QGP parameter extraction via a global analysis of event-by-event flow coefficient distributions

Jonah E. Bernhard[1] Christopher E. Coleman-Smith[1]
Peter W. Marcy[2] Steffen A. Bass[1]

[1]Department of Physics
Duke University

[2]Department of Statistics
University of Wyoming

20 July 2013

Model to data comparison

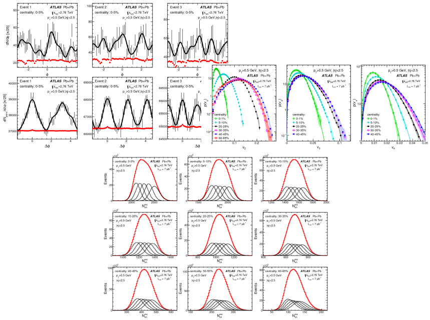
Input parameters
η/s, ...

Model
initial conditions, hydro, sampler, UrQMD

Output (observables)
flows, spectra, ...

?

Experimental data



Model to data comparison

The generic recipe:

- » Choose a set of input parameters.
- » Vary parameters, calculate observables.
- » Determine parameters that optimally describe reality.

Easy if the model is fast:

- » Use e.g. MCMC to find the optimal parameters.
- » Requires many points in parameter space, $\mathcal{O}(10^6)$ or more.
- » Only feasible if the model runs in $\mathcal{O}(1 \text{ second})$.

Heavy-ion collision models are *not* "fast".

- » Need a different approach.

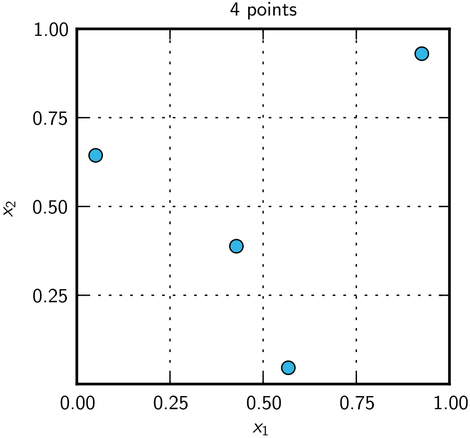
Slow models

Strategy: emulate the model.

- Run at predetermined set of parameter points.
  - Latin-hypercube sample.
- Interpolate between points.
  - Emulator.

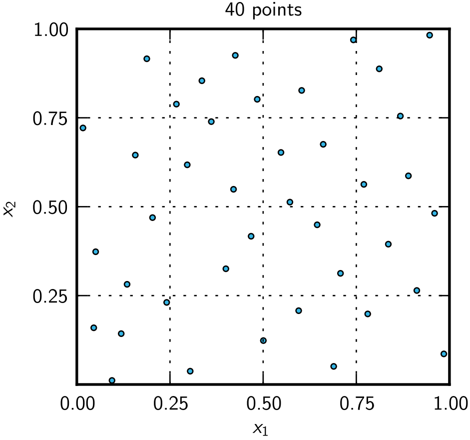
Latin-hypercube sampling

- Provides an optimal set of parameter points.
- *Maximizes* the *minimum* distance between points.



Latin-hypercube sampling

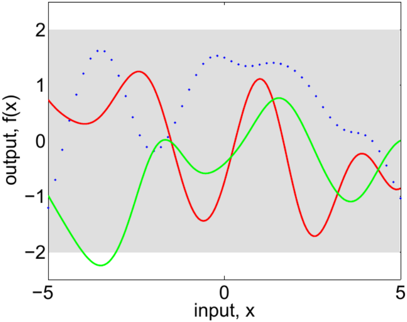
- Provides an optimal set of parameter points.
- *Maximizes* the *minimum* distance between points.



Gaussian processes

- *Assume* the model is a Gaussian process.
- A Gaussian *process* is a generalization of a Gaussian *distribution*.
  - Draw a set of Gaussian values with a specified covariance.

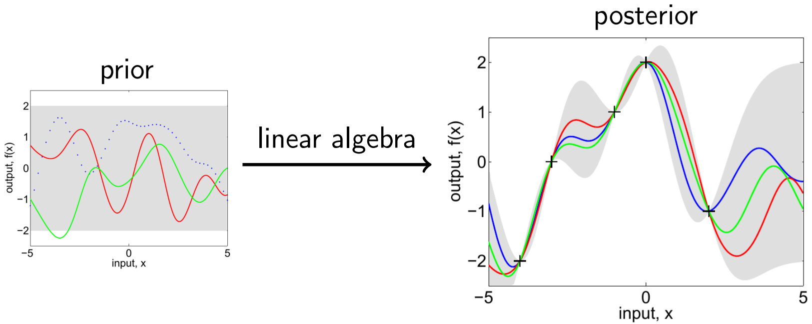
$$\mathrm{cov}(x_1, x_2) \propto \exp\left[-\frac{(x_1 - x_2)^2}{2\ell^2}\right]$$



Gaussian Processes for Machine Learning, Rasmussen and Williams, 2006.

Gaussian process emulators

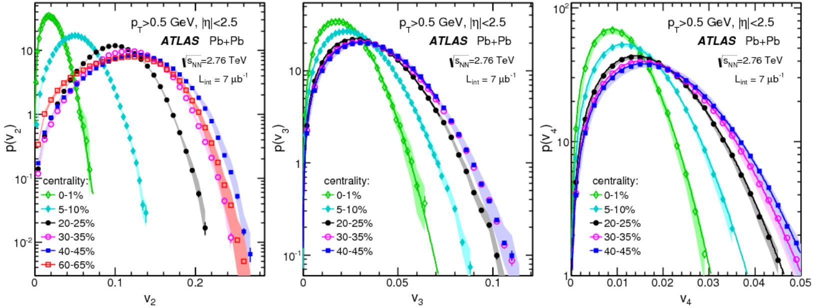
- Prior: the model is a Gaussian process.
- Posterior: Gaussian process conditioned on model outputs.



- Emulator is a fast surrogate to the actual model.
  - More certain near calculated points.
  - Less certain in gaps.

The data

- ATLAS event-by-event flow distributions v_n, $n = 2$–6.
- Could provide a much more sensitive probe than average flows
  - especially high-order ($n > 3$).

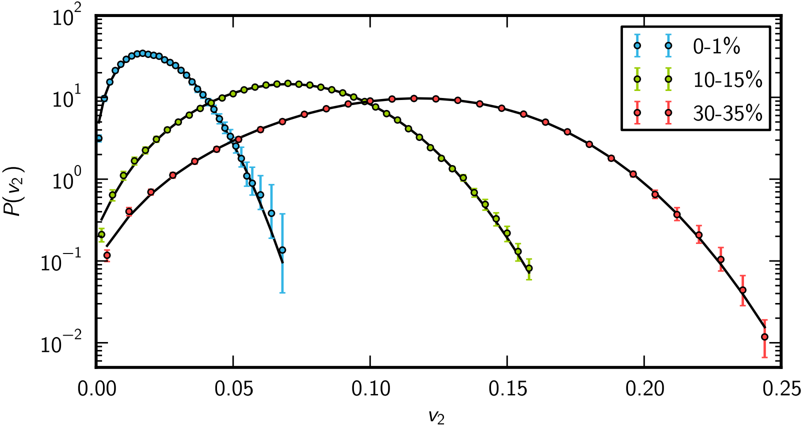


hep-ex/1305.2942

Data reduction

- Fit to "Generalized Reverse Weibull" distribution.
  - Represent each distribution by four fit parameters.

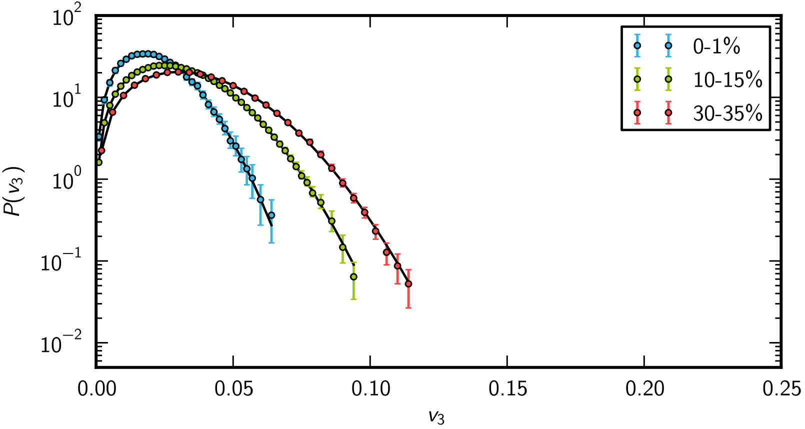
$$f(x; m, s, \alpha, \gamma) = \frac{\alpha}{s\,\Gamma(\gamma)} \left(\frac{x - m}{s} \right)^{\alpha\gamma - 1} \exp\left[-\left(\frac{x - m}{s} \right)^{\alpha} \right]$$



Data reduction

- Fit to "Generalized Reverse Weibull" distribution.
  - Represent each distribution by four fit parameters.

$$f(x; m, s, \alpha, \gamma) = \frac{\alpha}{s\,\Gamma(\gamma)} \left(\frac{x-m}{s}\right)^{\alpha\gamma-1} \exp\left[-\left(\frac{x-m}{s}\right)^{\alpha}\right]$$



The model

Modern version of the OSU+Duke hybrid model VISHNU (Viscous Hydro and UrQMD):

» MC-Glauber/KLN initial conditions
» 2+1 viscous hydro (OSU)
» Cooper-Frye hypersurface sampler (OSU)
» UrQMD

Similar to OSU iEBE, but organized differently; different analysis.

Main goal

Calibrate the event-by-event model to ATLAS flow distributions.

Experiment design

- 256 Latin-hypercube points across 5 parameters:
  - IC normalization
  - IC-specific parameter (wounded nucleon / binary collision for Glauber, saturation exponent for KLN)
  - hydro start time τ_0
  - viscosity η/s
  - shear stress relaxation time τ_Π
- Observables:
  - v_n distributions
  - multiplicities
  - identified particle spectra
  - ...

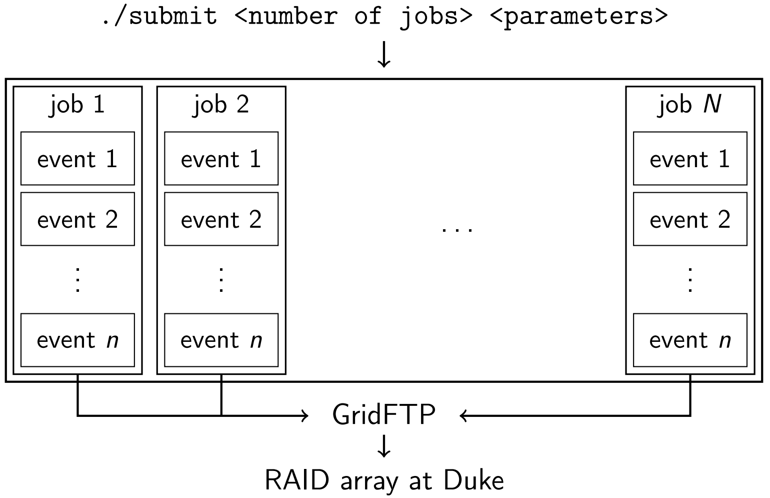
CPU time

- 3 centrality bins
- 256 points/bin
- 1000 events/point
- ~ 1 hour/event

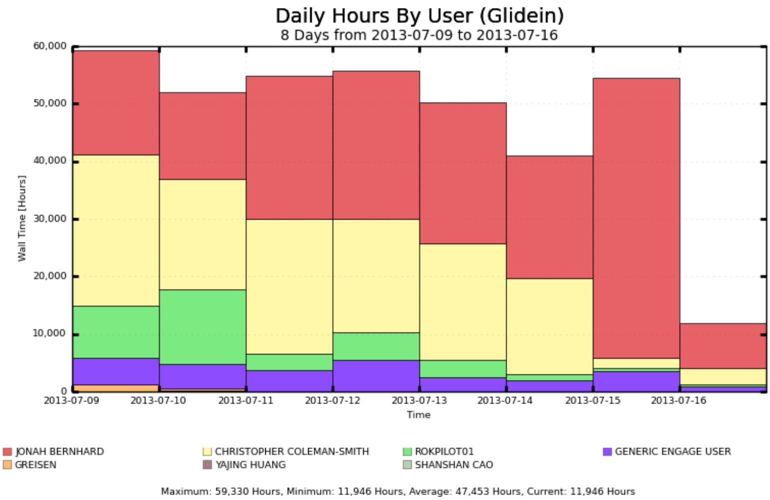
$$\sim 768\,000 \text{ CPU hours} \sim 87 \text{ years}$$

- Open Science Grid (OSG)

EbE-OSG: job automation



github.com/jbernhard/ebe-osg



» roughly 500 000 events complete

EbE analysis

- Python + Numpy/Scipy
- parses event files
- calculates flows & fits flow distributions
- calculates other observables
- makes common plots (Matplotlib)

Planned:

- store results in database via ORM
- optimization via custom C++ extentions

`github.com/jbernhard/ebe-analysis`

Preliminary results

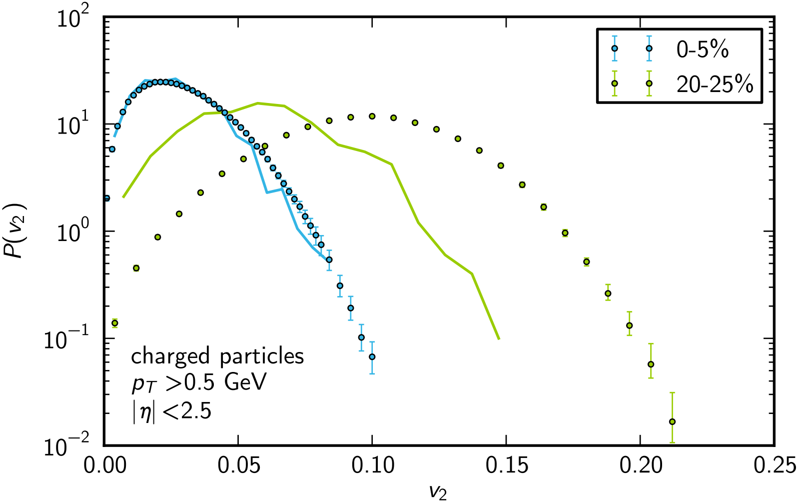
analysis of uncalibrated Latin-hypercube sample point

two centrality bins, 0–5% and 20–25%
1000 events each

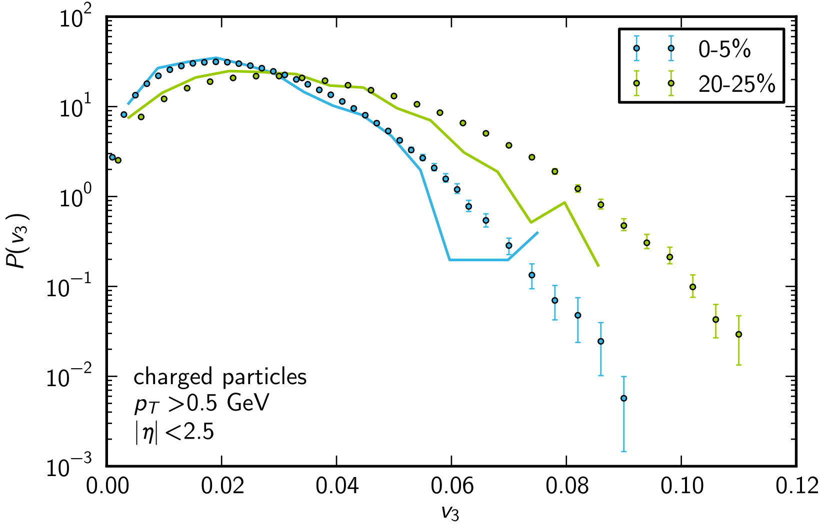
MC-Glauber with $\alpha = 0.06$

$\eta/s = 0.29 \qquad \tau_0 = 0.93$ fm

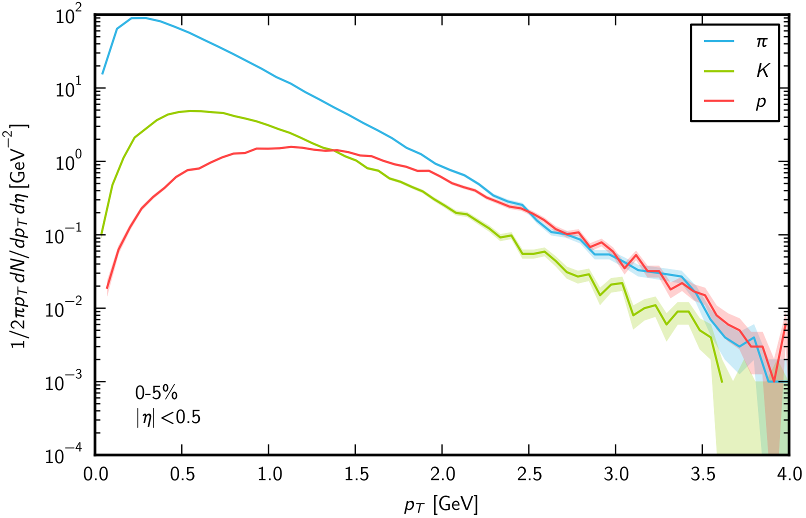
v_2 distributions



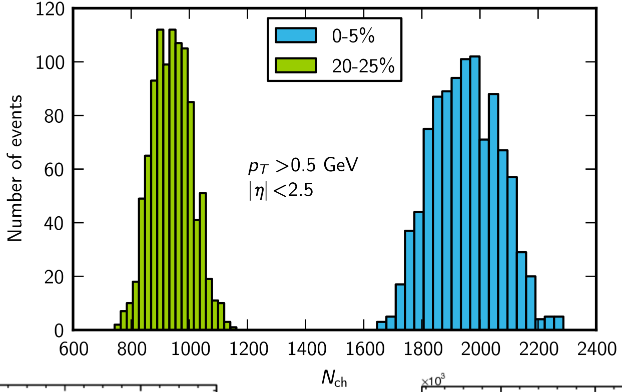
v_3 distributions

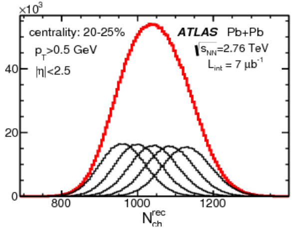


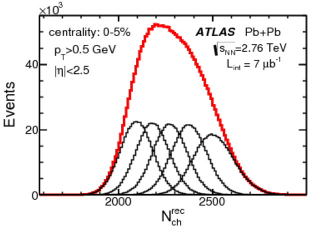
Identified particle spectra



Multiplicity







Goals

- Higher-order flows v_4, v_5, v_6.
- More realistic model (IP-Glasma, 3+1D, ...).
- Systematically analyze all events.
- Train emulator to calculated flow distributions, identified particles, etc.
- Calibrate parameters to data.
  - Which are the most important?
- Improve statistics for likely parameters.