.u))
```

Abstract Calculus Pattern

```
u_t = -(.grad.p)/rho              &
      + nu * (.laplacian.u)       &
      -(u .dot. (.grad.u))
```

$$\vec{u}_t = -\frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} - \vec{u}\cdot\nabla\vec{u}$$

Pure user-defined operators

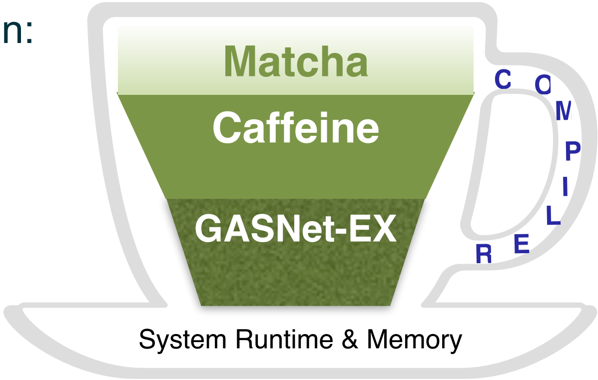
Distributed objects

Matcha: Motility Analysis of T-Cell histories in Activation

A parallel virtual T-cell model.

- Matcha tracks the stochastic T-cell motions according to multiple distributions of speeds and angles, accounting for the dependence of speed on the turning angle and on the previous speed.
- T cells must mount a coordinated attack in order to avoid overwhelming the host tissue.
- The study of T-cell/T-cell interactions remains in its infancy [1].
- Some communication occurs via secreting soluble mediators, e.g., cytokines and chemokines.
- Matcha models mediator spread via a 3D diffusion equation:

$$\phi_t = D\nabla^2\phi \quad \text{where} \quad \phi_t = \partial\phi/\partial t$$



https://go.lbl.gov/matcha

Collaborator:
Prof. David Torres
Northern New Mexico College
via Sustainable Research Pathways

[1] L.F. Uhl and A. Ge´rard A. "Modes of communication between T cells and relevance for immune responses." *Int. J. Mol. Sci.* **2020**, *21*, 2674; doi:10.3390/ijms21082674

shinstitute.org/sustainable-research-pathways

Sustainable Research Pathways – Sustainable Horizons Institute

SUSTAINABLE HORIZONS INSTITUTE

Home | About | Our Work | Testimonies | Get Involved | Make a Donation

Sustainable Research Pathways

Sustainable Research Pathways (SRP) is a comprehensive workforce development program designed to connect students and faculty working with a broad group of students who might not otherwise be recruited by the U.S.Department of Energy (DOE) National Laboratories with lab scientists to encourage lasting collaborations, jump start careers, and build vibrant workplace environments.

We customize our SRP programs for each team and each institution. SHI handles all the recruiting, workshop applications, speaker invitations, workshop facilitation, summer internship applications, and post-workshop data collection. If you are interested in bringing SRP to your team or your institution, please contact us at SRP@shinstitute.org.

Oak Ridge National Lab Senior Computer Scientist staff William Godoy (left) Grand Valley State University Professor Christian Trefftz (center) and Grand Valley State University student Elise Dettling (right)

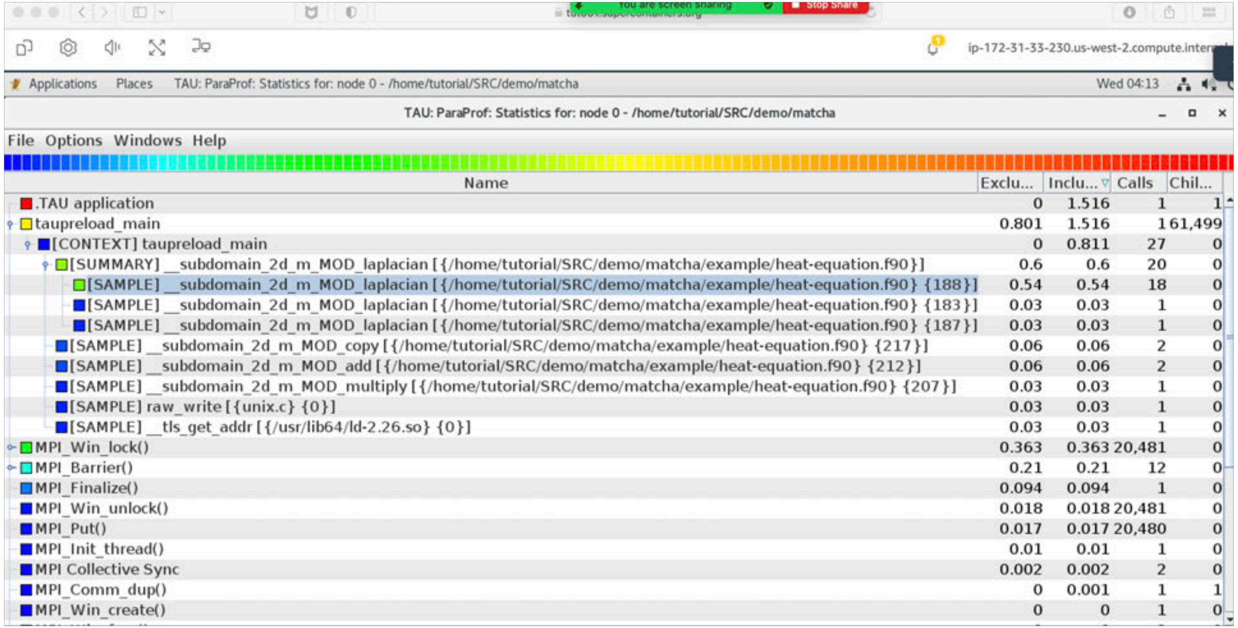
```fortran
function functional_programming_time()
! ...
  type(subdomain_t) T
! ...
  associate(dt => T%dt_stable(alpha))
    functional_programming: &
    do step = 1, steps
      T =  T + dt * alpha * .laplacian. T
    end do functional_programming
  end associate
! ...


function procedural_programming_time()
! ...
  type(subdomain_t) T
! ...
  associate(dt => T%dt_stable(alpha))
    procedural_programming: &
    do step = 1, steps
      call T%step(alpha*dt)
    end do procedural_programming
  end associate
```

```
git clone git@github.com:berkeleylab/matcha
cd matcha
fpm run \
 --example time-paradigm \
 --compiler caf \
 --runner “cafrun -n 2”

...

 Functional program time:    1.55110109
 Procedural program time:    1.54629397
```

Demonstration with gfortran 14.2.0 & OpenCoarrays 2.10.2

Loop-Level Parallelism

BERKELEY LAB
Bringing Science Solutions to the World



Name	Exclu...	Inclu...	Calls	Chil...
.TAU application	0	1.516	1	1
taupreload_main	0.801	1.516	1	61,499
[CONTEXT] taupreload_main	0	0.811	27	0
[SUMMARY] __subdomain_2d_m_MOD_laplacian [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90}]	0.6	0.6	20	0
[SAMPLE] __subdomain_2d_m_MOD_laplacian [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90} {188}]	0.54	0.54	18	0
[SAMPLE] __subdomain_2d_m_MOD_laplacian [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90} {183}]	0.03	0.03	1	0
[SAMPLE] __subdomain_2d_m_MOD_laplacian [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90} {187}]	0.03	0.03	1	0
[SAMPLE] __subdomain_2d_m_MOD_copy [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90} {217}]	0.06	0.06	2	0
[SAMPLE] __subdomain_2d_m_MOD_add [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90} {212}]	0.06	0.06	2	0
[SAMPLE] __subdomain_2d_m_MOD_multiply [{/home/tutorial/SRC/demo/matcha/example/heat-equation.f90} {207}]	0.03	0.03	1	0
[SAMPLE] raw_write [{unix.c} {0}]	0.03	0.03	1	0
[SAMPLE] __tls_get_addr [{/usr/lib64/ld-2.26.so} {0}]	0.03	0.03	1	0
MPI_Win_lock()	0.363	0.363	20,481	0
MPI_Barrier()	0.21	0.21	12	0
MPI_Finalize()	0.094	0.094	1	0
MPI_Win_unlock()	0.018	0.018	20,481	0
MPI_Put()	0.017	0.017	20,480	0
MPI_Init_thread()	0.01	0.01	1	0
MPI Collective Sync	0.002	0.002	2	0
MPI_Comm_dup()	0	0.001	1	1
MPI_Win_create()	0	0	1	0

line continuation

```
188 do concurrent(j=2:ny-1)
189   laplacian_rhs%s_(i, j) = &
      (halo_left(j)   - 2*rhs%s_(i, j) + rhs%s_(i+1,j  ))/dx_**2 + &
190   (rhs%s_(i, j-1) - 2*rhs%s_(i, j) + rhs%s_(i  ,j+1))/dy_**2
191 end do
```


18

The World's Shortest Fortran Program

```
end
```

The World’s Shortest Bug Reproducer

```
end
```

Overview

From Software Archaeology to Software Modernity

01
A Walking Tour

02
The Dream

03
HPC

Δ

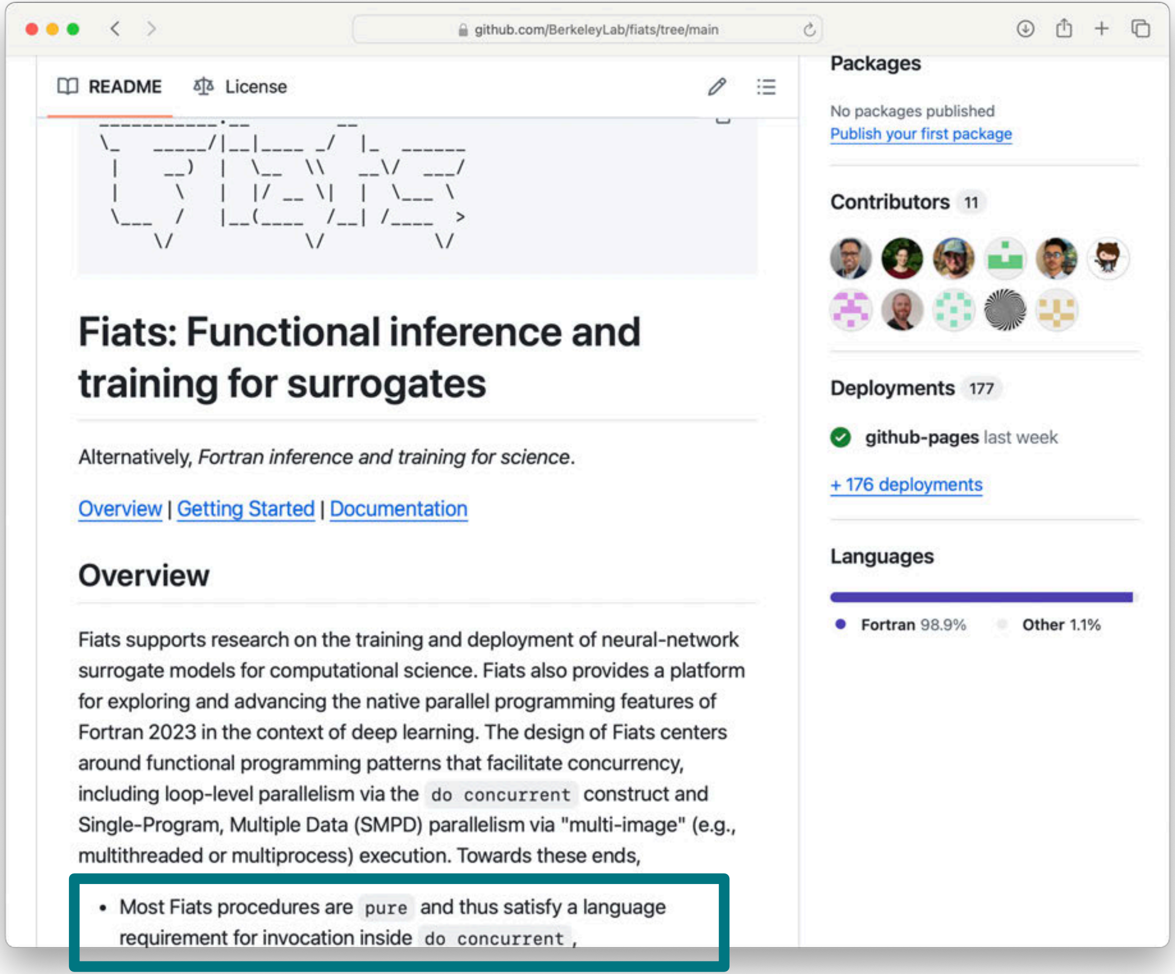
04
AI

ϵ I

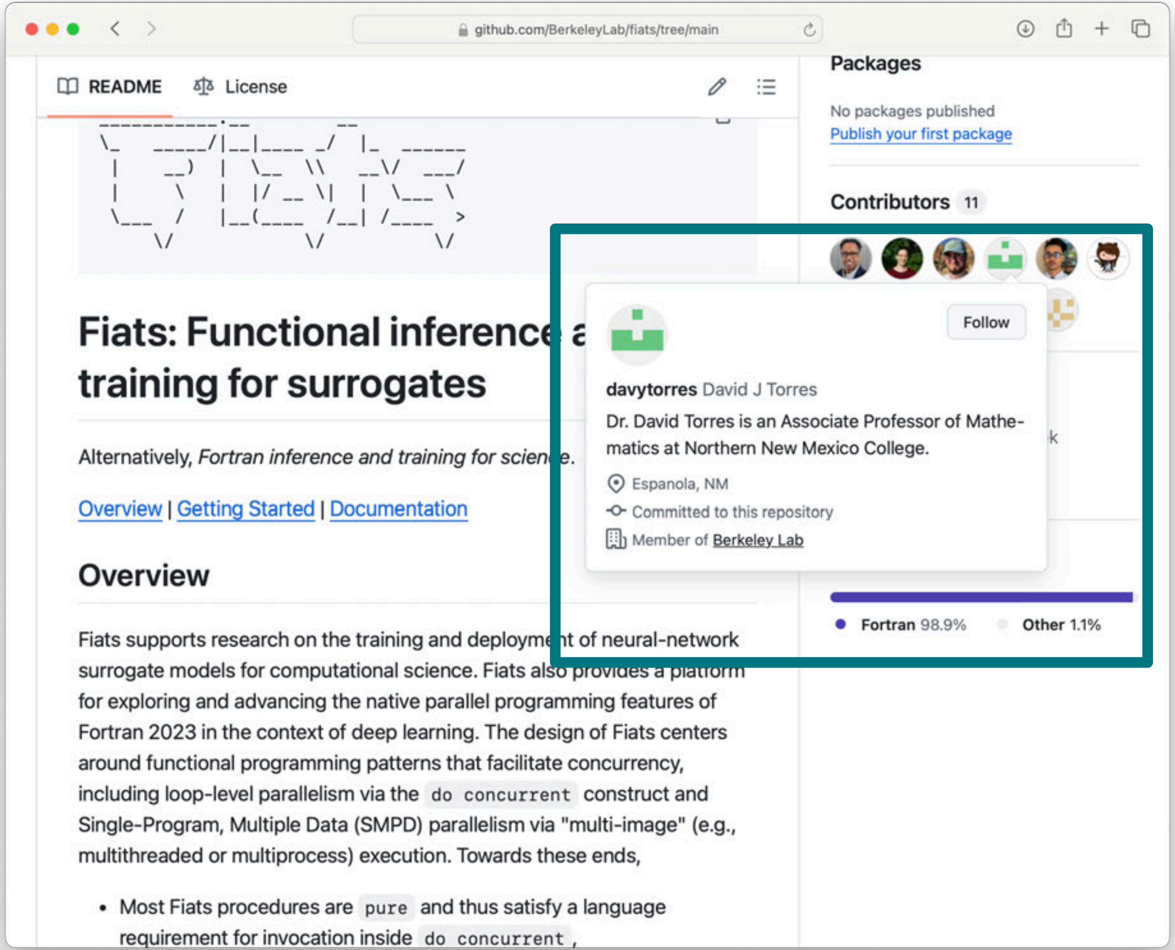
05
Background Matters

06
Conclusions

Fiats


github.com/BerkeleyLab/fiats/tree/main
README
License
Fiats: Functional inference and training for surrogates
Alternatively, *Fortran inference and training for science*.
Overview | Getting Started | Documentation
Overview
Fiats supports research on the training and deployment of neural-network surrogate models for computational science. Fiats also provides a platform for exploring and advancing the native parallel programming features of Fortran 2023 in the context of deep learning. The design of Fiats centers around functional programming patterns that facilitate concurrency, including loop-level parallelism via the `do concurrent` construct and Single-Program, Multiple Data (SMPD) parallelism via "multi-image" (e.g., multithreaded or multiprocess) execution. Towards these ends,
• Most Fiats procedures are `pure` and thus satisfy a language requirement for invocation inside `do concurrent`,
Packages
No packages published
Publish your first package
Contributors 11
Deployments 177
github-pages last week
+ 176 deployments
Languages
Fortran 98.9%
Other 1.1%

Fiats


github.com/BerkeleyLab/fiats/tree/main
README
License
Fiats: Functional inference and training for surrogates
Alternatively, Fortran inference and training for science.
Overview | Getting Started | Documentation
Overview
Fiats supports research on the training and deployment of neural-network surrogate models for computational science. Fiats also provides a platform for exploring and advancing the native parallel programming features of Fortran 2023 in the context of deep learning. The design of Fiats centers around functional programming patterns that facilitate concurrency, including loop-level parallelism via the do concurrent construct and Single-Program, Multiple Data (SMPD) parallelism via "multi-image" (e.g., multithreaded or multiprocess) execution. Towards these ends,
Most Fiats procedures are pure and thus satisfy a language requirement for invocation inside do concurrent,
Packages
No packages published
Publish your first package
Contributors 11
Follow
davytorres David J Torres
Dr. David Torres is an Associate Professor of Mathematics at Northern New Mexico College.
Espanola, NM
Committed to this repository
Member of Berkeley Lab
Fortran 98.9%
Other 1.1%

Fiats: Inference

example — vim concurrent-inferences.f90 — 90×9

```fortran
  !$omp parallel do default(none) shared(neural_network,inputs,outputs) collapse(3)
  do j=1,lon
    do k=1,lev
      do i=1,lat
        outputs(i,k,j) = neural_network%infer(inputs(i,k,j))
      end do
    end do
  end do
```

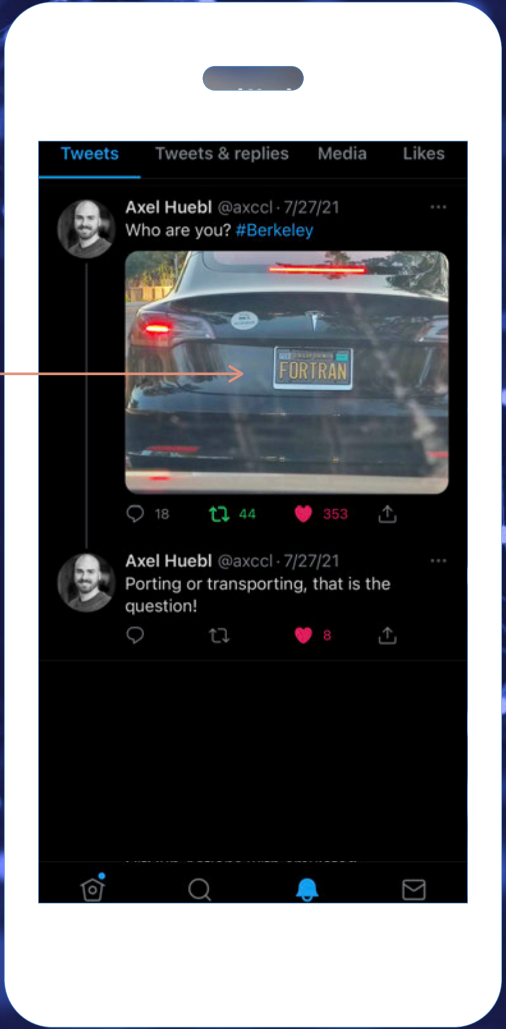
example — vim concurrent-inferences.f90 — 65×5

```fortran
  do concurrent(i=1:lat, k=1:lev, j=1:lon)
    outputs(i,k,j) = neural_network%infer(inputs(i,k,j))
  end do

```

example — vim concurrent-inferences.f90 — 50×5

```fortran
  !$omp workshare
  outputs = neural_network%infer(inputs)
  !$omp end workshare

```

Fortran at the Intersection



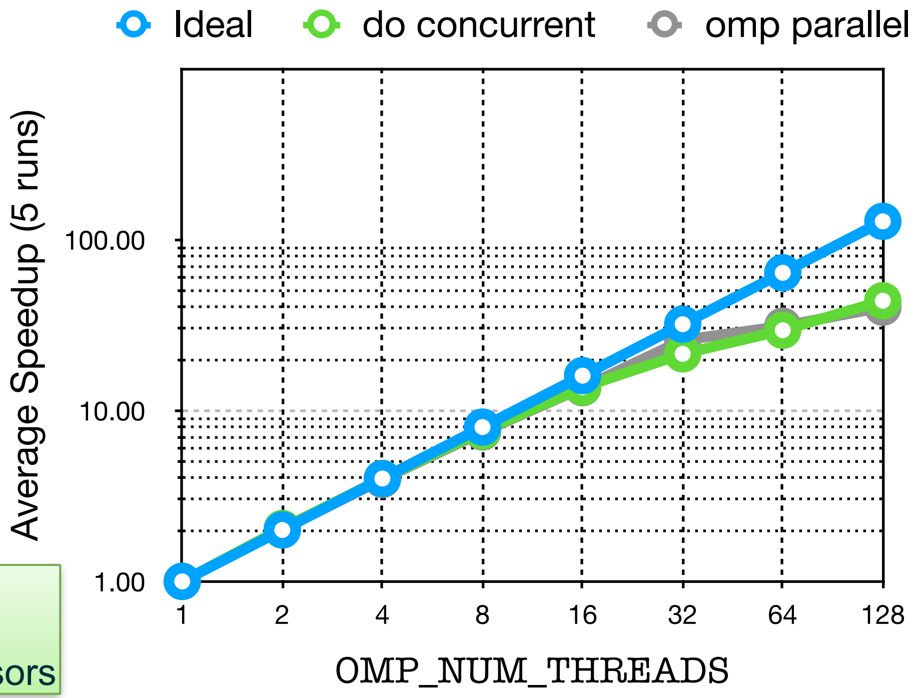
CPU Parallelism on Perlmutter

Compiler:
Berkeley Lab [llvm-project fork]
git tag paw-atm24-fiats
Commits pulled from ROCm fork

Neural network:
- Activation function: GELU
- Numbers of inputs: 80
- Number of outputs: 31
- Nodes per hidden layer: 256, 384, 256

Source: Z. Bai

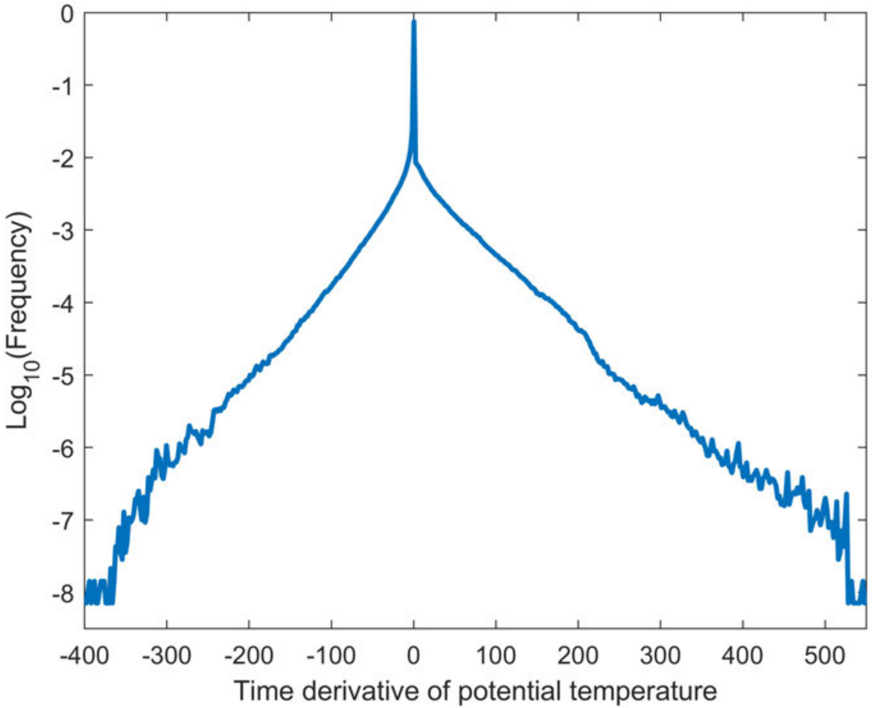
Platform:
Dedicated interactive node
2x 64-core AMD EPYC 7663 processors

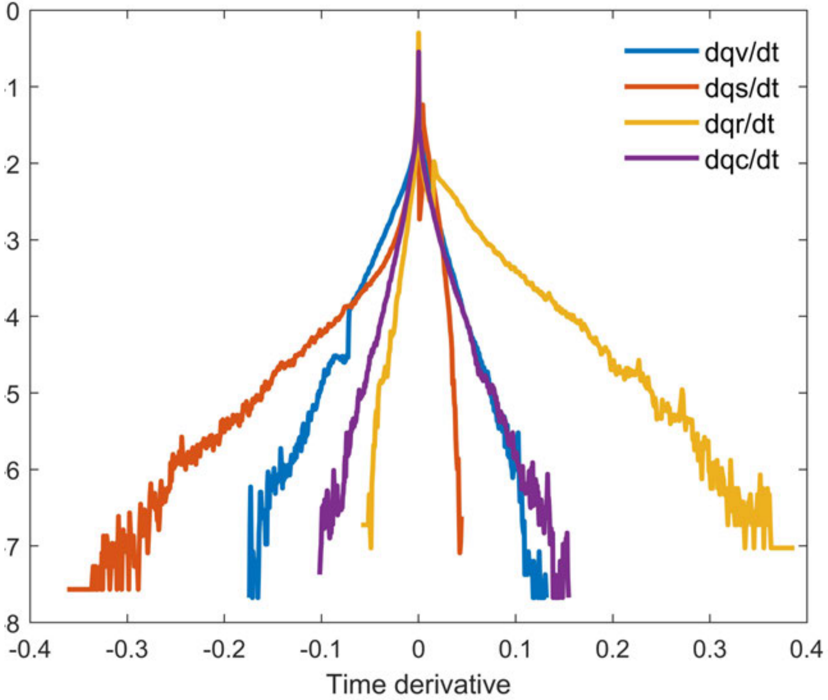


Example Command:

```
OMP_NUM_THREADS = 128  fpm run --example concurrent-inferences \
  --runner "srun --cpu_bind=cores -c 128 -n 1" -- --network model.json
```

Training: Flattening the Histograms





Flattening as Maximum-Entropy Sampling

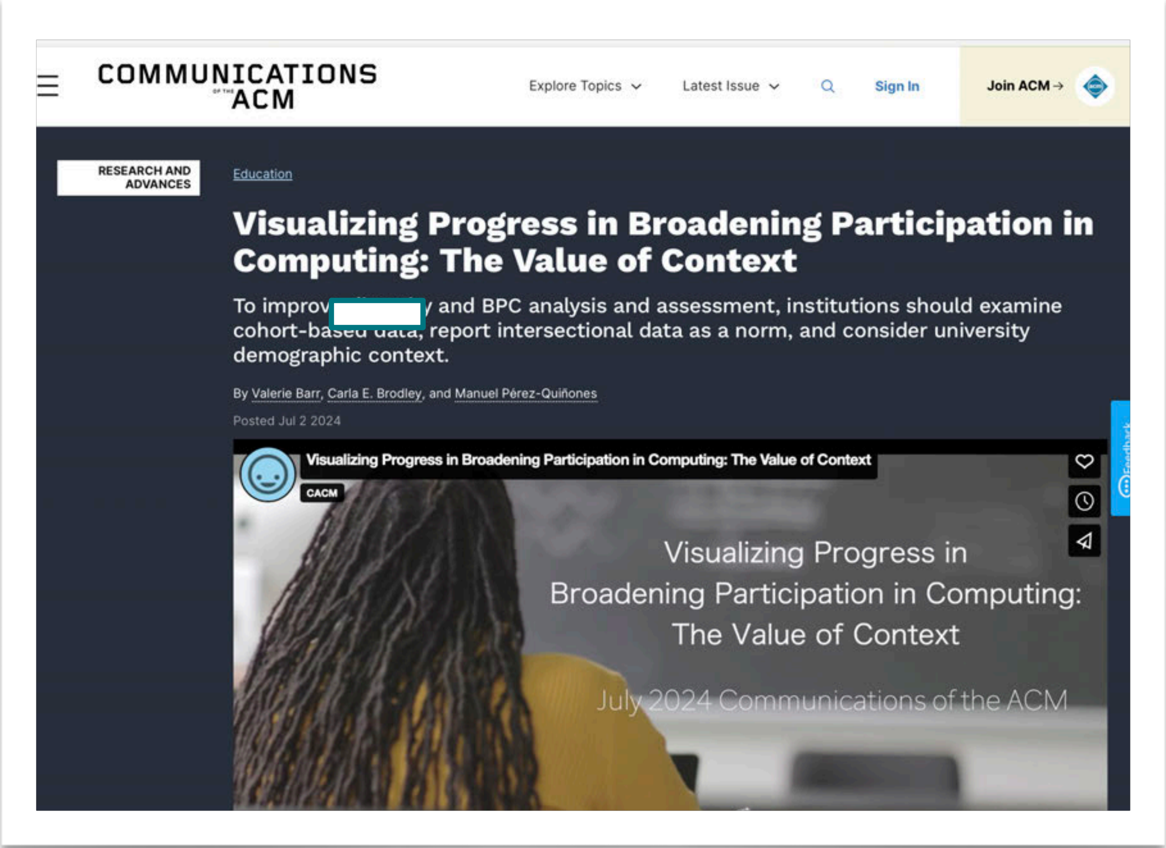
Information Entropy

$$I = -\sum_{i=1}^{N} p(x_i) \log p(x_i)$$

Maximal condition:

$$p_i = 1/N$$

$$\Rightarrow I = \log N$$



Overview

From Software Archaeology to Software Modernity

01
A Walking Tour

02
The Dream

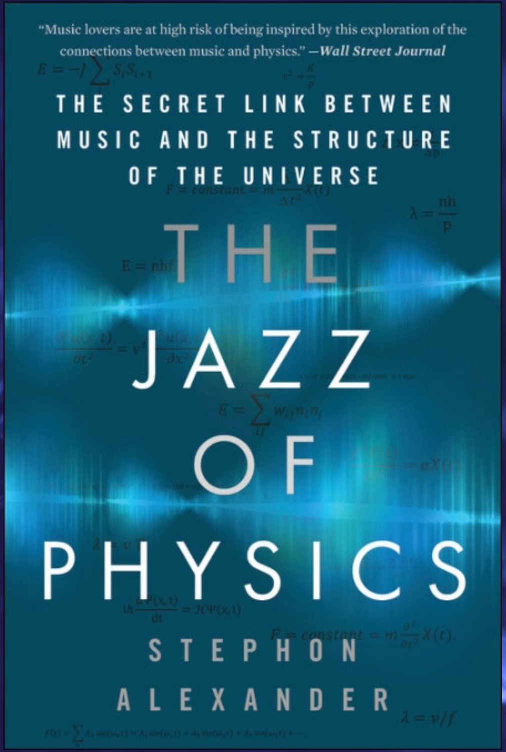
03
HPC

04
AI

$\epsilon \quad I$

05
Background Matters

06
Conclusions



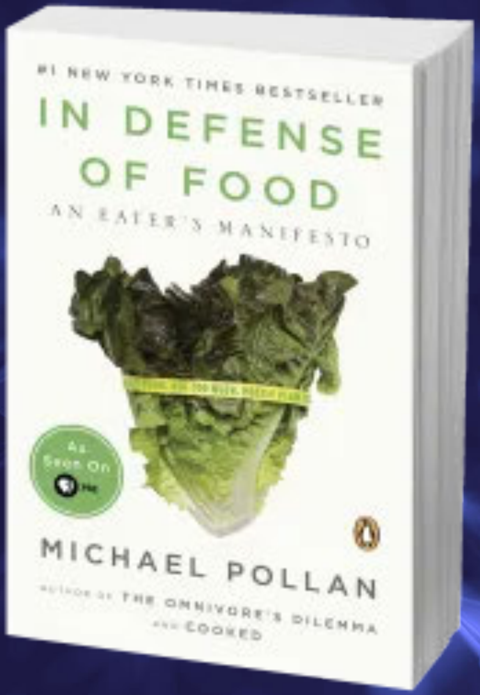
Background Matters

“… Then, out of the blue, Salam says to Jim: ‘One day, when your people do physics, it will be like jazz.’

What a great compliment, an affirmation and acknowledgment of the improvisational, inclusive, cultural, and intellectual contributions of this music called jazz.”

—Stephon Alexander (*The Jazz of Physics*, p. 231) quoting physics Nobel laureate Abdus Salam as relayed by physicist Jim Gates.

“It occurred to me by intuition, and music was the driving force behind that intuition. My discovery was the result of musical perception.”
— Albert Einstein (when asked about his theory of relativity)

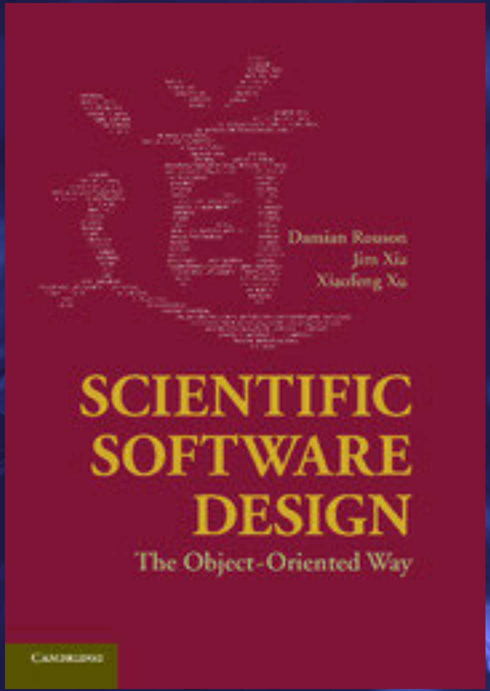


In Defense of Food: An Eater's Manifesto

Eat food.

Not too much.

Mostly plants.



In Defense of Software: A Developer’s Manifesto

Write software.

Not too much.

Mostly pure functions.

The [redacted] Innovation Paradox in Science

The [redacted]-Innovation Paradox in Science

Bas Hofstra[a,1], Vivek V. Kulkarni[b], Sebastian Munoz-Najar Galvez[a], Bryan He[b], Dan Jurafsky[b,c], and Daniel A. McFarland[a,1]

[a]Graduate School of Education, Stanford University, Stanford, CA 94305; [b]Department of Computer Science, Stanford University, Stanford, CA 94305; and [c]Department of Linguistics, Stanford University, Stanford, CA 94305

Edited by Peter S. Bearman, Columbia University, New York, NY, and approved March 16, 2020 (received for review September 5, 2019)

Prior work finds a diversity paradox: Diversity breeds innovation, yet underrepresented groups that diversify organizations have less successful careers within them. Does the diversity paradox hold for scientists as well? [illegible] a near-complete population of ~1.2 [illegible] 1977 to 2015 and following [illegible] into publishing [illegible] positions. We [illegible] that [illegible] diversity's role in innovation and par[illegible]ains the underr[illegible]tation of some groups in academia.

diversity | inn[illegible] | inequality | [illegible]

Innovation drives scie[illegible]nnovation prop[illegible] into uncharted territories and expands humanity's unde[illegible]ing of the natural and social world. Innovation is also believe[illegible] be predictive of successful scientific careers: Innovators are sc[illegible]ence's trailblazers and discoverers, so producing innovative science may lead to successful academic careers (1). At the same time, a common hypothesis is that demographic diversity brings such innovation (2–5). Scholars from underrepresented groups have origins, concerns, and experiences that differ from groups traditionally represented, and their inclusion in academe diversifies scholarly perspectives. In fact, historically underrepresented groups often draw relations between ideas and concepts that have been traditionally missed or ignored (4–7). Given this, if demographic groups are unequally represented in academia, then one would expect underrepresented groups to generate more scientific innovation than overrepresented groups and have more successful careers (*SI Appendix*). Unfortunately, the combination of these two relationships—diversity–innovation and innovation–careers—fails to result and poses a paradox. If gender and racially underrepresented scholars are likely to innovate and innovation supposedly leads to successful academic careers, then how do we explain persistent inequalities in scientific careers between minority and majority groups (8–13)? One explanation is that the scientific innovations produced by some groups are discounted, possibly leading to differences in scientific impact and successful careers.

In this paper, we set out to identify the diversity–innovation paradox in science and explain why it arises. We provide a system-level account of science using a near-complete population of US doctorate recipients (~1.2 million) where we identify scientific innovations (14–19) and analyze the rates at which different demographic groups relate scientific concepts in novel ways, the extent to which those novel conceptual relations get taken up by other scholars, how "distal" those linkages are (14), and the subsequent returns they have to scientific careers. Our analyses use observations spanning three decades, all scientific disciplines, and all US doctorate-awarding institutions. Through them we are able 1) to compare minority scholars' rates of scientific novelty vis-à-vis majority scholars and then ascertain whether and why their novel conceptualizations 2) are taken up by others and, in turn, 3) facilitate a successful research career.

Innovation as Novelty and Impactful Novelty in Text

Our dataset stems from ProQuest dissertations (20), which includes records of nearly all US PhD theses and their metadata from 1977 to 2015: student names, advisors, institutions, thesis titles, abstracts, disciplines, etc. These structural and semantic footprints enable us to consider students' rates of innovation at the very onset of their scholarly careers and their academic trajectory afterward, i.e., their earliest conceptual innovations and how they correspond to successful academic careers (21). We link these data with several data sources to arrive at a near-complete ecology of US PhD students and their career trajectories. Specifically, we link ProQuest dissertations to the US Census data (2000 and 2010) and Social Security Administration data (1900 to 2016) to infer demographic information on students' gender and race (i.e., name signals for white, Asian, or [illegible]errepresented minority [Hispanic, African American, or [illegible] American]; see *Materials and Methods* and *SI Appendix*); [illegible]roQuest dissertations to Web of Science, a large-scale [illegible]database with ~38 million academic publications [illegible] to find out which students have continued re[illegible], and we weigh our inferential analyses by populatio[illegible]ds of the number of PhD recipients for each distinct university–year combination to render results generalizable to the population (*SI Appendix*).

Significance

By analyzing data from nearly all US PhD recipients and their dissertations across three decades, this paper finds demographically underrepresented students innovate at higher rates than majority students, but their novel contributions are discounted and less likely to earn them academic positions. The discounting of minorities' innovations may partly explain their underrepresentation in influential positions of academia.

Author contributions: B. Hofstra, V.V.K., and D.A.M. designed research; B. Hofstra, V.V.K., S.M.-N.G., B. He, D.J., and D.A.M. performed research; B. Hofstra, V.V.K., S.M.-N.G., B. He, D.J., and D.A.M. contributed new reagents/analytic tools; B. Hofstra, V.V.K., S.M.-N.G., B. He, D.J., and D.A.M. analyzed data; and B. Hofstra and D.A.M. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

See online for related content such as Commentaries.

[1]To whom correspondence may be addressed. Email: bhofstra@stanford.edu or mcfarland@stanford.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.1915378117/-/DCSupplemental.

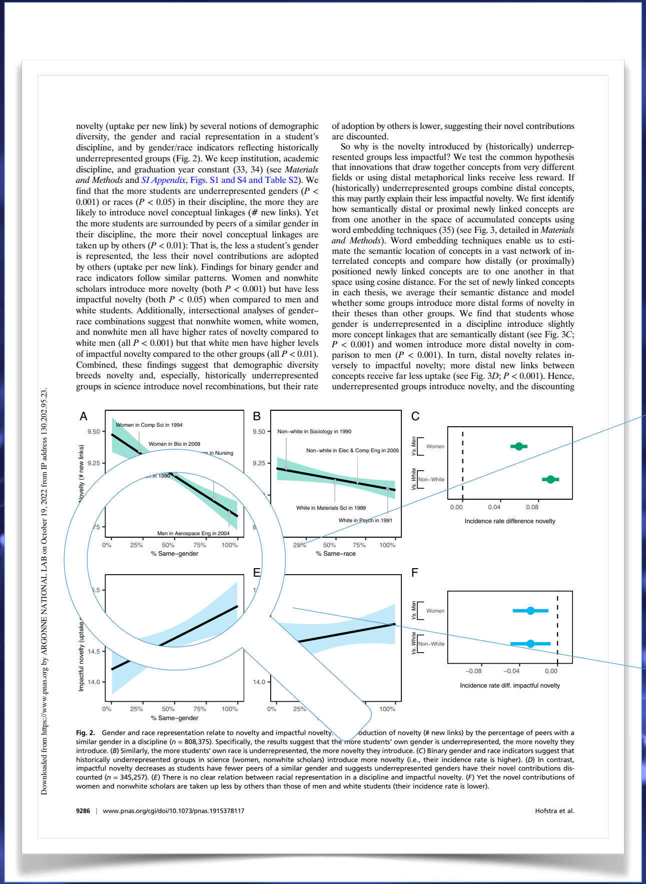
First published April 14, 2020.

www.pnas.org/cgi/doi/10.1073/pnas.1915378117

rewarded? Our analyses show that underrepresented groups produce higher rates of scientific novelty. However, their novel contributions are devalued and discounted: For example, novel contributions by gender and racial minorities are taken up by other scholars at lower rates than novel contributions by gender and racial majorities, and equally impactful contributions of gender and racial minorities are less likely to result in successful scientific careers than for majority groups. These results suggest there may

Hofstra, B., Kulkarni, V. V., Munoz-Najar Galvez, S., He, B., Jurafsky, D., & McFarland, D. A. (2020). The [redacted]-innovation paradox in science. *Proceedings of the National Academy of Sciences*, *117*(17), 9284-9291. Chicago

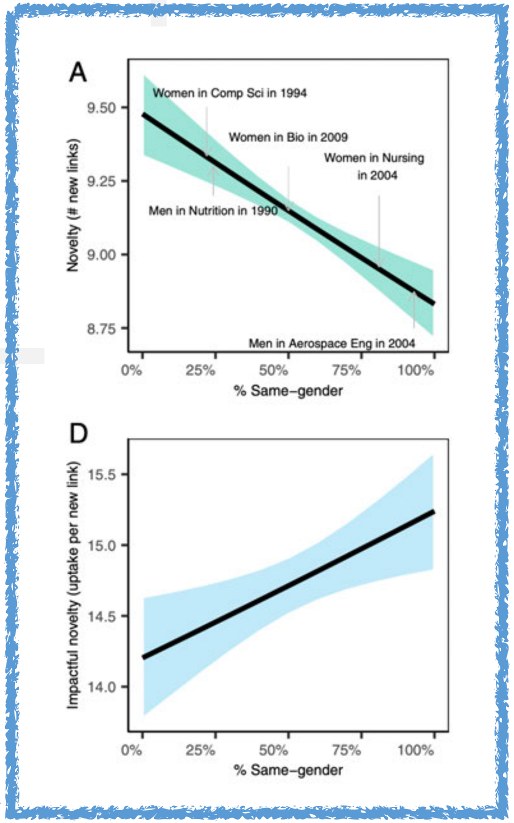
The [redacted] Innovation Paradox in Science



novelty (uptake per new link) by several notions of demographic diversity, the gender and racial representation in a student's discipline, and by gender/race indicators reflecting historically underrepresented groups (Fig. 2). We keep institution, academic discipline, and graduation year constant (33, 34) (see *Materials and Methods* and *SI Appendix*, Figs. S1 and S4 and Table S2). We find that the more students are underrepresented genders ($P < 0.001$) or races ($P < 0.05$) in their discipline, the more they are likely to introduce novel conceptual linkages (# new links). Yet the more students are surrounded by peers of a similar gender in their discipline, the more their novel conceptual linkages are taken up by others ($P < 0.01$): That is, the less a student's gender is represented, the less their novel contributions are adopted by others (uptake per new link). Findings for binary gender and race indicators follow similar patterns. Women and nonwhite scholars introduce more novelty (both $P < 0.001$) but have less impactful novelty (both $P < 0.05$) when compared to men and white students. Additionally, intersectional analyses of gender–race combinations suggest that nonwhite women, white women, and nonwhite men all have higher rates of novelty compared to white men (all $P < 0.001$) but that white men have higher levels of impactful novelty compared to the other groups (all $P < 0.01$). Combined, these findings suggest that demographic diversity breeds novelty and, especially, historically underrepresented groups in science introduce novel recombinations, but their rate of adoption by others is lower, suggesting their novel contributions are discounted.

So why is the novelty introduced by (historically) underrepresented groups less impactful? We test the common hypothesis that innovations that draw together concepts from very different fields or using distal metaphorical links receive less reward. If (historically) underrepresented groups combine distal concepts, this may partly explain their less impactful novelty. We first identify how semantically distal or proximal newly linked concepts are from one another in the space of accumulated concepts using word embedding techniques (35) (see Fig. 3, detailed in *Materials and Methods*). Word embedding techniques enable us to estimate the semantic location of concepts in a vast network of interrelated concepts and compare how distally (or proximally) positioned newly linked concepts are to one another in that space using cosine distance. For the set of newly linked concepts in each thesis, we average their semantic distance and model whether some groups introduce more distal forms of novelty in their theses than other groups. We find that students whose gender is underrepresented in a discipline introduce slightly more concept linkages that are semantically distant (see Fig. 3*C*; $P < 0.001$) and women introduce more distal novelty in comparison to men ($P < 0.001$). In turn, distal novelty relates inversely to impactful novelty; more distal new links between concepts receive far less uptake (see Fig. 3*D*; $P < 0.001$). Hence, underrepresented groups introduce novelty, and the discounting

Fig. 2. Gender and race representation relate to novelty and impactful novelty. … oduction of novelty (# new links) by the percentage of peers with a similar gender in a discipline ($n = 808{,}375$). Specifically, the results suggest that the more students' own gender is underrepresented, the more novelty they introduce. (*B*) Similarly, the more students' own race is underrepresented, the more novelty they introduce. (*C*) Binary gender and race indicators suggest that historically underrepresented groups in science (women, nonwhite scholars) introduce more novelty (i.e., their incidence rate is higher). (*D*) In contrast, impactful novelty decreases as students have fewer peers of a similar gender and suggests underrepresented genders have their novel contributions discounted ($n = 345{,}257$). (*E*) There is no clear relation between racial representation in a discipline and impactful novelty. (*F*) Yet the novel contributions of women and nonwhite scholars are taken up less by others than those of men and white students (their incidence rate is lower).



Hofstra, B., Kulkarni, V. V., Munoz-Najar Galvez, S., He, B., Jurafsky, D., & McFarland, D. A. (2020). The [redacted]-innovation paradox in science. *Proceedings of the National Academy of Sciences*, *117*(17), 9284-9291. Chicago

www.nytimes.com/202

The New York Times

PLAY THE CROSSWORD

Account

OPINION

PAMELA PAUL

A Paper That Says Science Should Be Impartial Was Rejected by Major Journals. You Can't Make This Up.

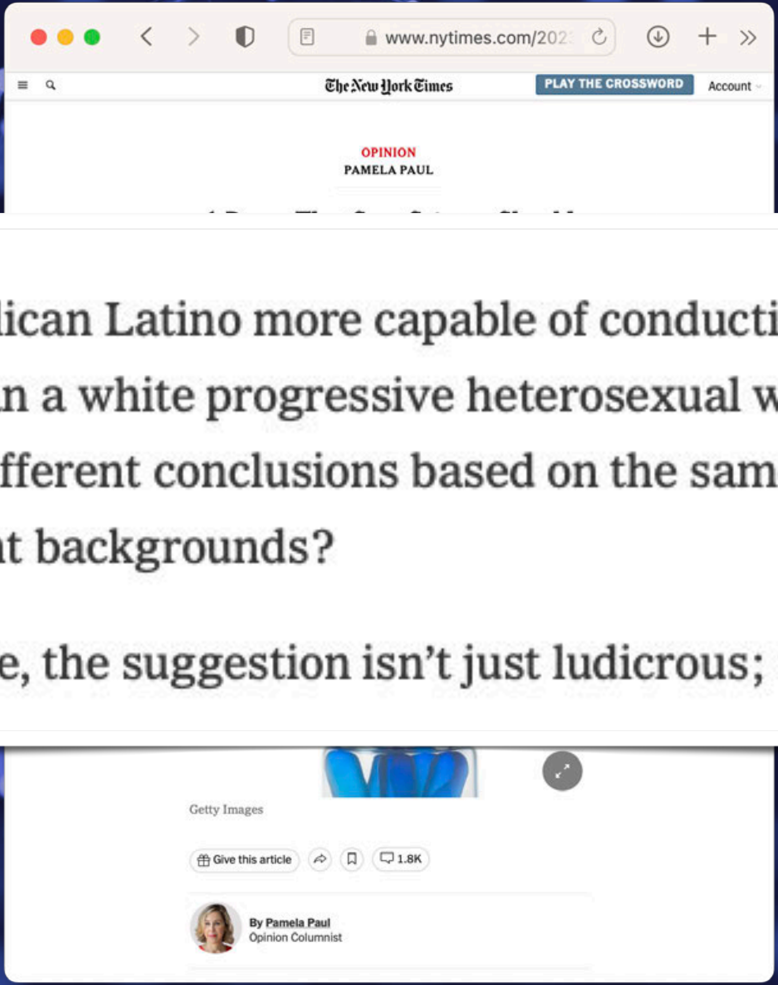
May 4, 2023

Getty Images

Give this article

1.8K

By Pamela Paul
Opinion Columnist



Is a gay Republican Latino more capable of conducting a physics experiment than a white progressive heterosexual woman? Would they come to different conclusions based on the same data because of their different backgrounds?

For most people, the suggestion isn't just ludicrous; it's offensive.

A Different Take

- Science answers questions that scientists ask.
- Asking different questions might yield very different answers — even sometimes seemingly contradictory ones.
- This seems especially true in the social sciences and the humanities and we're operating closer to the social sciences (cf. *Communications of the ACM*) and the humanities (cf. *The Physics of Jazz*) than most of us likely realize.
- The author's opening question suffers from at least two problems:

  1. The use of "more capable" implies a capability metric with which one can establish a mathematical inequality.
  2. Opening with "is" suggests a single deterministic answer.

Is a gay Republican Latino more capable of conducting a physics experiment than a white progressive heterosexual woman? Would they come to different conclusions based on the same data because of their different backgrounds?

For most people, the suggestion isn't just ludicrous; it's offensive.

A Different Question

- Let’s try reframing the question:

  “Might a gay Republican Latino have different capabilities that could influence how they would conduct a physics experiment or which physics experiments they might choose to conduct as compared to a white progressive heterosexual woman?”

- Here, “might” clarifies that the the answer is probabilistic: the traits given are insufficient to fully determine the answer but that doesn’t mean those traits have no influence on outcomes.
- Also, “different capabilities” allows for a much richer and more complex understanding of human capabilities than a single measure for which one can write mathematical inequalities.
- The answer to the rephrased question is a resounding “Yes!”

Programming Languages Are Human Languages: Bias Exists

www.cs.virginia.edu/~evans/cs655/

University of Virginia, Department of Computer Science
CS655: Programming Languages, Spring 2001

How do we tell truths that might hurt?

Edsger W.Dijkstra, 18 June 1975
from https://www.cs.utexas.edu/users/EWD/ewd04xx/EWD498.PDF

Sometimes we discover unpleasant truths. Whenever we do so, we are in difficulties: suppressing them is scientifically dishonest, so we must tell them, but telling them, however, will fire back on us. If the truths are sufficiently impalatable, our audience is psychically incapable of accepting them and we will be written off as totally unrealistic, hopelessly idealistic, dangerously revolutionary, foolishly gullible or what have you. (Besides that, telling such truths is a sure way of making oneself unpopular in many circles, and, as such, it is an act that, in general, is not without personal risks. Vide Galileo Galilei.....)

Computing Science seems to suffer severely from this conflict. On the whole, it remains silent and tries to escape this conflict by shifting its attention. (For instance: with respect to COBOL you can really do only one of two things: fi… the disease or pretend that it does not exist. Most Computer Science Departments have opted for the latter easy way out.) But, Brethern, I ask yo… is this honest? Is not our prolonged silence fretting away Computing Science's intellectual integrity? Are we decent by remaining silent? If not, how do we speak up?

To give you some idea of the scope of the problem I have listed a number of such truths. (Nearly all computing scientists I know well will agree without hesitation to nearly all of them. Yet we allow the world to behave as if we di… not know them....)

- Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians.
- The easiest machine applications are the technical/scientific computations.
- The tools we use have a profound (and devious!) influence on our …
- FORTRAN --"the infantile disorder"--, by now nearly 20 years old, is hopelessly inadequate for whatever computer application you have in mind today: it is now too clumsy, too risky, and too expensive to use.

1975-2001

blog.codinghorror.com/

CODING HORROR
programming and human factors

ENHANCED BY Google

22 Apr 2005

You Can Write FORTRAN in any Language

A recent user-submitted CodeProject article took an interesting perspective on the VB.NET/C# divide by proposing that the **culture of Visual Basic** is not conducive to professional software development:

> We've seen that the cultures of VB and C# are very different. And we've seen that this is no fault of the programmers that use them. Rather this is a product of the combination of factors that collectively could be called their upbringing -- business environment, target market, integrity and background of the original language developers, and a myriad other factors.

2005

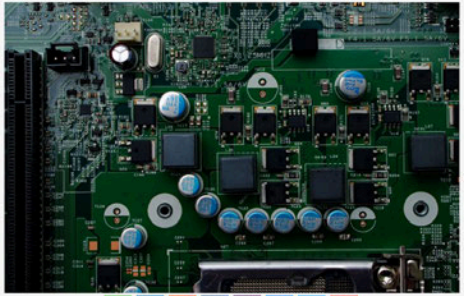
www.efinancialcareers.com/news/

efc COVID-19 Jobs & Resources | Search Jobs | News | Advice | Register | Sign In | Recruiters

Select Region

Real programmers write Java like FORTRAN

by Frank Becker 05 January 2021



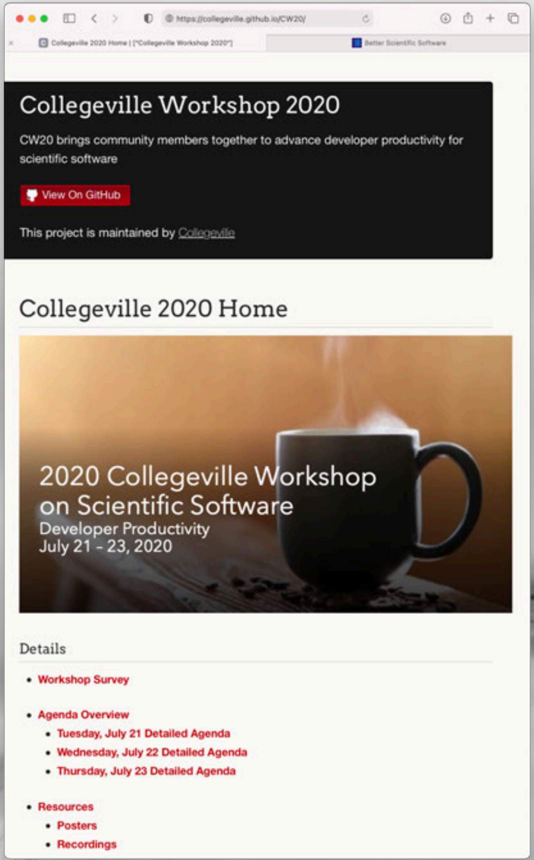
I'm a Java coder and I work on high speed trading systems. I know that people dispute whether Java is the best language for really low latency work, but my experience is that it is absolutely viable. I have seen optimized Java code that is only about 20-30% slower than very, very optimized C code and this is pretty awesome. The application was able to respond to market signals within single digit microseconds every time.

Real programmers can write FORTRAN in any language. The issue with Java isn't that you can't write low latency code. It's that you are left almost completely without tools. There's very little you can use from Java standard libraries. You're therefore left scratching your head about how to solve even the simplest problem like managing memory without something coming and suddenly creating a lot of latency where you don't want it to happen.

If you can code a FORTRAN-like Java, you can overcome this. And Java has multiple advantages, as follows:

1. Java is a simpler language than C++, but it allows for easy object modelling, which is not available in plain C. The smaller feature set of the language also helps the developers stay focused on the logic of the application rather than on expressing their technical superiority through application of all the bells and whistles available in the standard (which often makes C++ code hard to read and improve by others).

2021



https://collegeville.github.io/CW20/

Collegeville 2020 Home | ["Collegeville Workshop 2020"]

Better Scientific Software

Collegeville Workshop 2020

CW20 brings community members together to advance developer productivity for scientific software

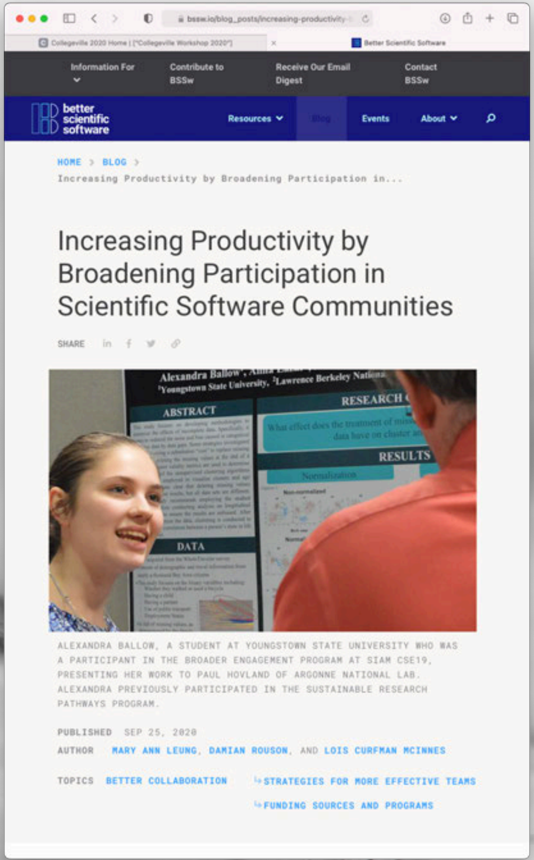
View On GitHub

This project is maintained by Collegeville

Collegeville 2020 Home

Details

- Workshop Survey
- Agenda Overview
  - Tuesday, July 21 Detailed Agenda
  - Wednesday, July 22 Detailed Agenda
  - Thursday, July 23 Detailed Agenda
- Resources
  - Posters
  - Recordings



bssw.io/blog_posts/increasing-productivity-

Collegeville 2020 Home | ["Collegeville Workshop 2020"]

Better Scientific Software

Information For | Contribute to BSSw | Receive Our Email Digest | Contact BSSw

better scientific software | Resources | Blog | Events | About

HOME > BLOG >

Increasing Productivity by Broadening Participation in...

Increasing Productivity by Broadening Participation in Scientific Software Communities

SHARE

ALEXANDRA BALLOW, A STUDENT AT YOUNGSTOWN STATE UNIVERSITY WHO WAS A PARTICIPANT IN THE BROADER ENGAGEMENT PROGRAM AT SIAM CSE19, PRESENTING HER WORK TO PAUL HOVLAND OF ARGONNE NATIONAL LAB. ALEXANDRA PREVIOUSLY PARTICIPATED IN THE SUSTAINABLE RESEARCH PATHWAYS PROGRAM.

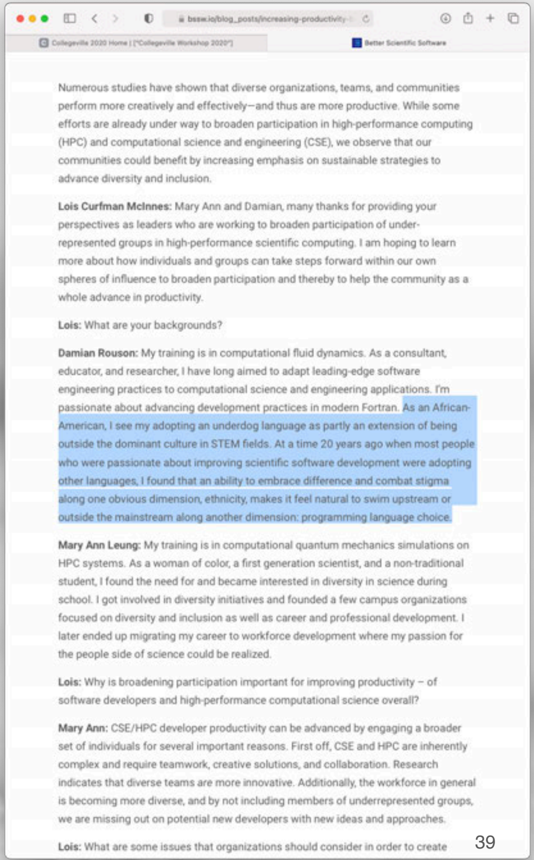
PUBLISHED SEP 25, 2020

AUTHOR MARY ANN LEUNG, DAMIAN ROUSON, AND LOIS CURFMAN MCINNES

TOPICS BETTER COLLABORATION

↳ STRATEGIES FOR MORE EFFECTIVE TEAMS

↳ FUNDING SOURCES AND PROGRAMS



bssw.io/blog_posts/increasing-productivity-

Collegeville 2020 Home | ["Collegeville Workshop 2020"]

Better Scientific Software

Numerous studies have shown that diverse organizations, teams, and communities perform more creatively and effectively—and thus are more productive. While some efforts are already under way to broaden participation in high-performance computing (HPC) and computational science and engineering (CSE), we observe that our communities could benefit by increasing emphasis on sustainable strategies to advance diversity and inclusion.

Lois Curfman McInnes: Mary Ann and Damian, many thanks for providing your perspectives as leaders who are working to broaden participation of under-represented groups in high-performance scientific computing. I am hoping to learn more about how individuals and groups can take steps forward within our own spheres of influence to broaden participation and thereby to help the community as a whole advance in productivity.

Lois: What are your backgrounds?

Damian Rouson: My training is in computational fluid dynamics. As a consultant, educator, and researcher, I have long aimed to adapt leading-edge software engineering practices to computational science and engineering applications. I'm passionate about advancing development practices in modern Fortran. As an African-American, I see my adopting an underdog language as partly an extension of being outside the dominant culture in STEM fields. At a time 20 years ago when most people who were passionate about improving scientific software development were adopting other languages, I found that an ability to embrace difference and combat stigma along one obvious dimension, ethnicity, makes it feel natural to swim upstream or outside the mainstream along another dimension: programming language choice.

Mary Ann Leung: My training is in computational quantum mechanics simulations on HPC systems. As a woman of color, a first generation scientist, and a non-traditional student, I found the need for and became interested in diversity in science during school. I got involved in diversity initiatives and founded a few campus organizations focused on diversity and inclusion as well as career and professional development. I later ended up migrating my career to workforce development where my passion for the people side of science could be realized.

Lois: Why is broadening participation important for improving productivity – of software developers and high-performance computational science overall?

Mary Ann: CSE/HPC developer productivity can be advanced by engaging a broader set of individuals for several important reasons. First off, CSE and HPC are inherently complex and require teamwork, creative solutions, and collaboration. Research indicates that diverse teams are more innovative. Additionally, the workforce in general is becoming more diverse, and by not including members of underrepresented groups, we are missing out on potential new developers with new ideas and approaches.

Lois: What are some issues that organizations should consider in order to create

Intersectionality

Prof. Kimberle Crenshaw
UCLA School of Law

University of Chicago Legal Forum

Volume 1989 | Issue 1 Article 8

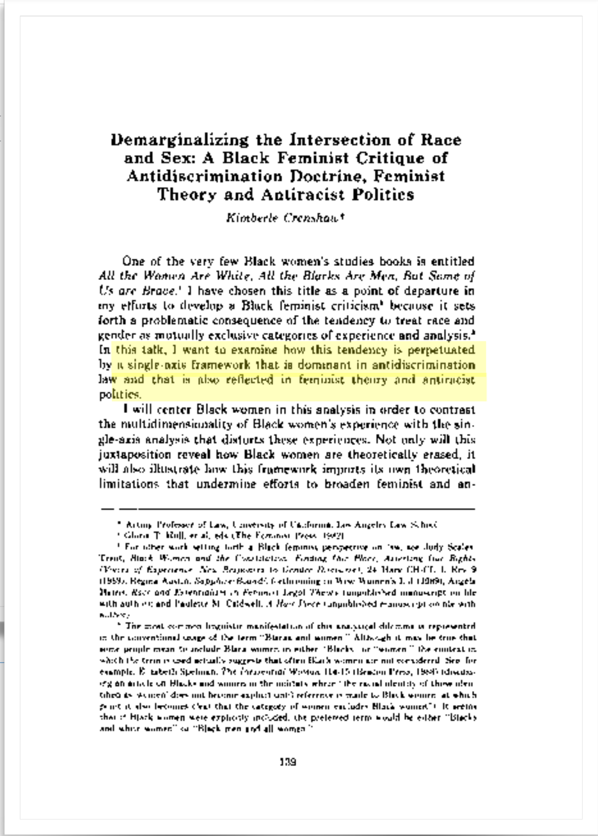
Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics

Kimberle Crenshaw
Kimberle.Crenshaw@chicagounbound.edu

Follow this and additional works at: http://chicagounbound.uchicago.edu/uclf

Recommended Citation
Crenshaw, Kimberle () "Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics," *University of Chicago Legal Forum*: Vol. 1989: Iss. 1, Article 8.
Available at: http://chicagounbound.uchicago.edu/uclf/vol1989/iss1/8

This Article is brought to you for free and open access by Chicago Unbound. It has been accepted for inclusion in University of Chicago Legal Forum by an authorized administrator of Chicago Unbound. For more information, please contact unbound@law.uchicago.edu.



Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics

Kimberle Crenshaw†

One of the very few Black women's studies books is entitled *All the Women Are White, All the Blacks Are Men, But Some of Us are Brave*.[1] I have chosen this title as a point of departure in my efforts to develop a Black feminist criticism[2] because it sets forth a problematic consequence of the tendency to treat race and gender as mutually exclusive categories of experience and analysis.[3] In this talk, I want to examine how this tendency is perpetuated by a single-axis framework that is dominant in antidiscrimination law and that is also reflected in feminist theory and antiracist politics.

I will center Black women in this analysis in order to contrast the multidimensionality of Black women's experience with the single-axis analysis that distorts these experiences. Not only will this juxtaposition reveal how Black women are theoretically erased, it will also illustrate how this framework imports its own theoretical limitations that undermine efforts to broaden feminist and an-

† Acting Professor of Law, University of California, Los Angeles Law School.

[1] Gloria T. Hull, et al, eds (The Feminist Press, 1982).

[2] [illegible]

[3] [illegible]

139

Google Scholar
citation count: 44,949

Conclusions

What Happens to a Dream Deferred?

01

Sometimes it sags like a heavy burden.

02

Sometimes it explodes in popularity.

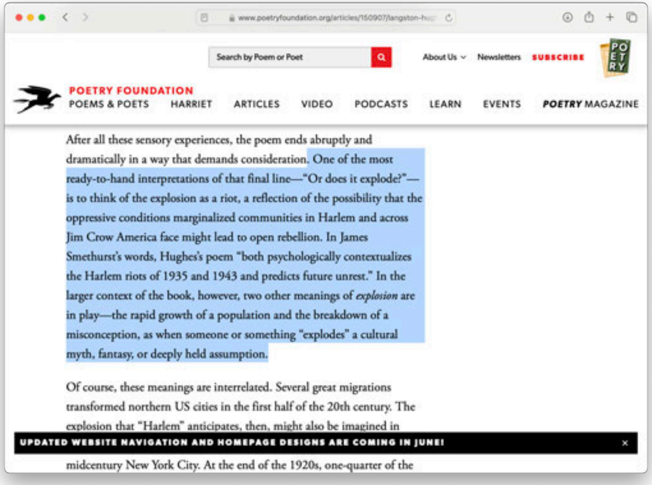
03

Sometimes it leads to open rebellion.

04

Background matters: $\Delta \in I$

Let's bring our whole selves to our work and bring math to the whole world.

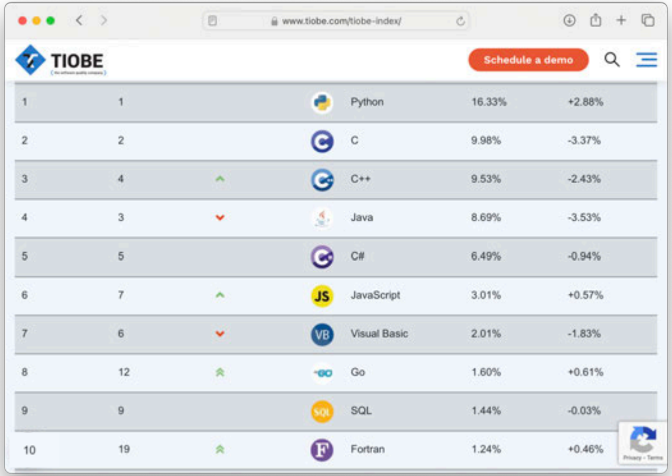


After all these sensory experiences, the poem ends abruptly and dramatically in a way that demands consideration. One of the most ready-to-hand interpretations of that final line—"Or does it explode?"—is to think of the explosion as a riot, a reflection of the possibility that the oppressive conditions marginalized communities in Harlem and across Jim Crow America face might lead to open rebellion. In James Smethurst's words, Hughes's poem "both psychologically contextualizes the Harlem riots of 1935 and 1943 and predicts future unrest." In the larger context of the book, however, two other meanings of *explosion* are in play—the rapid growth of a population and the breakdown of a misconception, as when someone or something "explodes" a cultural myth, fantasy, or deeply held assumption.

Of course, these meanings are interrelated. Several great migrations transformed northern US cities in the first half of the 20th century. The explosion that "Harlem" anticipates, then, might also be imagined in

UPDATED WEBSITE NAVIGATION AND HOMEPAGE DESIGNS ARE COMING IN JUNE!

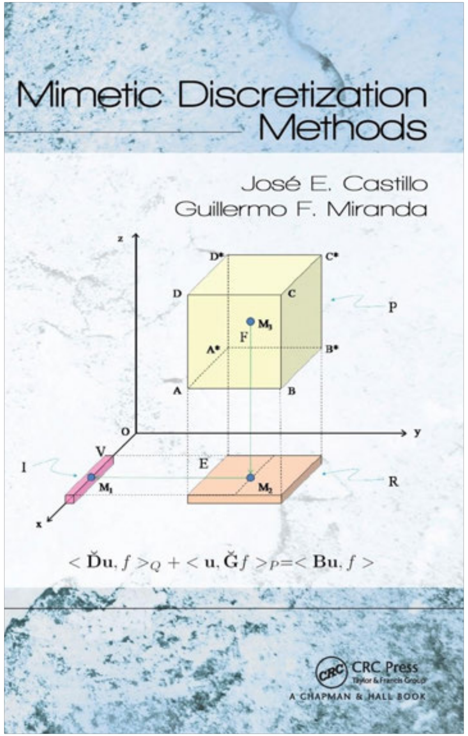
midcentury New York City. At the end of the 1920s, one-quarter of the

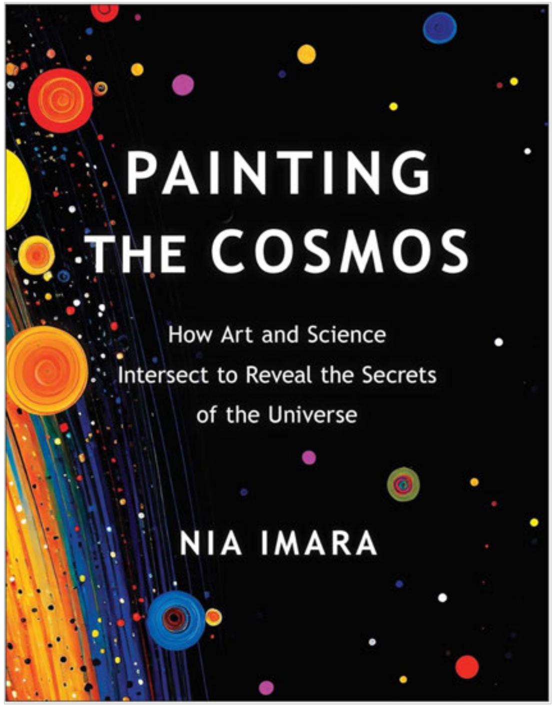


1	1			Python	16.33%	+2.88%
2	2			C	9.98%	-3.37%
3	4	^		C++	9.53%	-2.43%
4	3	v		Java	8.69%	-3.53%
5	5			C#	6.49%	-0.94%
6	7	^	JS	JavaScript	3.01%	+0.57%
7	6	v	VB	Visual Basic	2.01%	-1.83%
8	12	^^	GO	Go	1.60%	+0.61%
9	9		SQL	SQL	1.44%	-0.03%
10	19	^^	F	Fortran	1.24%	+0.46%

https://www.poetryfoundation.org/articles/150907/langston-hughes-harlem

Future Work:
My Reading List





Acknowledgements

The Berkeley Lab Fortran Team

Dan Bonachea, Paul Hargrove, Hugh Kadhem, Kate Rasmussen

Past and Present Collaborators

Zhe Bai, Jeremy Bailey, Kareem Ergawy, Michael Klemm, Ethan Gutmann, Baboucarr Dibba, Brad Richardson, David Torres, Kareem Jabbar Weaver, Jordan Welsman, Yunhao Zhang

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research