Fermilab Quantum Computing Testbed Approaches

James Amundson, *Fermilab*

with contributions from James Kowalkowski, Adam Lyon, Alexandru Macridin, Gabriel Perdue and Panagiotis Spentzouris

December 6, 2017

Background

– Fermilab and Fermilab Computing

– Quantum Computing Entering 2018

Fermi National Accelerator Laboratory

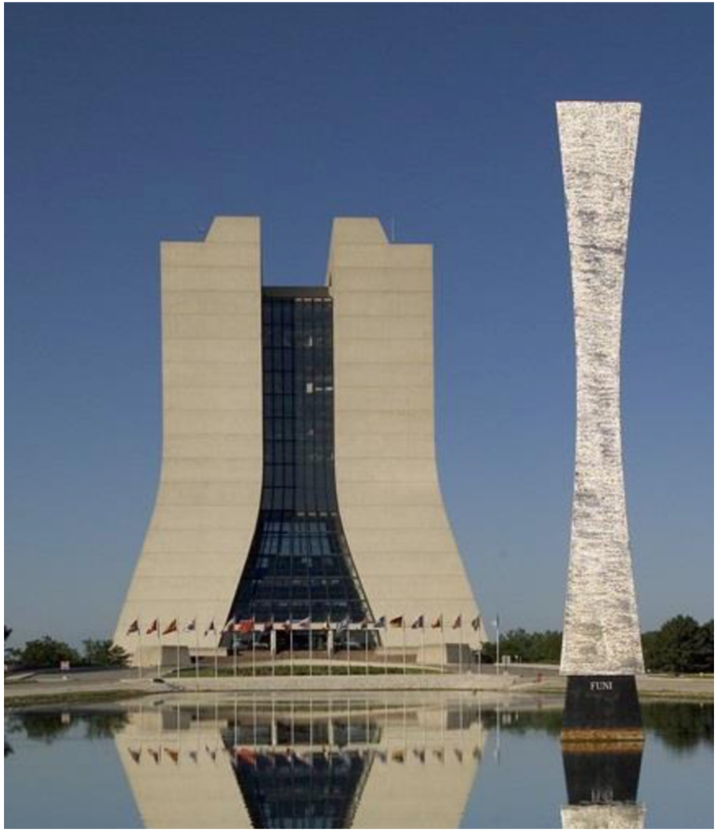
America’s premier laboratory for particle physics and particle accelerator research

One of the few single-purpose DOE national labs

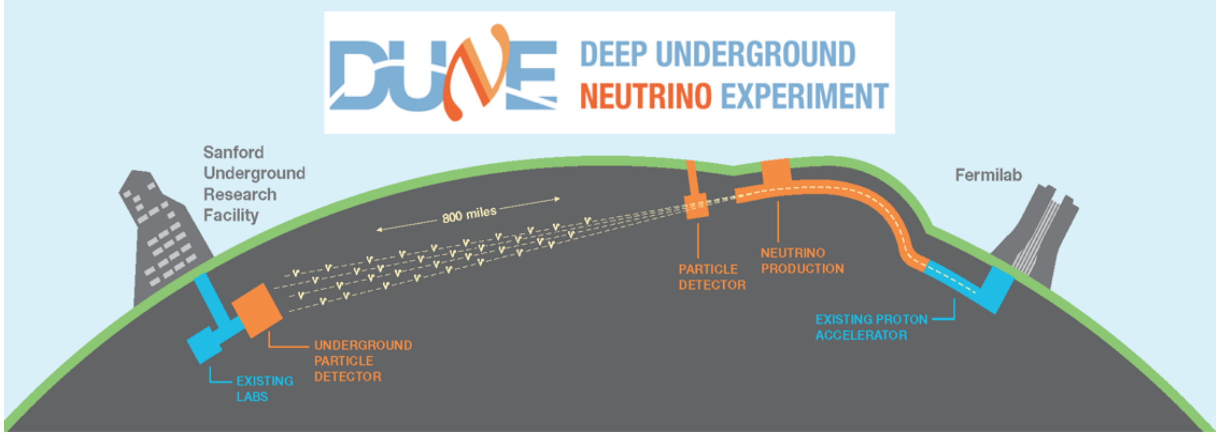
With 4,500 scientists from 50 countries, we aim to discover what the universe is made of and how it works

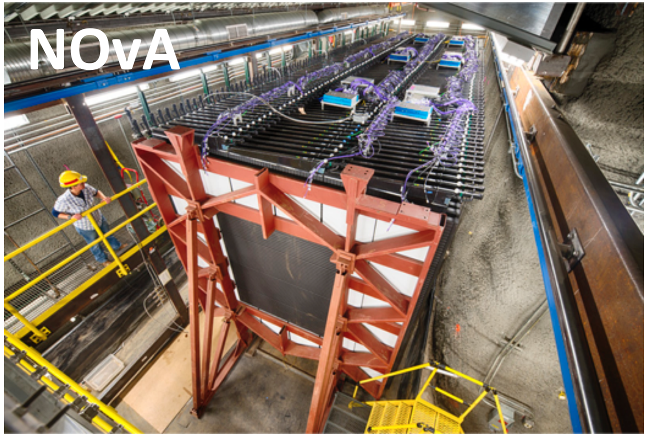
We study the smallest building blocks of matter and probe the farthest reaches of the universe using some of the most complex particle accelerators, detectors, and computing systems in the world

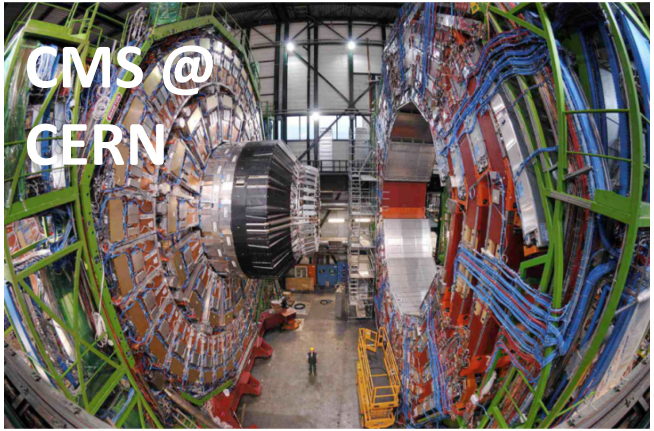
Fermilab is managed by Fermi Research Alliance for the U.S. Department of Energy Office of Science

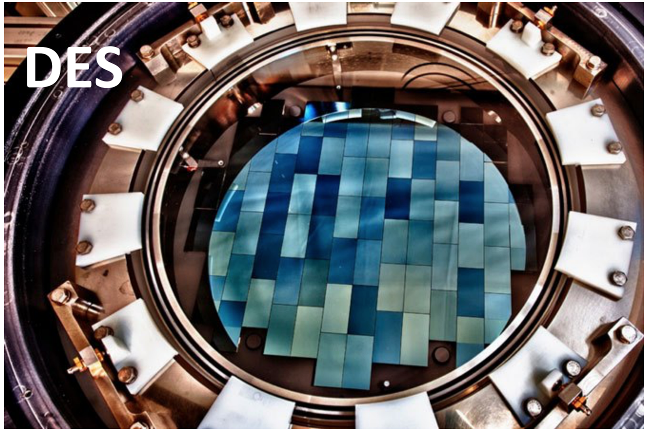


Experiments (LHC, Neutrinos, Muons)





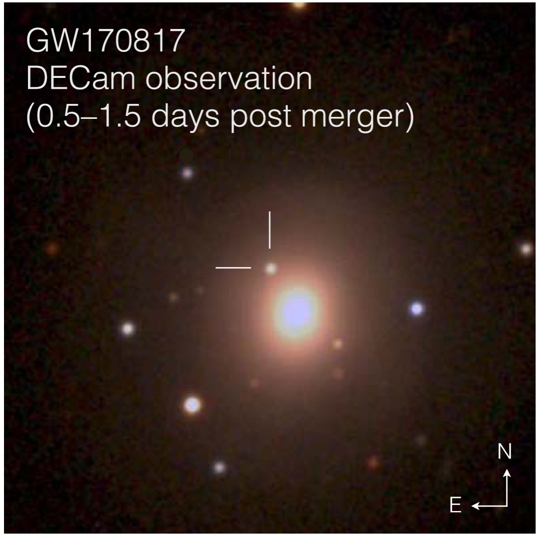


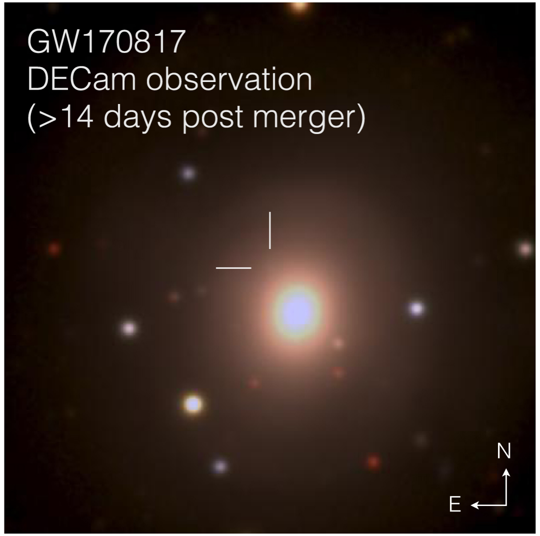


Discovery of Optical Counterpart to GW170817 with DECam

LIGO and Virgo recently announced discovery of Gravitational Waves from colliding neutron stars

Resulting kilonova imaged in many wavelengths by many telescopes, including the Blanco 4m in Chile with the Fermilab built Dark Energy Camera (DECam)





- Very intense high-throughput computing utilization to process images in search for source
- Project uses resources at Fermilab and opportunistic resources on the Open Science Grid
- Processing involves many algorithms for image subtraction, cleanup and source detection
- Backgrounds from moving objects and point-source transients are rejected with Machine Learning *(doi:10.1088/0004-6256/150/3/82)*

Talk by Marcelle Soares Santos, Brandeis University
http://iopscience.iop.org/article/10.3847/2041-8213/aa9059/meta

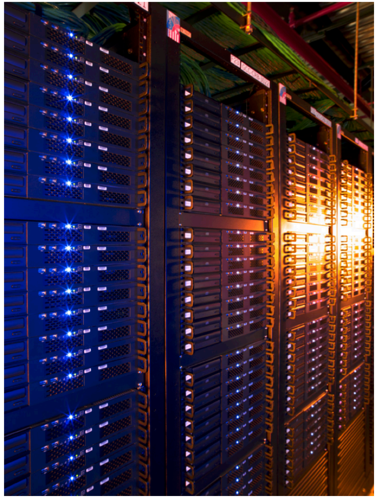
High Energy Physics (HEP) Computing at Fermilab

Fermilab is the largest source of HEP computing support in the US

- Hardware
  - Large-scale high-throughput computing resources
    - CPU
    - Storage
- Common Services
  - Core software development support
    - Frameworks
      - CMSSW and *art*
        - Two closely related frameworks for CMS and Intensity Frontier experiments (muons, neutrinos, etc.), respectively
  - Scientific Workflows
  - Grid Computing
  - HEPCloud

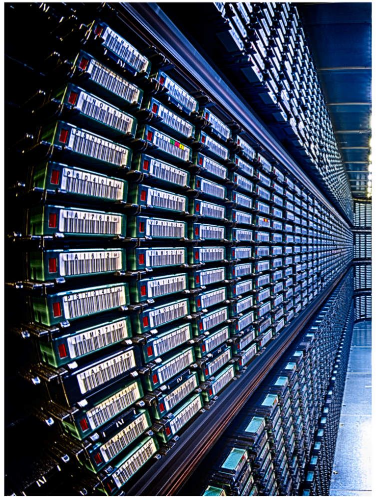
Fermilab Facilities

CPU



48,000 cores
(plus 20,000 HPC cores)

Tape

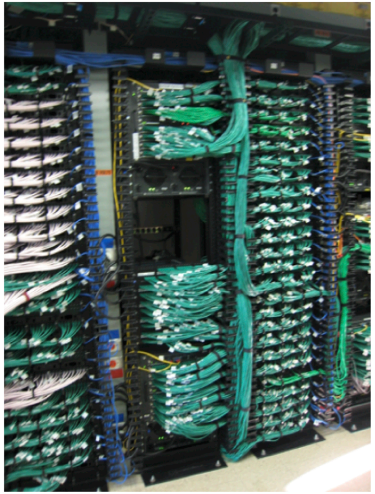


1 Exabyte Capacity
100 PB of tape media, 90 PB used

Disk

35 PB spinning disk

Network

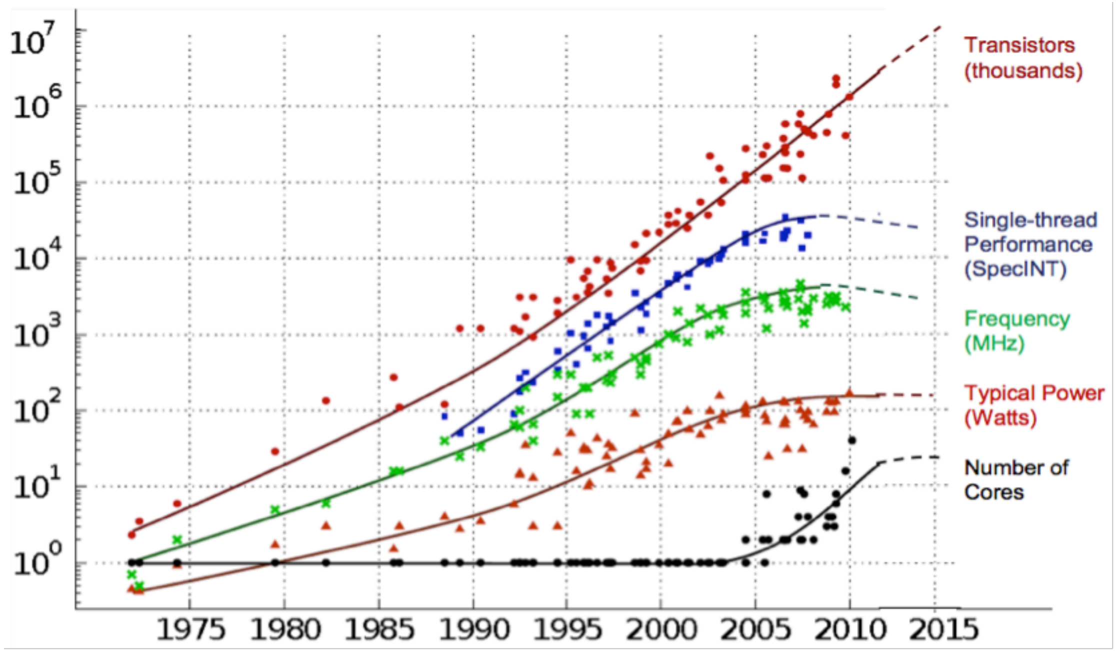


2x100 Gb/s offsite
~35,000 internal network ports

Open Science Grid

WLCG
Worldwide LHC Computing Grid

Growth in Classical Computing is not What it Used to Be



“Data Processing in Exascale-Class Computing Systems”, Chuck Moore, AMD Corporate Fellow and CTO of Technology Group, presented at the 2011 Salishan Conference on High-speed Computing, Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten, dotted line extrapolations by C. Moore

Background

– Fermilab and Fermilab Computing

– Quantum Computers Entering 2018

Fermilab

Few-qubit Quantum Computers Have Merged

- Several companies and labs have announced quantum computers in the 5-22 qubit range
  - Rigetti, Google, IBM, Intel, others…
  - Academic efforts
  - D-Wave has quantum *annealing* machines with more qubits
- These machines can be simulated on moderate-sized classical computers
- Preskill: Quantum Supremacy
  - Demonstrate a quantum computer that can do things that are beyond the limits of classical computers
    - n. b.: not necessarily *useful*
  - Estimated to require roughly 50 qubits
    - Recent advances in classical simulation have pushed that up a little…

Fermilab

Newer Quantum Hardware is Becoming Interesting

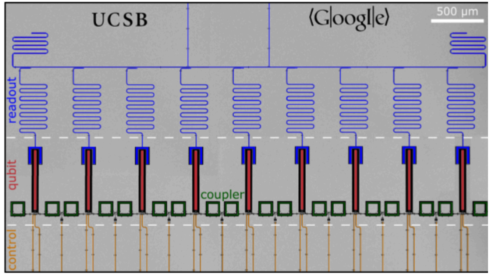
MIT Technology Review

Intelligent Machines

Google Reveals Blueprint for Quantum Supremacy

The ability of quantum machines to outperform classical computers is called quantum supremacy. Now Google says it has this goal firmly in its sights.

by Emerging Technology from the arXiv   October 4, 2017



Intelligent Machines

IBM Raises the Bar with a 50-Qubit Quantum Computer

Researchers have built the most sophisticated quantum computer yet, signaling progress toward a powerful new way of processing information.

by Will Knight   November 10, 2017

Counting Qubits is not Enough

Quantum Volume

Lev S. Bishop, Sergey Bravyi, Andrew Cross, Jay M. Gambetta, John Smolin

March 4, 2017

1 Executive Summary

As we build larger quantum computing devices capable of performing more complicated algorithms, it is important to quantify their power. The origin of a quantum computer's power is already subtle, and a quantum computer's performance depends on many factors that can make assessing its power challenging. These factors include:

1. The number of physical qubits;
2. The number of gates that can be applied before errors make the device behave essentially classically;
3. The connectivity of the device;
4. The number of operations that can be run in parallel.

Quantum Computing ideal is still far away

- Early results generated excitement about the possibilities of quantum computers
- One of the first examples: factoring large numbers
  - Taken from LA-UR-97-4986 “Cryptography, Quantum Computation and Trapped Ions,” Richard J. Hughes (1997)

Size of modulus (bits)	1,024	2,048	4,096
Factoring time in 1997	10^7 years	$3x10^{17}$ years	$2x10^{31}$ years
Factoring time in 2006	10^5 years	$5x10^{15}$ years	$3x10^{29}$ years
Factoring time in 2015	2,500 years	$7x10^{13}$ years	$4x10^{27}$ years
Factoring time in 2024	38 years	10^{12} years	$7x10^{25}$ years
Factoring time in 2033	7 months	$2x10^{10}$ years	10^{24} years
Factoring time in 2042	3 days	$3x10^8$ years	$2x10^{22}$ years

Table 2: Projected future factoring times using the GNFS for various moduli using 1,000 workstations.

Size of modulus (bits)	512	1,024	2,048	4,096
Quantum memory (qubits)	2,564	5,124	10,244	20,484
Number of quantum gates	$3x10^9$	$3x10^{10}$	$2x10^{11}$	$2x10^{12}$
Quantum factoring time	33 seconds	4.5 minutes	36 minutes	4.8 hours

Table 3: Quantum factoring times of various moduli on a hypothetical 100-MHz QC.

Fermilab

Quantum Computing ideal is still far away

- Current machines can use O(100) gates
  - Compared to today: 10^2x – 10^3x qubits required for factoring, 10^7x – 10^{10}x usable gates

Size of modulus (bits)	512	1,024	2,048	4,096
Quantum memory (qubits)	2,564	5,124	10,244	20,484
Number of quantum gates	3×10^9	3×10^{10}	2×10^{11}	2×10^{12}
Quantum factoring time	33 seconds	4.5 minutes	36 minutes	4.8 hours

Table 3: Quantum factoring times of various moduli on a hypothetical 100-MHz QC.

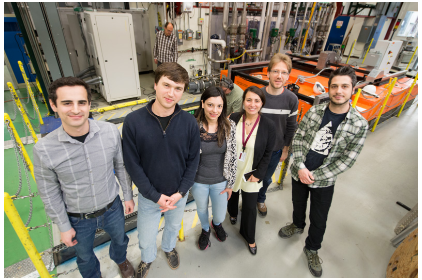
Quantum Testbeds for HEP

- Quantum Computing in HEP
- Quantum Testbed Plan
- Candidate Quantum Applications

Fermilab Quantum Hardware Initiatives

- Quantum sensors
  - Adapting quantum devices for use as quantum sensors for particle physics experiments such as direct dark matter detection
- Superconducting technologies
  - Some quantum computers use superconducting cavities similar to those we develop for accelerators.
- Quantum networks
  - We have agreed to host a quantum network on site in collaboration with Caltech and AT&T

Quantum sensors for axion search LDRD by Aaron Chou, Andrew Sonnenschein, and Dan Bowring



Fermilab SRF group is in a R&D collaboration with U. Chicago and Argonne

Quantum networks visit with John Donovan of AT&T

Quantum Computing in HEP

There is a significant body of QIS work from the theoretical HEP community

- Emphasis on "theoretical"
  - Example titles from Workshop on Computational Complexity and High Energy Physics (U. Maryland, 7/31 – 8/2):
    - *"Black holes, entropy, and holographic encoding"*
    - *"Computational complexity of cosmology in string theory"*
    - *"Computability theory of closed timelike curves"*

– See, however… this workshop!

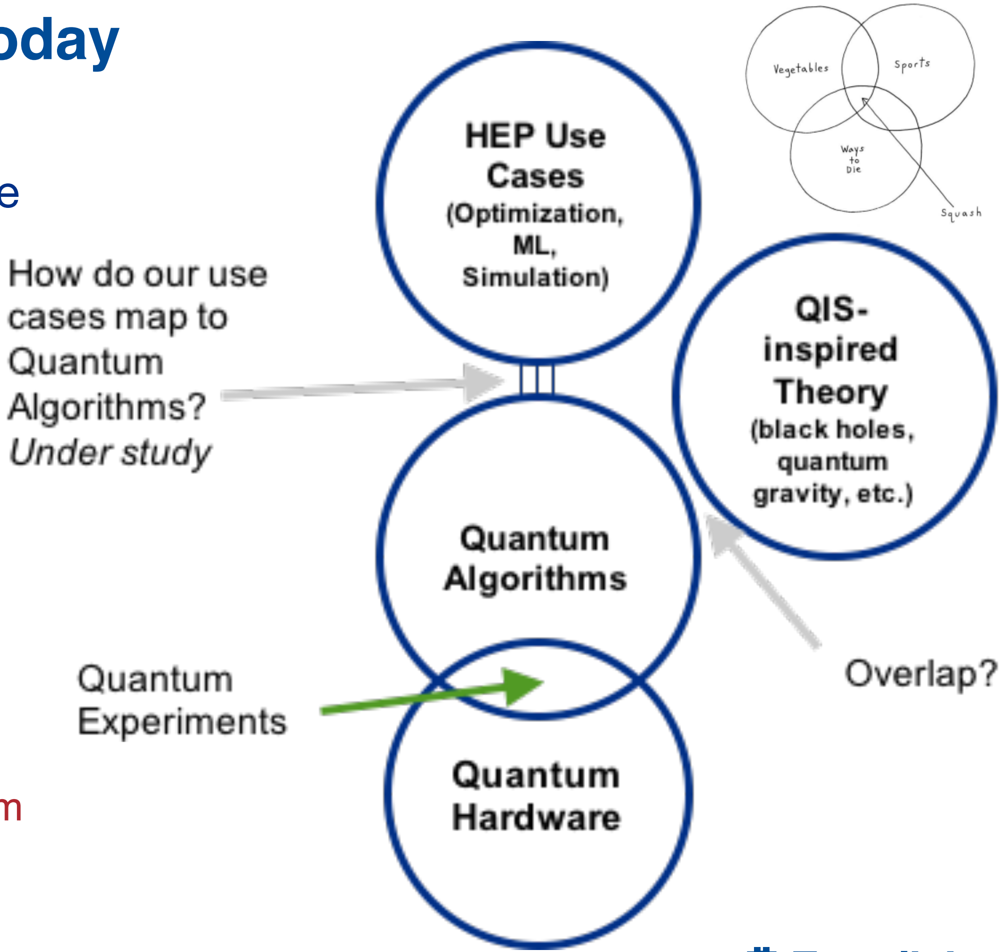
Majority of HEP computing is very different from current quantum computing ideas

- Trivially parallelizable problem (statistically independent events)
- Very complex code without dominant kernels
  - LHC experiment code is $O(10^7)$ lines C++

Quantum Computing in HEP Today

The gap between theoretical work and existing (or soon-to-exist) hardware is large

- We propose to facilitate the transition from theory to practice
- Implement algorithms, more likely parts of algorithms
  - Investigate parameters and scalability, impact of errors
    - Input and output, especially
      - We are data-driven
  - We do not need to solve a complete problem in order to make progress
- We need to start somewhere
  - We may not be directly pointed to Quantum Nirvana…



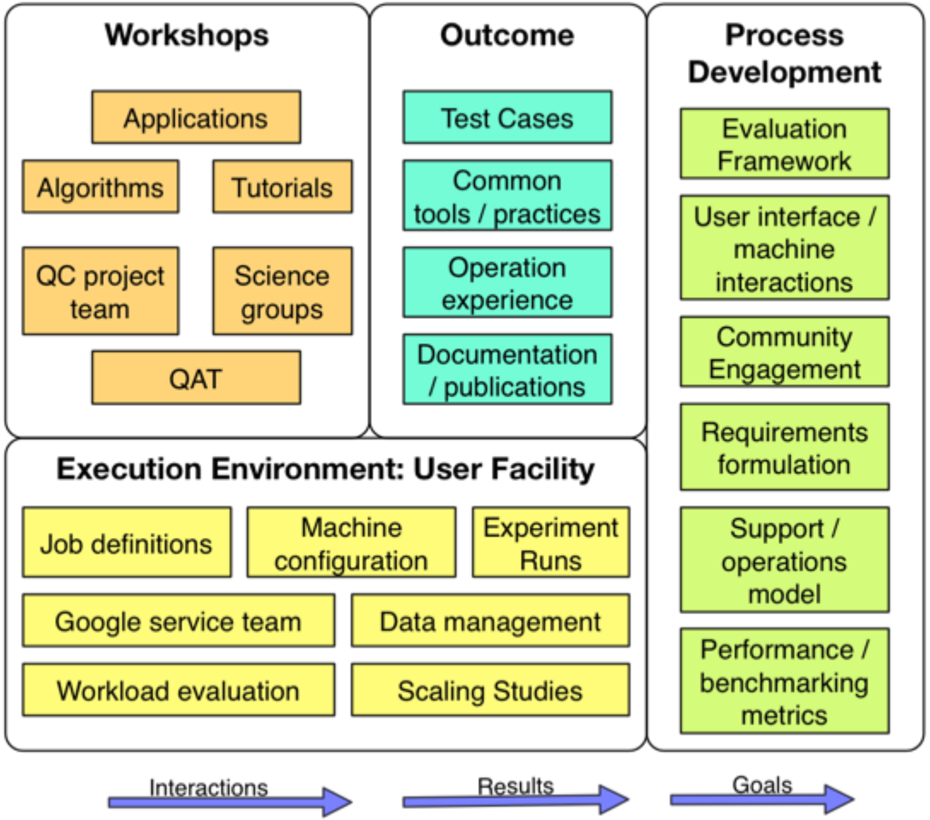
Fermilab

Quantum Testbeds for HEP

- Quantum Computing in HEP
- Quantum Testbed Plan
- Candidate Quantum Applications

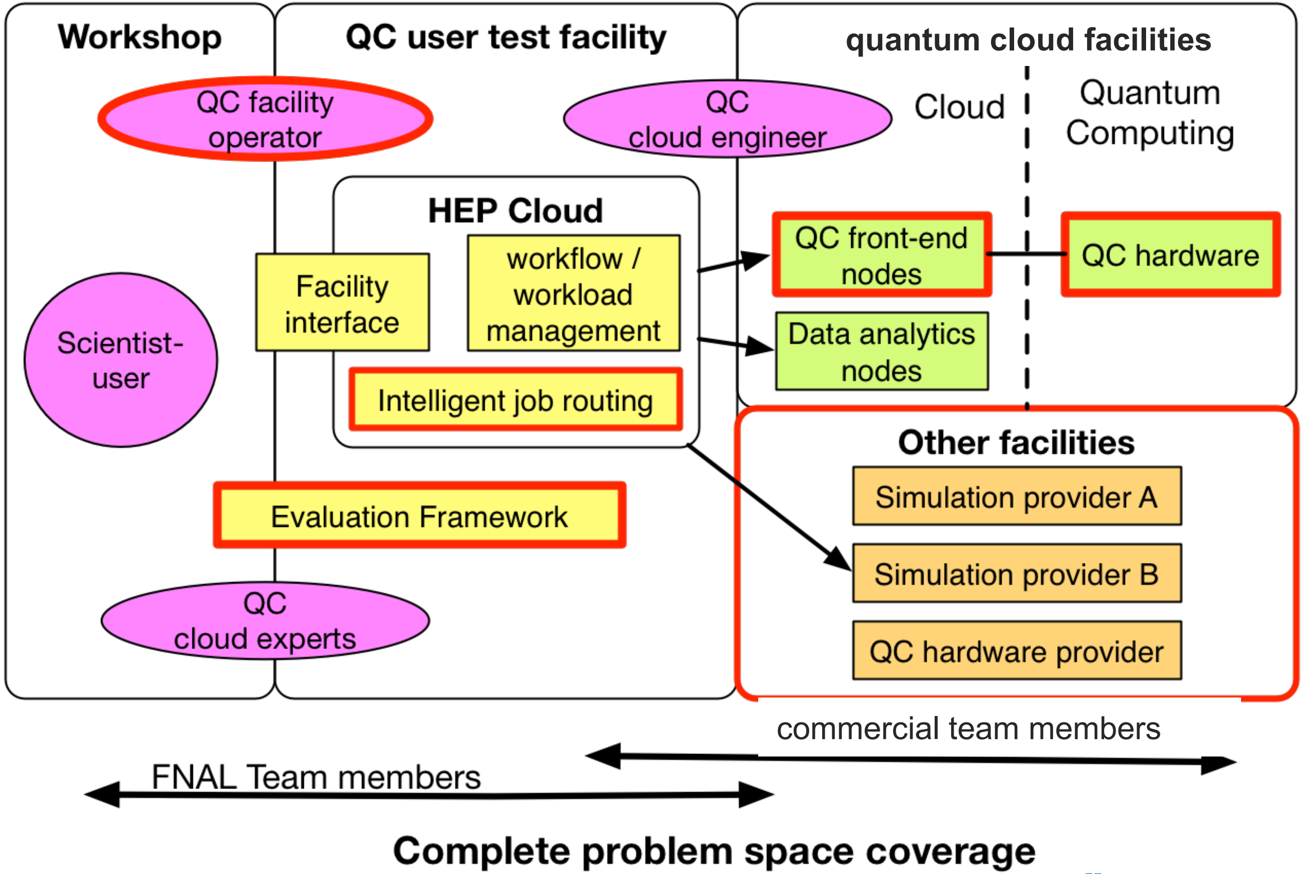
Our Proposed Plan of Work

- Host a series of workshops
  - Introduce HEP community to QC and Quantum Information Science
  - Introduce QC and Quantum Information Science community to HEP
  - Incorporate QC into our HEP user facility
  - Move forward with QC experiments that can eventually lead to algorithms useful to HEP



Establishing a Testbed

- Our HEP computing model matches commercial cloud offerings
- Excellent way to make QC resources available to HEP scientists



Quantum Testbeds for HEP

– Quantum Computing in HEP
– Quantum Testbed Plan
– Candidate Quantum Applications

Candidate HEP Quantum Applications

Quantum Computing is currently interesting for us as an accelerator

- Hybrid quantum/classical workflows

We have a few candidate quantum application areas

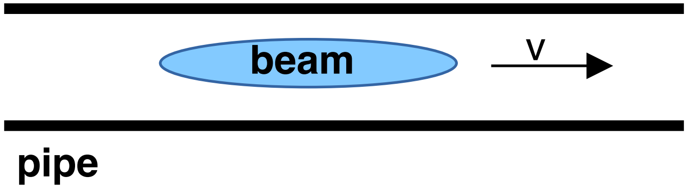
- Particle accelerator modeling utilizing PDEs
  - Poisson Equation, etc.
- Machine learning utilizing Boltzmann machines
- Optimization problems for HEP data analysis

Candidate Application Areas

- Particle accelerator modeling utilizing PDEs
- Machine learning utilizing Boltzmann machines
- Optimization problems for HEP data analysis

Particle Accelerator Modeling Utilizing PDEs

Space charge forces in accelerators



Rigid beam approximation:
electrostatic problem

$$\Delta\Phi(\vec{r}) = -f_h(\vec{r})$$

space charge force

$$\vec{F} = -q\nabla\Phi$$

Beam simulation

Particle simulation approach:

- The motion of a large number of particles is simulated.
- **F** is applied directly to the particles (momentum kicks).

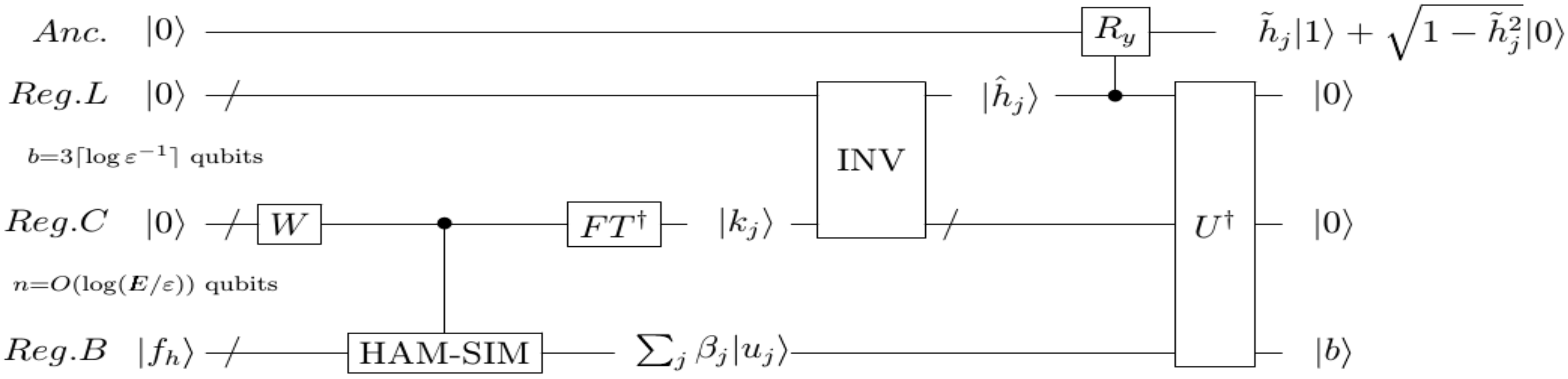
Approaches using the Vlasov equation:

$f(\vec{r}, \vec{p}, t)$ **-** particle density in the 6D phase space

$$\frac{\partial f}{\partial t} + \vec{v}\frac{\partial f}{\partial \vec{r}} + \vec{F}\frac{\partial f}{\partial \vec{p}} = 0$$

Quantum Algorithm for a Poisson Solver

Yudong Cao, *et al,* 2013, New J. Phys. 15, 013021



1. Input preparation
2. Phase estimation algorithm for the eigenvalues $\lambda_j = |k_j>$
3. Inverse eigenvalue calculation $h_j \approx C/\lambda_j$
4. Rotation of the ancilla qubit
5. Output use

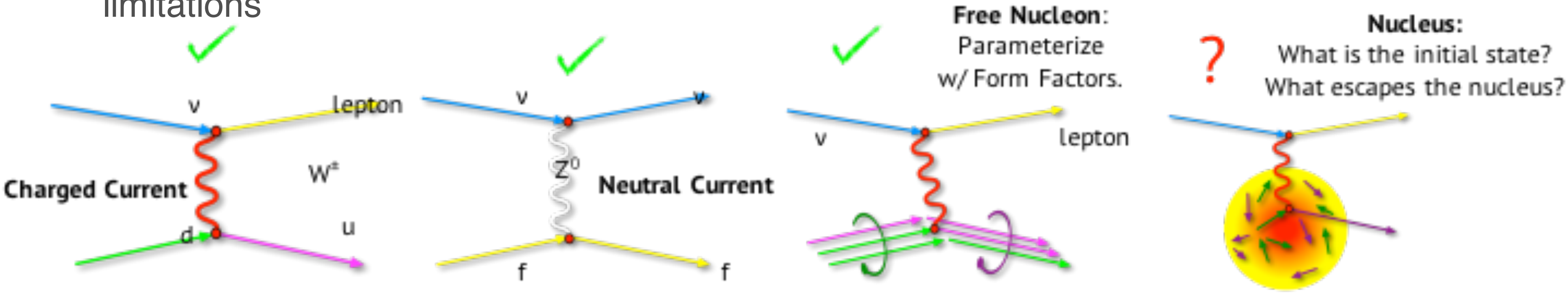
Particle Accelerator Modeling Utilizing PDEs Plan of Action

- Next Step
  - Start simple. Implement Cao's Poisson solver for small number of qubits and 1d case
  - Optimize approach in conjunction with collaborators
  - 2-d and 3-d Poisson solver

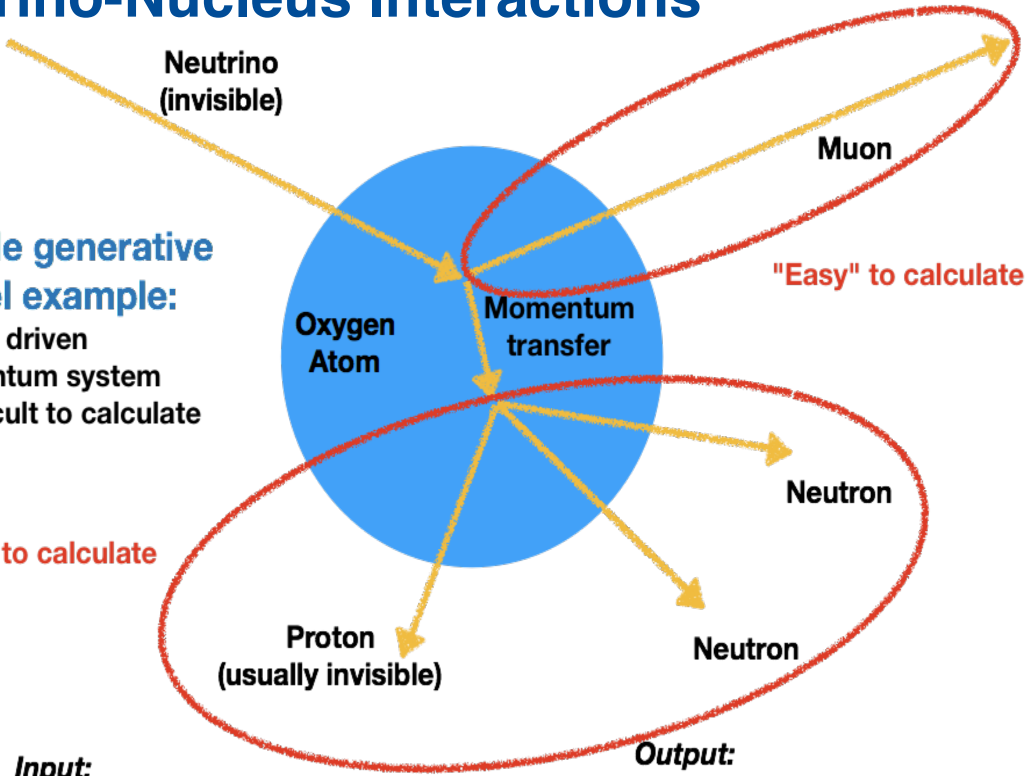
- Later
  - Implement different boundary conditions (corresponding to different pipe geometries). The Quantum Phase estimation part of the algorithm needs modifications.
  - Figure out how to use the output for beam study. It may lead to a quantum algorithm for the Vlasov equation.

Use of Simulation in HEP Analysis

- Basic workflow:
  - Establish a complete, high quality simulation system,
  - Use the simulation output to design features for an analysis,
  - Run the analysis on detector data.
- We have very detailed first-principles simulations - but, they can be slow, and often rely on models that contain incomplete physics.
  - We are interested in generative models improve simulation speed and to circumvent limitations



Simulating Neutrino-Nucleus Interactions



Simple generative model example:

* Data driven
* Quantum system
* Difficult to calculate

Input:
Momentum transfer (from muon)
(1 floating point number)

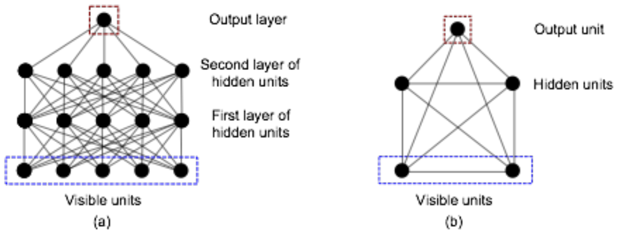
Output:
Number of neutrons
(1 integer)

BONUS: neutron directions

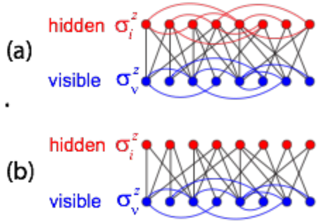
Fermilab

Boltzmann Machines

- Formally recurrent neural nets with undirected edges
- May provide a generative model for the data
- BM is modeling training data with an Ising model in thermal equilibrium
- The probability of a configuration is modeled with the Gibbs distribution $P(v,h) = e^{-E(v,h)}/Z,$
- Energy function $E(v,h) = -\sum_i v_i b_i - \sum_j h_j d_j - \sum_{i,j} w^{vh}_{ij} v_i h_j - \sum_{i,j} w^{v}_{i,j} v_i v_j - \sum_{i,j} w^{h}_{i,j} h_i h_j.$
  - System seeks the minimum energy
- The energy function is difficult to evaluate but some techniques (e.g., contrastive divergence) make it possible to estimate the gradient with only a few (or single) MCMC sampling step
  - Still very computationally expensive



arXiv 1412.3489



arXiv 1601.02036

Fermilab

Quantum Boltzmann Machines

- GEQS algorithm (arXiv 1412.3489) - Gradient Estimation via Quantum Sampling

Algorithm 3 GEQS algorithm for estimating the gradient of O_{ML}.

Input: Initial model weights w, visible biases b, hidden biases d, edge set E and κ, a set of training vectors x_{train}, a regularization term λ, and a learning rate r.
Output: Three arrays containing gradients of weights, hidden biases and visible biases: `gradMLw`, `gradMLb`, `gradMLd`.

for $i = 1 : N_{\mathrm{train}}$ **do**
  `success` $\leftarrow 0$
  while `success` $= 0$ **do**
    $|\psi\rangle \leftarrow$ `qGenModelState`(w, b, d, E, κ)
    `success` $\leftarrow$ result of measuring last qubit in $|\psi\rangle$
  end while
  `modelVUnits`$[i] \leftarrow$ result of measuring visible qubit register in $|\psi\rangle$.
  `modelHUnits`$[i] \leftarrow$ result of measuring hidden unit register in $|\psi\rangle$ using amplitude amplification.
  `success` $\leftarrow 0$
  while `success` $= 0$ **do**
    $|\psi\rangle \leftarrow$ `qGenDataState`$(w, b, d, E, \kappa, x_{\mathrm{train}}[i])$.
    `success` $\leftarrow$ result of measuring last qubit in $|\psi\rangle$ using amplitude amplification.
  end while
  `dataVUnits`$[i] \leftarrow$ result of measuring visible qubit register in $|\psi\rangle$.
  `dataHUnits`$[i] \leftarrow$ result of measuring hidden unit register in $|\psi\rangle$.
end for
for each visible unit i and hidden unit j **do**
  $\texttt{gradMLw}[i,j] \leftarrow r\left(\frac{1}{N_{\mathrm{train}}}\sum_{k=1}^{N_{\mathrm{train}}}(\texttt{dataVUnits}[k,i]\texttt{dataHUnits}[k,j] - \texttt{modelVUnits}[k,i]\texttt{modelHUnits}[k,j]) - \lambda w_{i,j}\right)$.
  $\texttt{gradMLb}[i] \leftarrow r\left(\frac{1}{N_{\mathrm{train}}}\sum_{k=1}^{N_{\mathrm{train}}}(\texttt{dataVUnits}[k,i] - \texttt{modelVUnits}[k,i])\right)$.
  $\texttt{gradMLd}[j] \leftarrow r\left(\frac{1}{N_{\mathrm{train}}}\sum_{k=1}^{N_{\mathrm{train}}}(\texttt{dataHUnits}[k,j] - \texttt{modelHUnits}[k,j])\right)$.
end for

Machine Learning Utilizing Boltzmann Machines Plan of Action

- Examine RBMs using classical computers (e.g., TensorFlow) in the context of simulation (as a generative model – "competing" with a GAN)
- Study quantum algorithm implementation
- How do we input data (here a long, simple list of floats) and extract output (here, a long list of paired integers)?
- This problem is simple but interesting
  - Obvious extensions: distinguish between prompt and delayed neutrons, get neutron energy and angle, predict the existence of pions and other particles, etc.
- Initial quantum example: data-driven neutron counting: single variable input (Q^2), output is integer number of neutrons

Fermilab

Candidate Application Areas

- Particle accelerator modeling utilizing PDEs
- Machine learning utilizing Boltzmann machines
- Optimization problems for HEP data analysis

High-dimensional Parameter Estimation

- Part of most analyses across all HEP experiments
  - Techniques such as MCMC frequently employed
  - Need for evaluation of expensive likelihood functions involving experimental results
  - Produce posterior probability distributions

$$P(H_i|D,I) = \frac{P(D|H_i,I)P(H_i|I)}{\sum_i P(D|H_i,I)P(H_i|I)}$$

$$p(\boldsymbol{d}|\boldsymbol{\theta},\boldsymbol{s},I)$$

$$p(\boldsymbol{\theta}|\boldsymbol{d},I) = \int p(\boldsymbol{\theta},\boldsymbol{s}|\boldsymbol{d},I)\,\mathrm{d}\boldsymbol{s}$$

$$p(\boldsymbol{\theta},\boldsymbol{s}|\boldsymbol{d},I)\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{s} = \frac{p(\boldsymbol{d}|\boldsymbol{\theta},\boldsymbol{s},I)p(\boldsymbol{\theta},\boldsymbol{s}|I)\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{s}}{\int p(\boldsymbol{d}|\boldsymbol{\theta},\boldsymbol{s},I)p(\boldsymbol{\theta},\boldsymbol{s}|I)\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{s}}$$

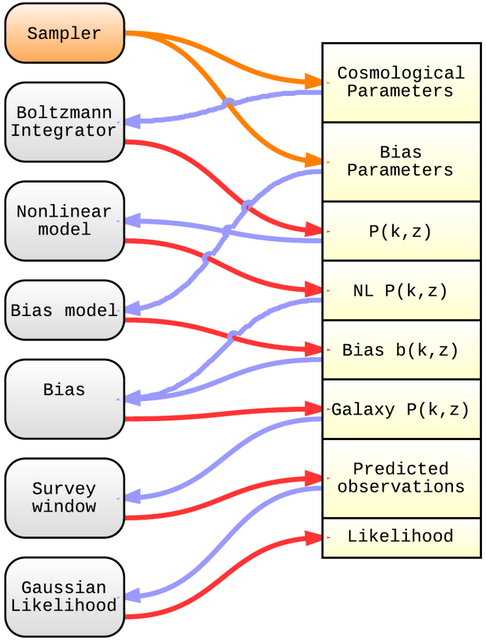
Another view: the high-dimensional parameter fitting problems can be abstracted as structured least-squares problems of the form

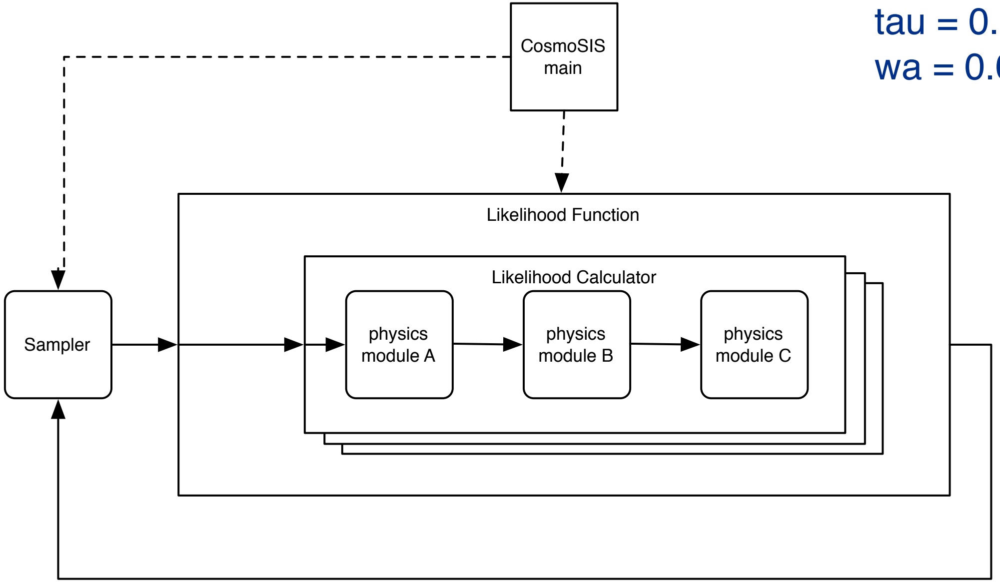
$$\min_{\lambda} \chi^2(\lambda) := \sum w_i \left(\frac{f_i(\lambda) - D_i}{\sqrt{f_i(\lambda)^2 + D_i^2}} \right)^2$$

Fitting as a Part of Current Analysis Tools

- CosmoSIS example – a modular framework for parameter estimation
  - MCMC module is typically used as the sampler
  - Allows for combining likelihoods

```
# Example configuration:
[cosmological_parameters]
omega_m = 0.2 0.3 0.4
h0 = 0.6 0.7 0.8
omega_b = 0.02 0.04 0.06
omega_k = 0.0 w=-1.0
A_s = 2.0e-9 2.1e-9 2.3e-9
n_s = 0.92 0.96 1.0
tau = 0.08
wa = 0.0
```





Fermilab

Optimization Problems for HEP Data Analysis Plan of Action

- Starting point: experiment with known algorithms
- Sampling
  - Gibbs, perhaps Metropolis-Hasting
  - Still trying to understand if these can actually be used
- Optimization – QAOA and Constraint Satisfaction Problems
  - MaxCut
  - SAT (Binary Satisfaction Problems)
  - Still not known
- Reading through papers from Farhi and Harrow, and many others

General Observations

- The ideas presented are only starting points
  - We expect further research to take us in new directions
- There is a great temptation to base quantum computing ideas on today's classical computations
  - Probably exactly the wrong approach.
- Physics models that are intractable on classical computers could be newly interesting on quantum computers
- Input (state preparation) and output are important areas for study

The End

Thank you for your attention