MIAPbP Workshop on Differentiable and Probabilistic Programming

Lukas Heinrich, TUM

Technische Universität München

ODSL ORIGINS Data Science Lab

MIAPbP

MIAPbP: Munich Institute for Astro- Particle- and Biophysics

45 Seats for 4 Weeks. Funded by TUM and ORIGINS Cluster of Excellence

Sanmay Ganguly

Our workshop was one of the first “Data Science/Computing” themed MIAPbP Workshop (together with QC)

MIAPbP

Munich Institute for Astro-, Particle and BioPhysics

Programs 2023

Interacting Supernovae
6 February - 3 March
L. Dessart, R. Margutti, J. Fuller, R. Kudritzki

Engineering life: Unifying concepts from system chemistry, biophysics and theoretical physics
13 - 24 March
C. Weber, J. Boekhoven, K. Göpfrich

Quantum Computing Methods for High Energy Physics
10 April - 5 May
C. Bauer, I. Cirac, M. Dalmonte, Z. Davoudi, H. S. Lamm, M. Spannowsky

The Present and Future of Heavy Flavour and Exotic Hadron Spectroscopy
8 May - 2 June
M. Barabanov, B. El-Bennich, M. Mikhasenko, S. Paul, E. Santopinto, L. Tolos

Differentiable and Probabilistic Programming for Fundamental Physics
5 - 30 June
L. Heinrich, T. Enßlin, M. Kagan, A. G. Baydin, V. Vassilev

The Extragalactic Distance Scale and Cosmic Expansion in the Era of Large Surveys and the James Webb Telescope
3 - 28 July
R. I. Anderson, S. H. Suyu, E. Di Valentino, S. W. Jha, H.-J. Seo

Stellar Astrophysics in the Era of Gaia, Spectroscopic, and Asteroseismic Surveys
31 July - 25 August
M. Bergemann, D. Huber, S. Hekker, A. Karakas, R. Kudritzki

Stellar Magnetic Fields from Protostars to Supernovae
9 October - 3 November
S. Bagnulo, L. Ferrario, A. Vidotto, G. A. Wade, K. Dolag

What is MIAPbP?

The Munich Institute for Astro-, Parti
(MIAPbP) hosts several programs per
biophysics, cosmology, nuclear and
MIAPbP serves as a center for scien
provides a stimulating platform for inf
collaborations and creative thinking.
www.munich-iapbp.de

Proposals for the 2024 program can be submitted until 3 October 2022:
www.munich-iapbp.de/propose

Application for participation:
www.munich-iapbp.de/register

MIAPbP Directors:
Prof. Dr. Andreas Weiler (TUM),
Prof. Dr. Rolf Kudritzki (LMU / University of Hawaii)

www.munich-iapbp.de
info@munich-iapbp.de
Excellence Cluster ORIGINS - MIAPbP -
Boltzmannstr. 2 - 85748 Garching, Germany

TUM LMU ORIGINS Excellence Cluster

and Exotic Hadron Spectroscopy
8 May - 2 June
M. Barabanov, B. El-Bennich, M. Mikhasenko, S. Paul, E. Santopinto, L. Tolos

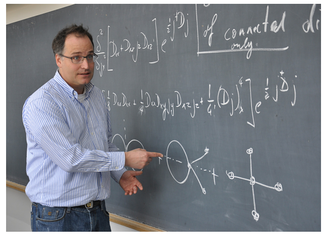
Differentiable and Probabilistic Programming for Fundamental Physics
5 - 30 June
L. Heinrich, T. Enßlin, M. Kagan, A. G. Baydin, V. Vassilev

The Extragalactic Distance Scale and Cosmic Expansion in

Proposals for the 2024 program can be submitted until 3 October 2022:
www.munich-iapbp.de/propose

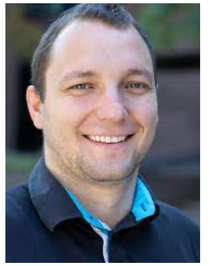
Application for participation:
www.munich-iapbp.de/register

Organizing Team



Torsten Ensslin
(MPA)

Atılım Güneş Baydin
(Oxford)



Vassil Vassilev
(Princeton)

Michael Kagan
(SLAC)

Lukas Heinrich
(TUM)

Why we did it

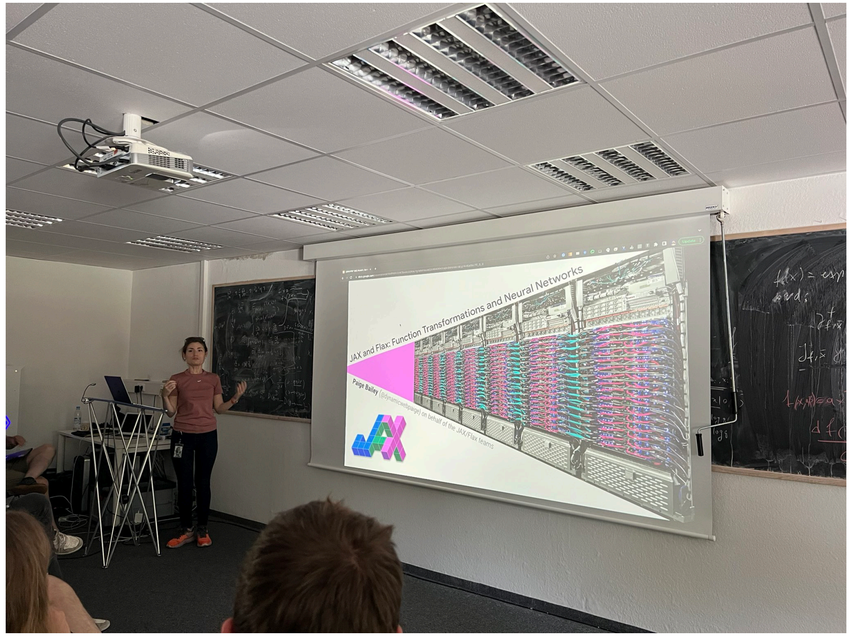
The topics of the workshop are still new to fundamental physics. Deeply interesting, but also outside of our usual wheelhouse.

Goals:

- **provide venue where there is a lot of space for discussions and time to think**
- **bring together physicists and computer scientists for an extended period of time**

Format

Format: 1-2 Talks a day - Rest is discussion!



Talks available here (not sometimes it’s a URL)

https://www.munich-iapbp.de/probabilistic-programming/schedule

Impromptu Sessions

We had many unplanned impromptu sessions as well: Measure Theory, Julia, Distributed DiffProg, Theorem Proving, Geant4

Takeaway (as HEPer): a lot of folks have deep expertise in things that often don't surface in HEP context. The format helped a lot.

Topical Workshop

We had a more traditional workshop for 3 days as well (~ 80 attendees)

Let me fix this

Why we did it

My reaction when learning about this program:

via GIPHY

Display a menu

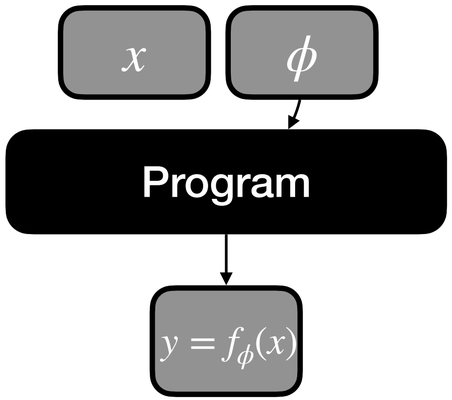
2

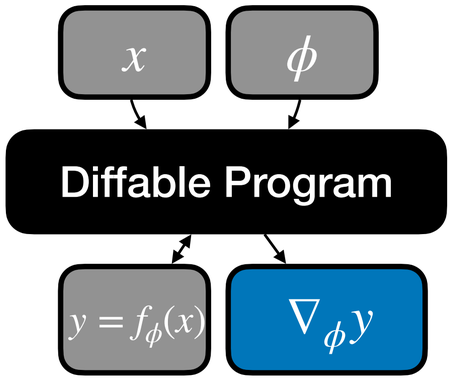
Francois Lenusse: First Talk of the Program

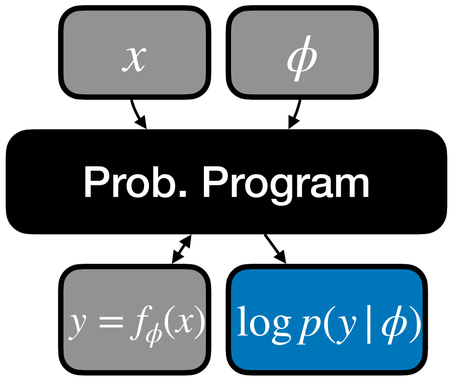
Why ProbProg and DiffProg

From a certain point of view Differentiable and Probabilistic Programming are closely connected.

- **but not a lot of cross-talk between communities**







DiffProg Needs ProbProg

To differentiate through some of the non-differentiable operations we often to, it's useful to make the program stochastic first (see. M. Kagan's Talk)

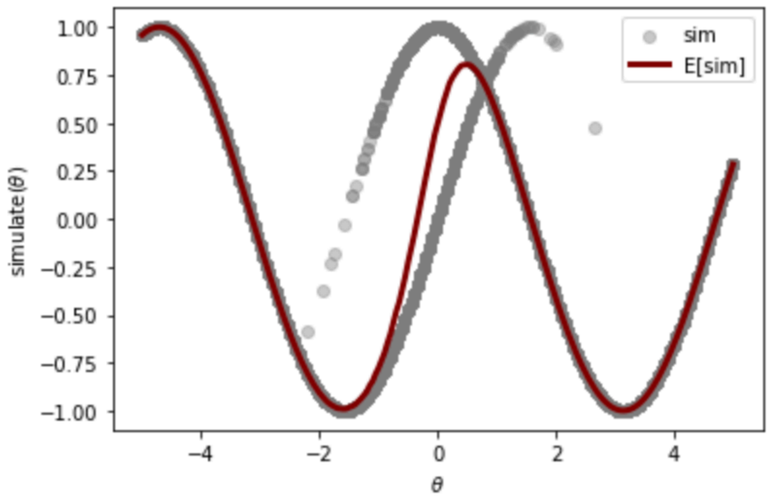
```python
def simulate(theta):
  p = sigmoid(theta)
  if p > 0.5:
    r = sin(theta)
  else:
    r = cos(theta)
  return r
```

→

```python
def simulate(theta):
  p = sigmoid(theta)
  b = bernoulli(b) # 0 or 1
  if b == 1:
    r = sin(theta)
  else:
    r = cos(theta)
  return r
```

→

Discrete Jumps

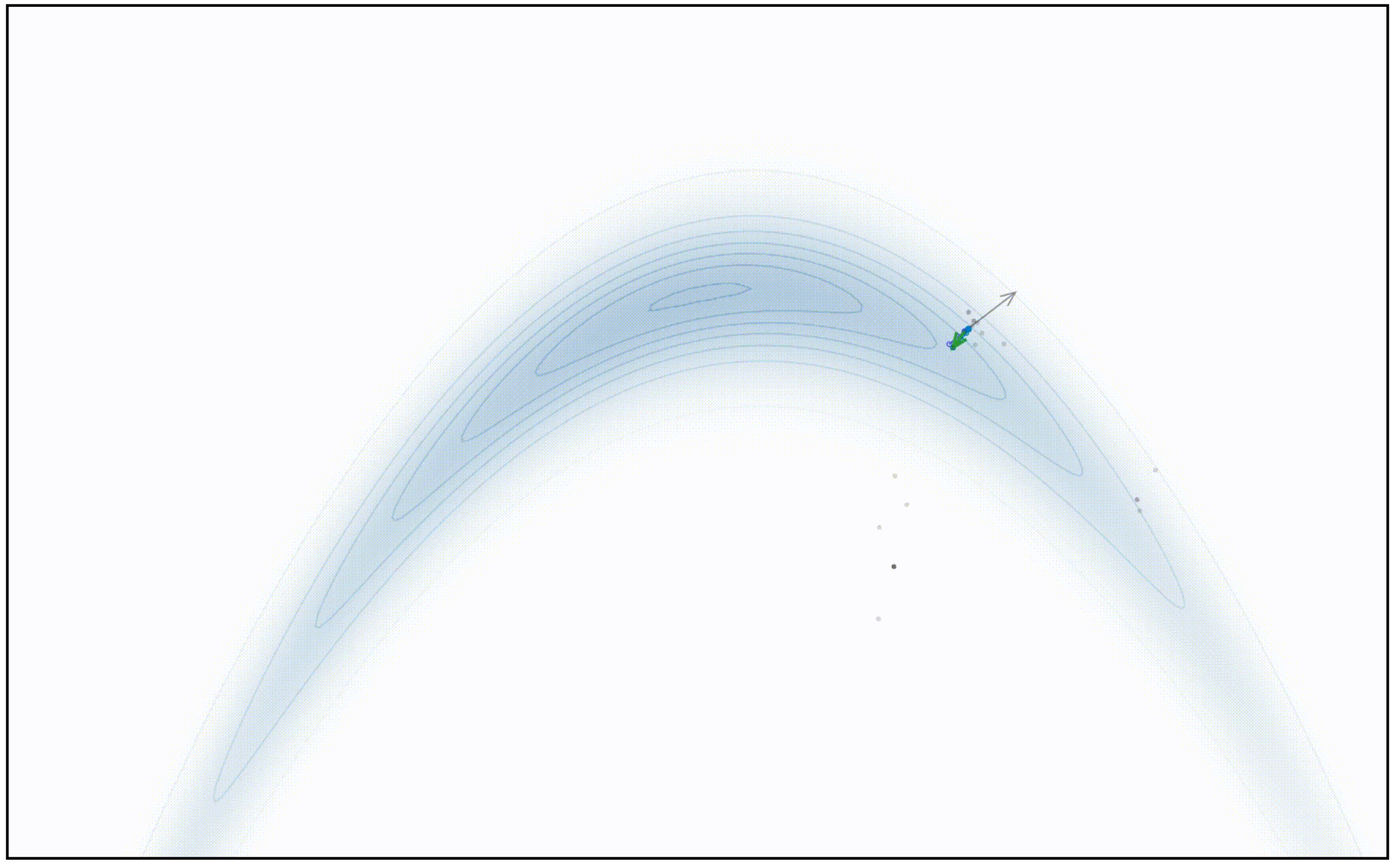


Smooth Expectation Value

ProbProg needs DiffProg

Statistical Inference with PPLs often <u>require</u> gradients for efficient exploration of the parameter space

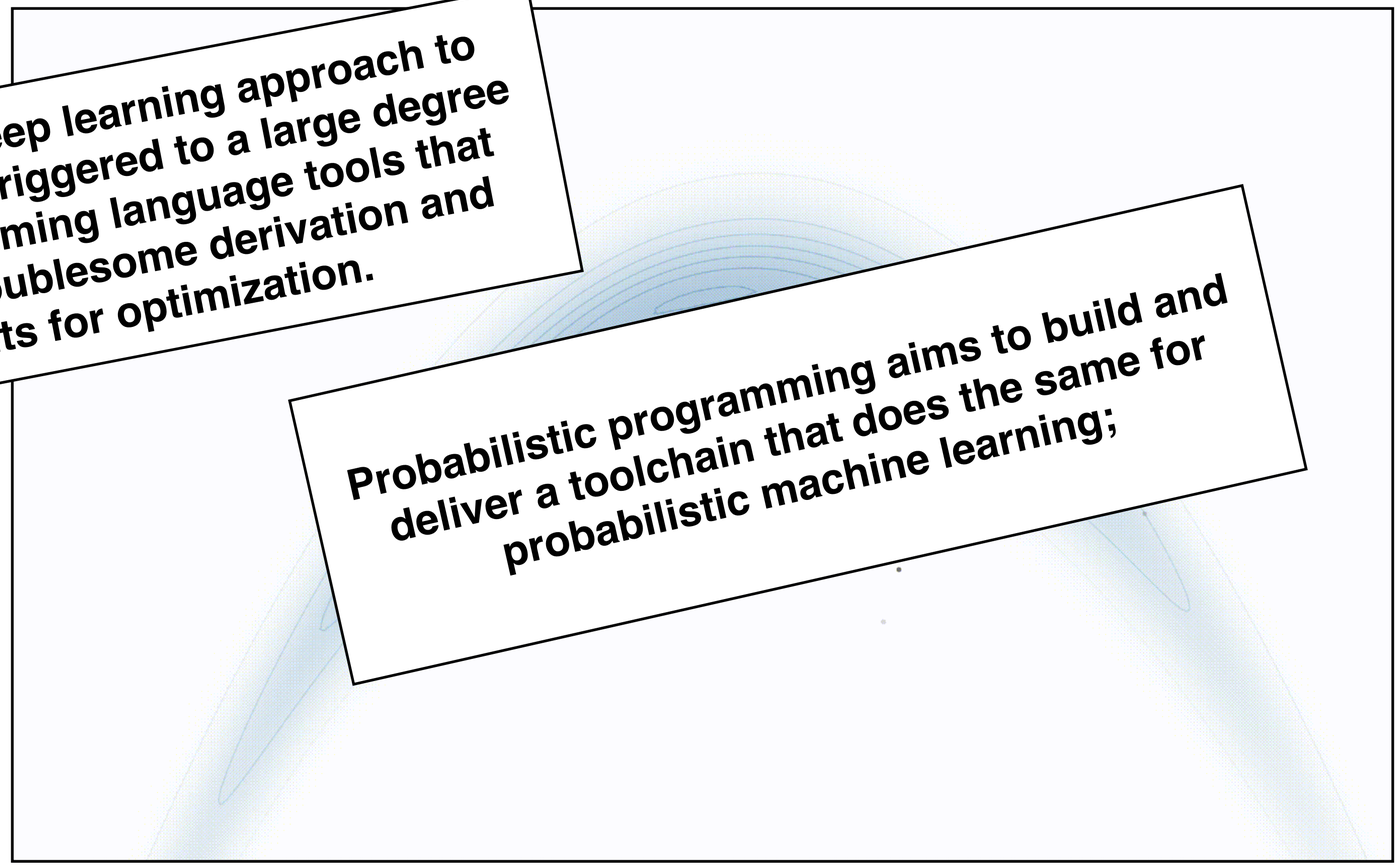
$$\nabla_\theta \log p(x, \theta)$$



ProbProg needs DiffProg

Statistical Inference with PPLs often <u>require</u> gradients for efficient exploration of the parameter space

∇_θ



A few Takeaways

Tooling

The tooling is continuously improving and is growing (or always has been) beyond ML.

JAX: the quasi-default for a lot of “new” differentiable programming work in particle physics

Why? Most people got exposed to DP through ML. DP as a way to add physics inductive bias. JAX much better suited than other ML frameworks

Case in Point:

Astro-folks are investing a lot into rewriting simulation code in JAX (and TF). From primordial fields to final inference.

Target: HMC

How complicated can it be to s

Display a menu

FlowPM

build passing | pypi package 0.1.1 | Open in Colab | astro-ph.IM arXiv:2010.11847 | youtube | code style yapf | docs failing

Particle Mesh Simulation in TensorFlow, based on fastpm-python simulations

Try me out: Open in Colab

To install:

```
$ pip install flowpm
```

For a minimal working example of FlowPM, see this notebook. The steps are as follows:

```
import tensorflow as tf
import numpy as np
import flowpm

cosmo = flowpm.cosmology.Planck15()
stages = np.linspace(0.1, 1.0, 10, endpoint=True)

initial_conditions = flowpm.linear_field(32,          # size of the cube
                                         100,         # Physical size of the cube
                                         ipklin,      # Initial power spectrum
                                         batch_size=16)

# Sample particles
state = flowpm.lpt_init(cosmo, initial_conditions, a0=0.1)

# Evolve particles down to z=0
final_state = flowpm.nbody(cosmo, state, stages, 32)
```

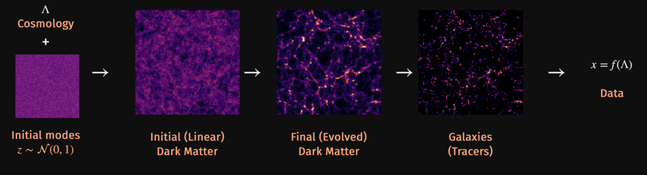
Francois Lenusse

Field-level Inference with forward modeling

→ Novel analysis with latents
Custom gradients for auto-dif
High dimensional inference

Use computational simulations to forward-model the field-level data without any compression.
→ Simultaneously infer the initial conditions of the Universe and the cosmological parameters.

Computational forward-models in cosmology

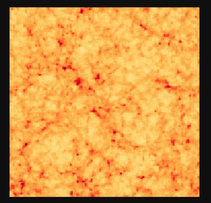


Boltzmann equations

Gravity evolution (N-Body/ PM simulations) (Λ)

Halo finding + Halo distribution/ Semi-analytic models (θ)

Survey systematics & summarize (θ)



Data
$x = f(z_0)$

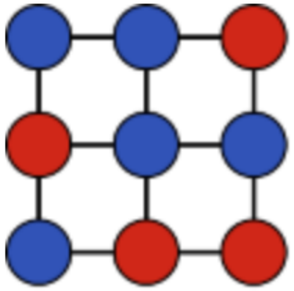
Chirag Modi

Beyond JAX

JAX can be a good choice for new projects, but most of our code is not even implemented in Python let alone JAX

A lot of tooling developing for multi-language AD and scientific computing in particular

TAPENADE



CoDiPack

Enzyme

CLAD

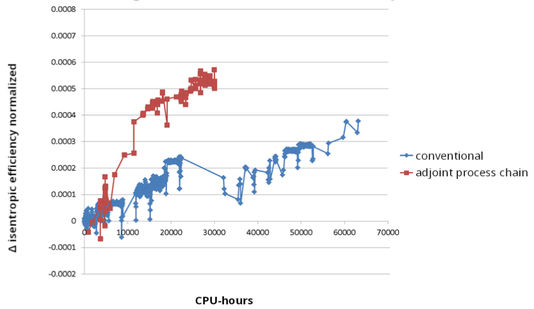
Takeaway: It Scales

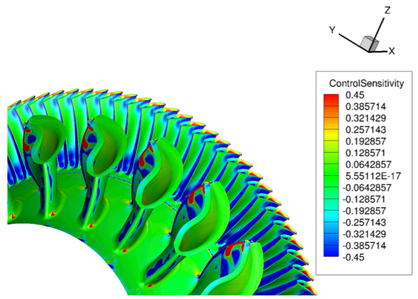
If things are differentiable, we shouldn't be scared of large-scale codebases and applications

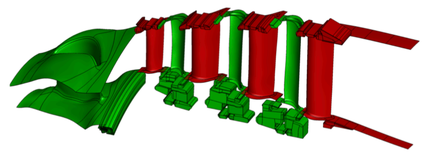
Scientific Computing — University of Kaiserslautern-Landau (RPTU)

Optimization of a 4-stage low pressure turbine (MTU Aero Engines)

- **Shape optimization:**
  4-stage low pressure turbine
  TMTF[1], Cavities
- **Mesh:** 7.2 Mio. Cells
- **Solver:** DLR TRACE
- **Simulation:** RANS, Transition + Turbulence
- **Target functional:** Efficiency







[1] - Turning Mid Turbine Frame

Scientific Computing — University of Kaiserslautern-Landau (RPTU)

Optimality System

- **Optimization Problem:**

$$\min_{\phi\in\Phi} J(W,\phi) \quad s.t. \quad R(W,\phi)=0 \quad \leftrightarrow$$

Fixed point iteration: $W = G(W,\phi)$

- **Lagrangian:**

$$L = J + \Lambda^T R \quad \leftrightarrow \quad L(W,\Lambda,\phi) = J(W,\phi) + \Lambda^T (G(W,\phi) - W)$$

- **Optimality condition (KKT system, 1. order necessary cond.):**

$$\frac{\partial L}{\partial \Lambda} = R \overset{!}{=} 0 \qquad \text{State equation}$$

$$\frac{\partial L}{\partial W} = \frac{\partial J}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} \overset{!}{=} 0 \qquad \text{Adjoint state equation}$$

$$\frac{\partial L}{\partial \phi} = \frac{\partial J}{\partial \phi} + \Lambda^T \frac{\partial R}{\partial \phi} \overset{!}{=} 0 \qquad \text{Design equation}$$

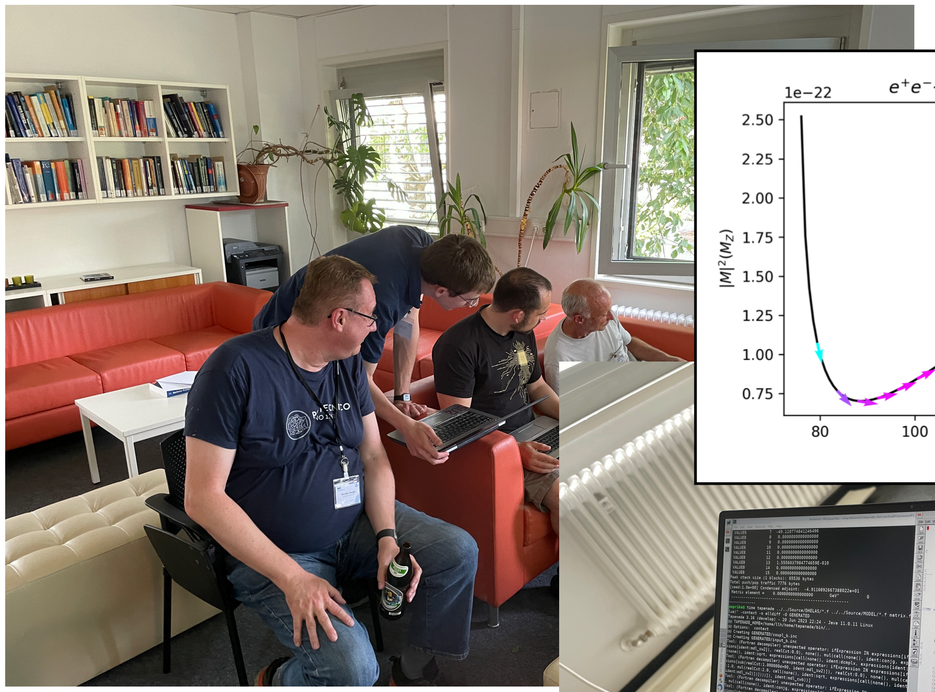
A Taste of what's possible

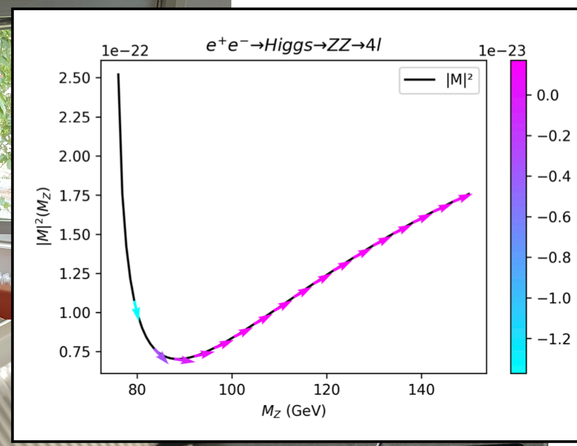
MadJax: Differentiable Matrix Elements from MadGraph (FORTRAN)

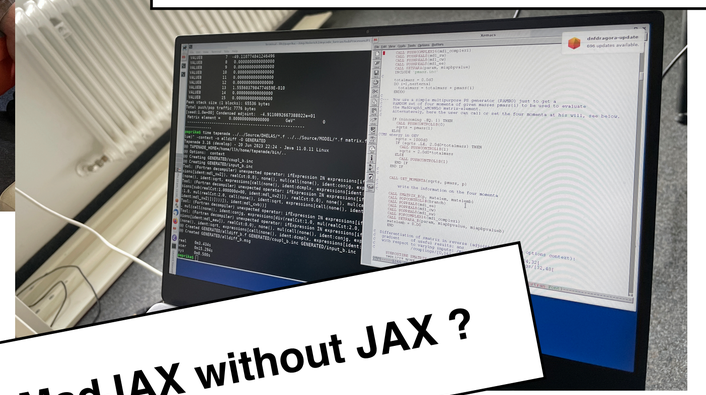
In JAX: painful compilation

In CoDiPack, Tapenade:
Gradients within O(day) on original code (C++ and/or Fortran)

Extra: Derivgrind derivatives on compiled artefact







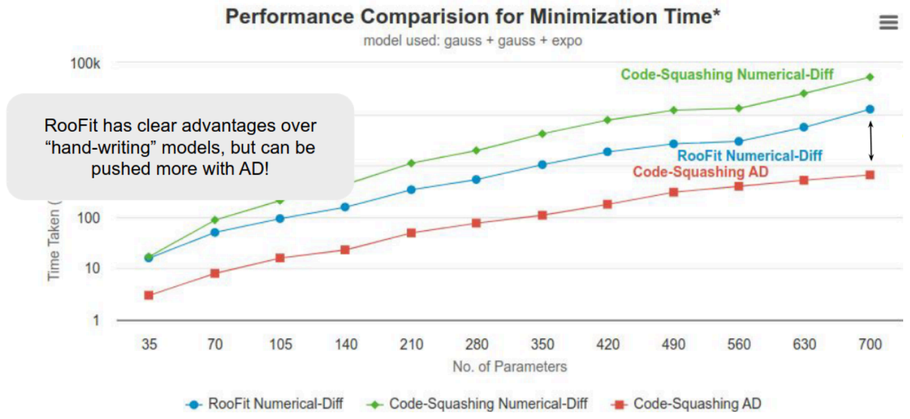
MadJAX without JAX ?

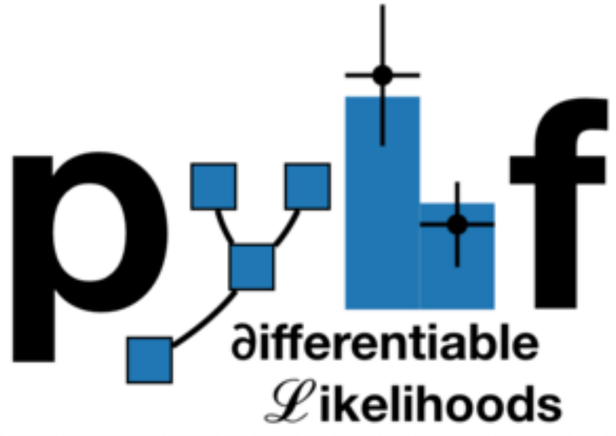
A Taste of what's possible

RooFit: after ~10 years, it actually is becoming differentiable

[Garima Singh]

Maybe indicative: first small-scale rewrite in JAX (***pyhf***) to demonstrate feasibility and added value, and then motivate larger effort & deployment (with C++ / CLAD)





Beyond Automatic Differentiation

Standard AD is not a magic black-box machine

Jan Hückelheim:

> ***"AD is for people who already know what the gradients are"***

There are a lot of "tricks-of-the-trade" that one has to be aware of and that should be added to languages as features

Examples

$$\partial_\theta \text{odesolve}(f, x_0)$$

$$\partial_\theta \int_{\Omega(\theta)}$$

$$\partial_\theta \mathbb{E}_{p(x,\theta)}$$

$$\partial_\theta \text{argmin}_x f(x, \theta)$$

Possible Outcome of Workshop: Handbook of AD constructs (B. Pearlputter, G. Baydin)

Application: Differentiable Rendering

Modern photorealistic rendering technology is insanely good

- at 2014, around 60-75% of all IKEA's product images (images showing only a single produ
  are CG. 35% of non-product images are fully CG.

these are all com

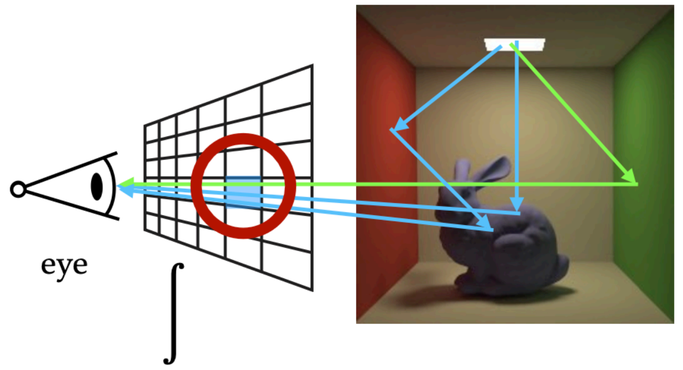
Mathematically

$$\partial_\theta \iint \blacksquare = \iint \partial_\theta \blacksquare + \int \blacksquare$$

area boundary

derived through Dirac delta or Reynolds transport theorem

Key observation: the **integrand** is discontinuous, but the **integral** is differentiable



Li, Aittala, Durand, Lehtinen, 2018

Differentiably Tracing Photons might be a good starting point for differentiably tracking more particles

Special Role of HEP

Notable difference in uptake of DP in Astro and HEP

HEP: local algorithms become differentiable or we have big surrogates (fits, track reconstruction, generative models)

- Natural blockage at discrete changes in data representation

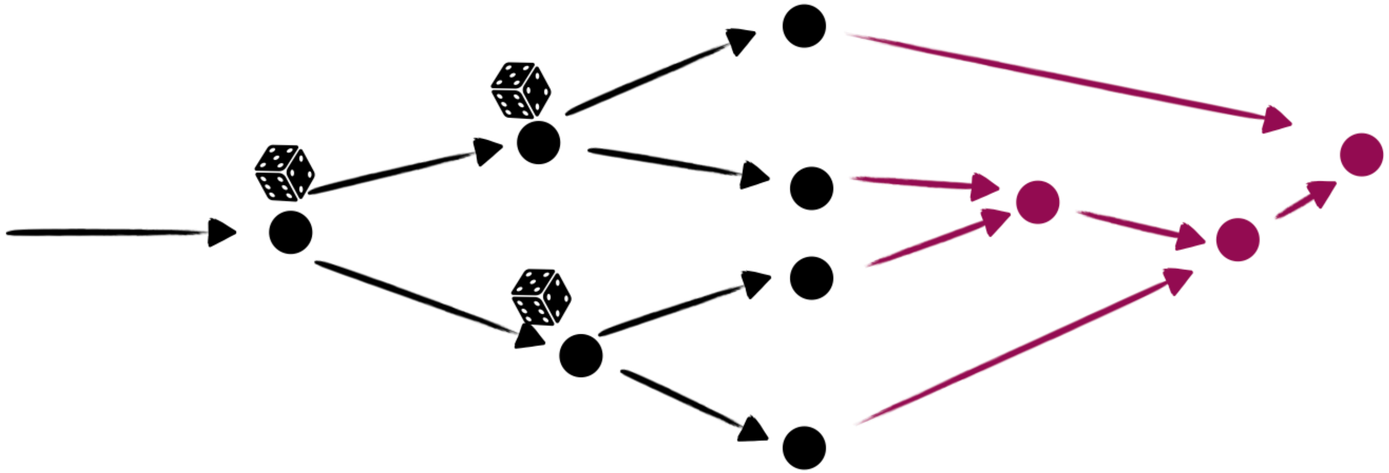
Astro: ambition is more towards soup-to-nuts differentiable pipelines

- A lot of applications are more “inherently differentiable”

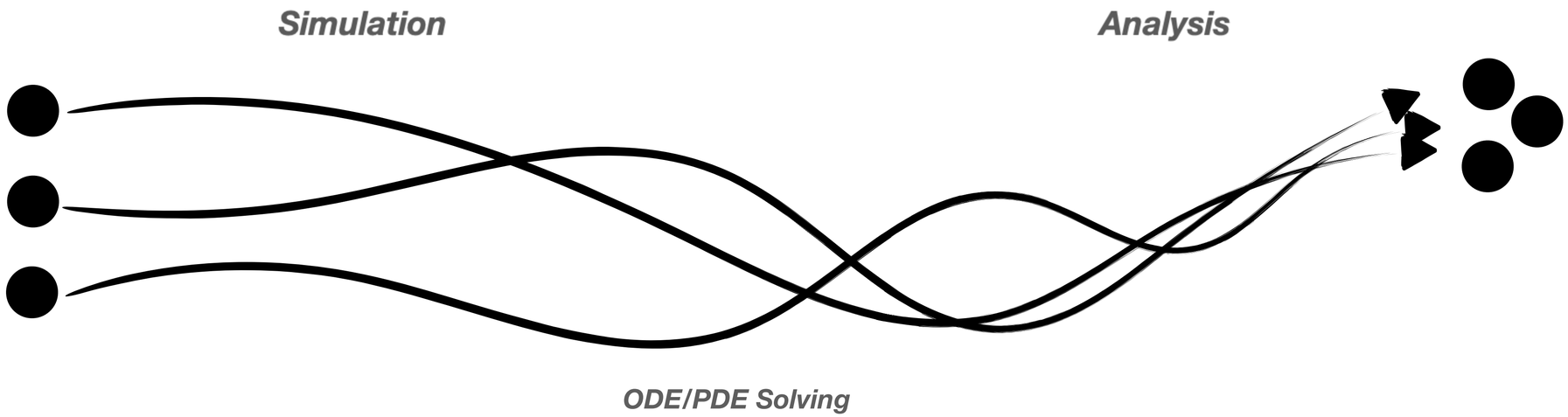
Special Role of HEP

Notable difference in uptake of DP in Astro and HEP

HEP:



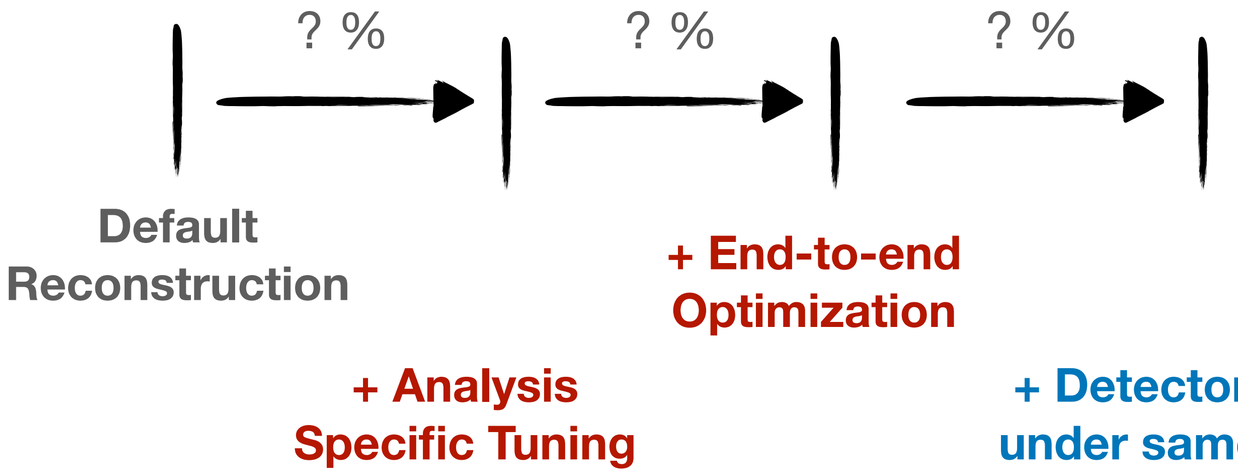
Astro:

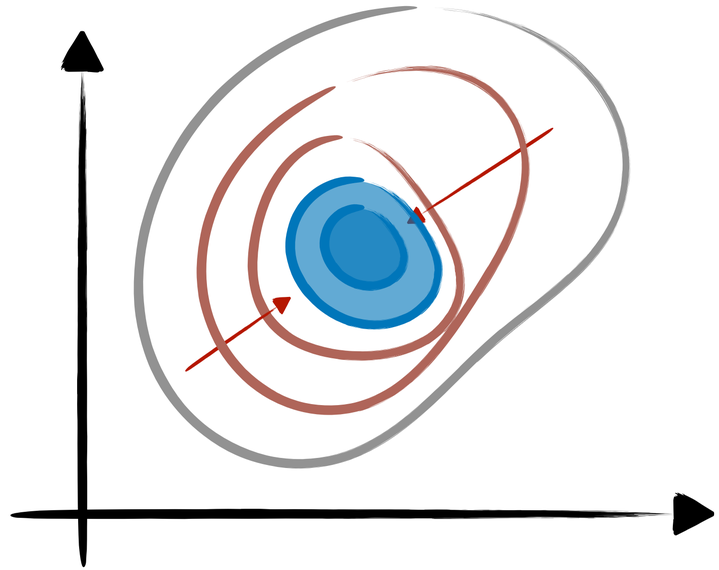


A Big Question

Why do we do differentiable programming. Presumably to gain performance. But how much is there to gain?

What is the hierarchy of approaches, how big are the gaps?





A Big Question

Why do we do differentiable programming. Presumably to gain performance[illegible]e to gain?

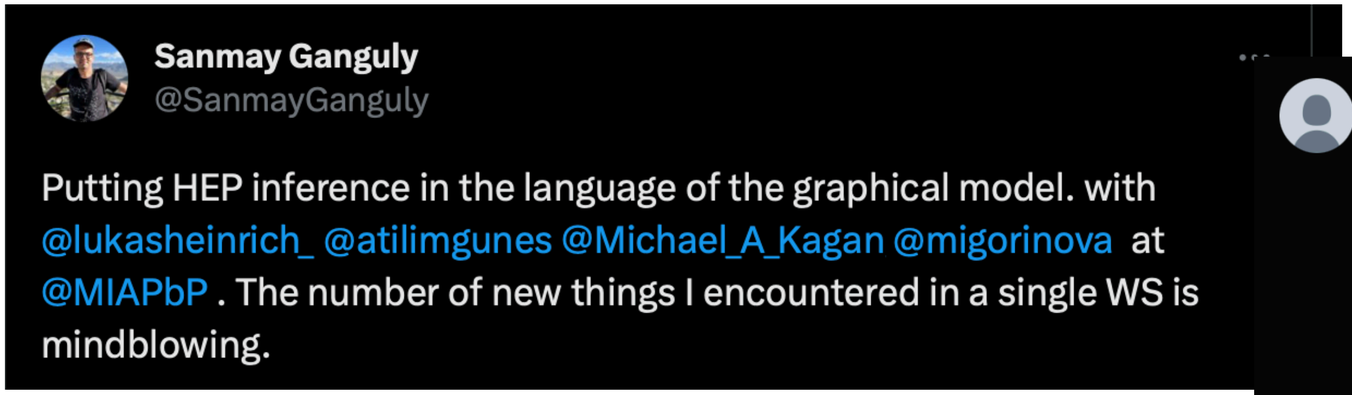
What is the h[illegible]s, how big are the gaps?

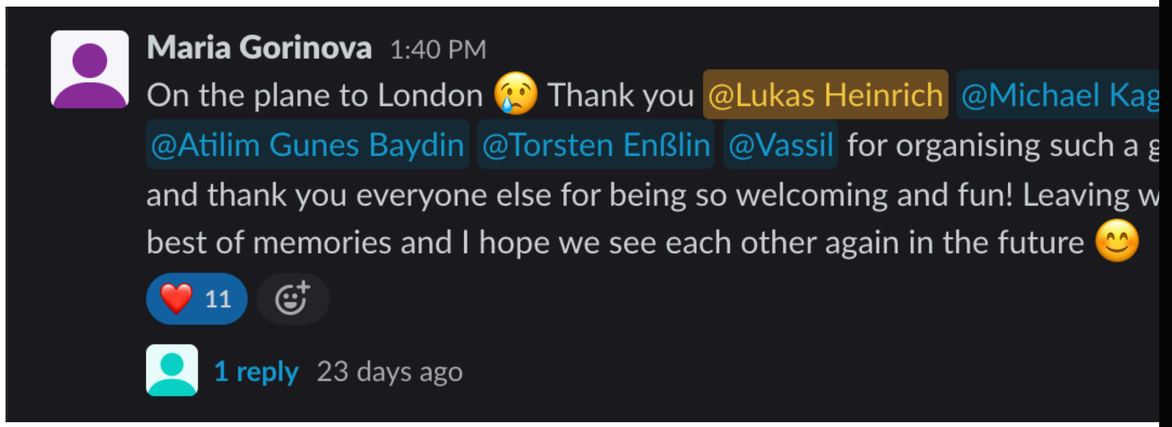
Possible outcome statistical analysis on impact of various inference simplifications we often do in HEP

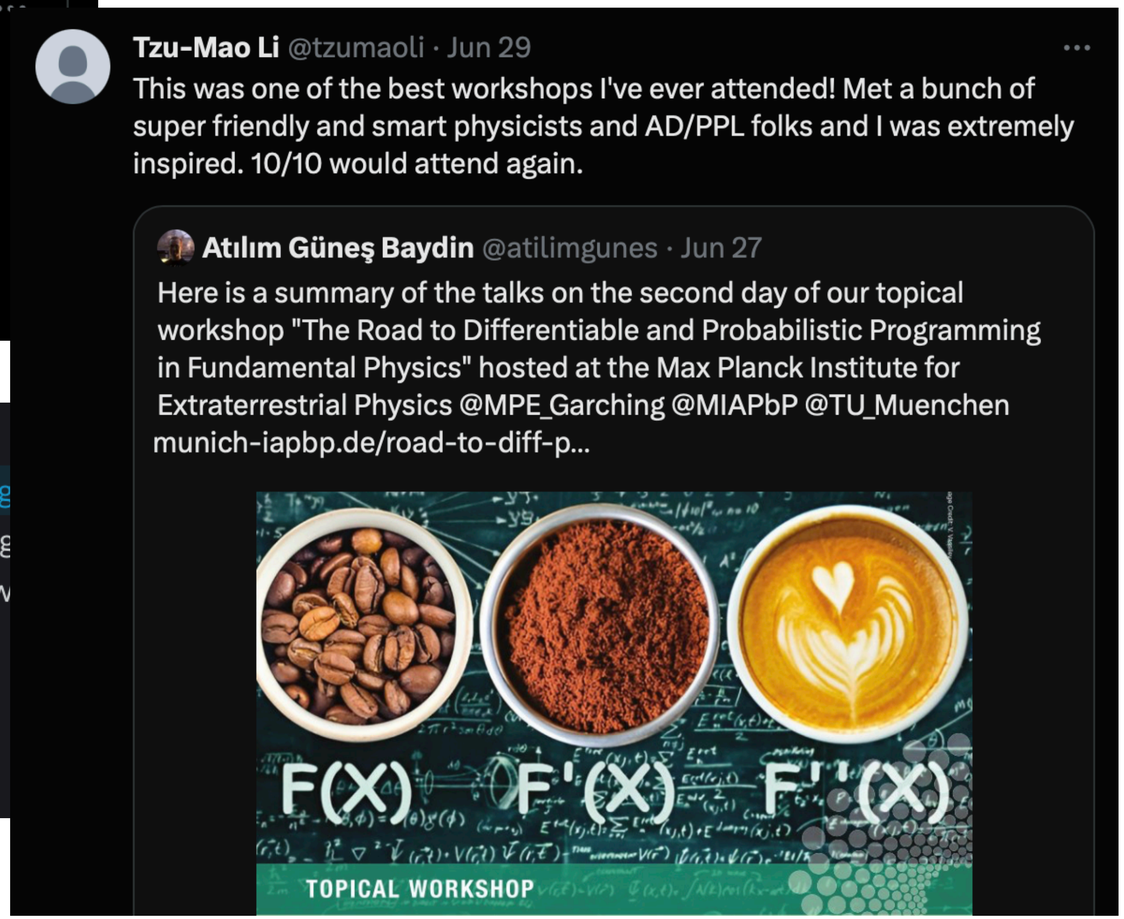
The Main Takeaway

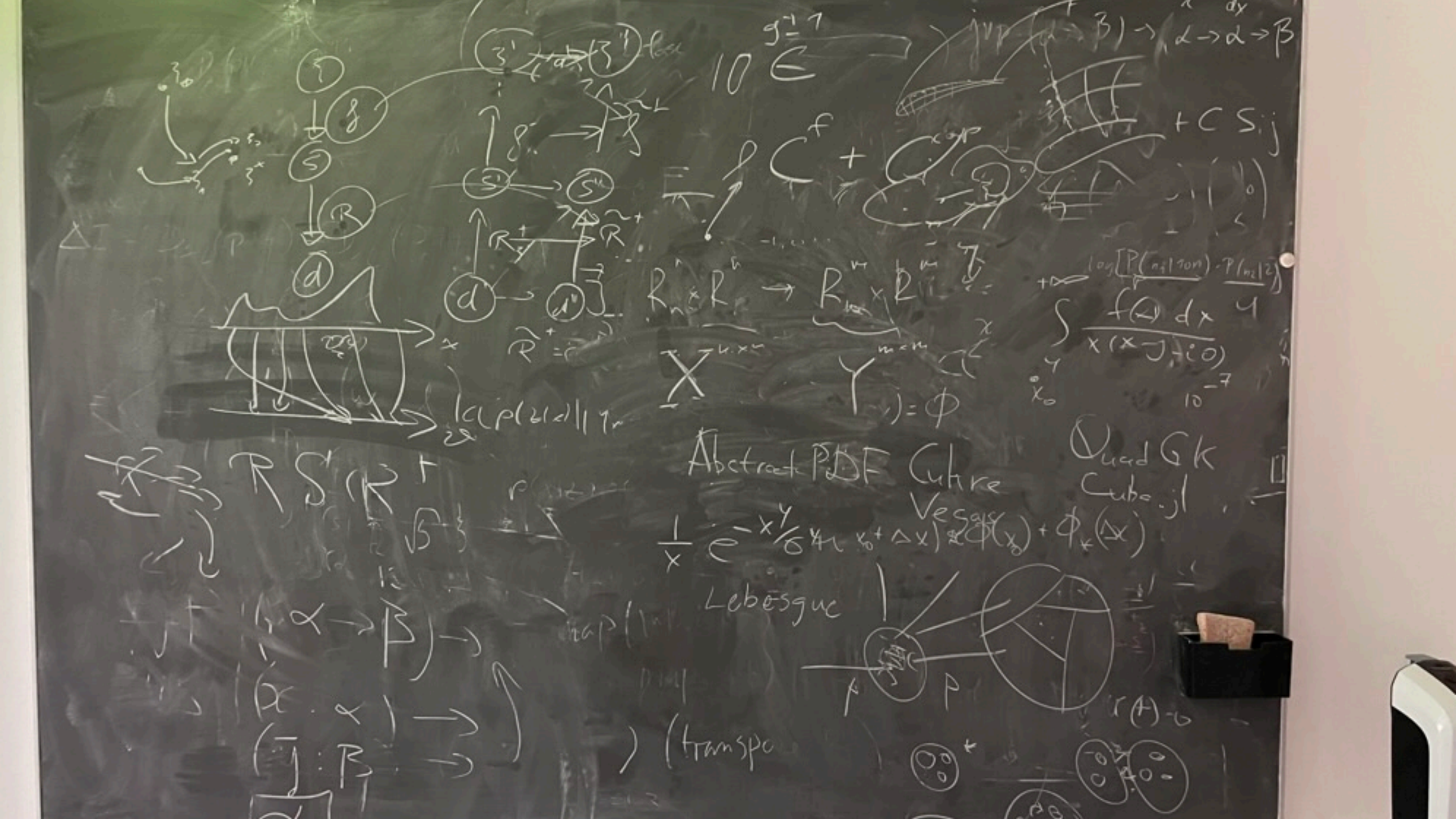
We should do it again sometime!

Currently exploring options around the world








Abstract PDF Cuhre
Vegas
QuadGk
Cube.jl
Lebesgue