Fusion Plasma Reconstruction

Google Applied Sciences

Presented by: Ian Langmore

Work is joint with:
Ted Baltz, Michael Dikovsky, Scott Geraedts, Nathan Neibauer, Peter. Norgaard, Rob von Behren, John Platt

TAE and the Plasma Debugger

Please interrupt me and ask questions!

Google -- TAE Partnership

Goal:

Accelerate development of viable fusion energy

tae TECHNOLOGIES

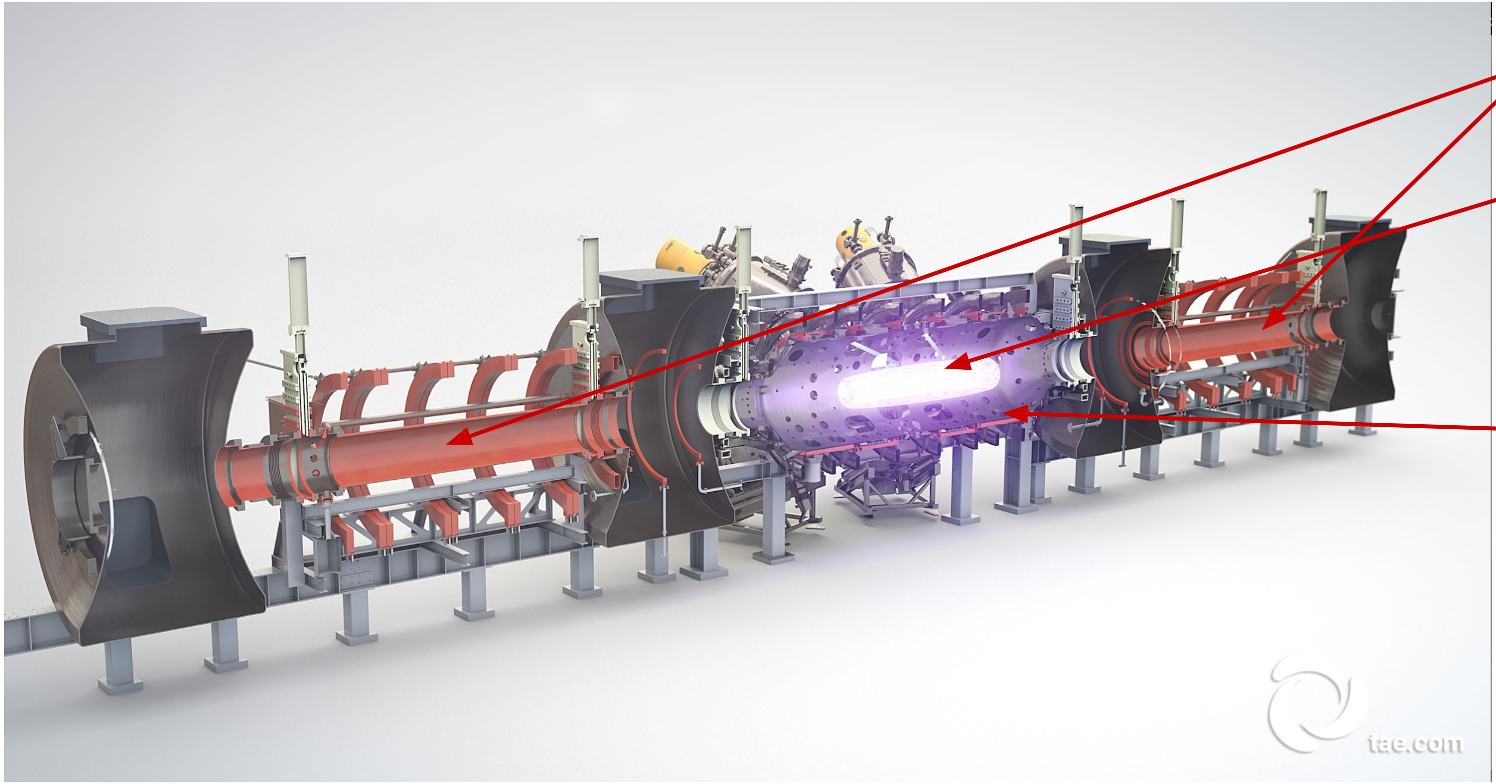
- Commercial fusion energy company
- Southern California
- TAE personnel on project
    - M. Binderbauer, D. Ewing, A. Smirnov
    - E. Trask, H. Gota, R. Mendoza, J. Romero, S. Dettrick

Google!

Google (Applied Sciences)

- Commercial web-search company
- Northern California
- Google personale on project (order of joining)
    - R. Koningstein, J. Platt
    - T. Baltz, M. Dikovsky, I. Langmore, T. Madams, P. Norgaard, Y. Carmon, N. Neibauer, R. von Behren, S. Geraedts

Norman: Experimental FRC Plasma Generator



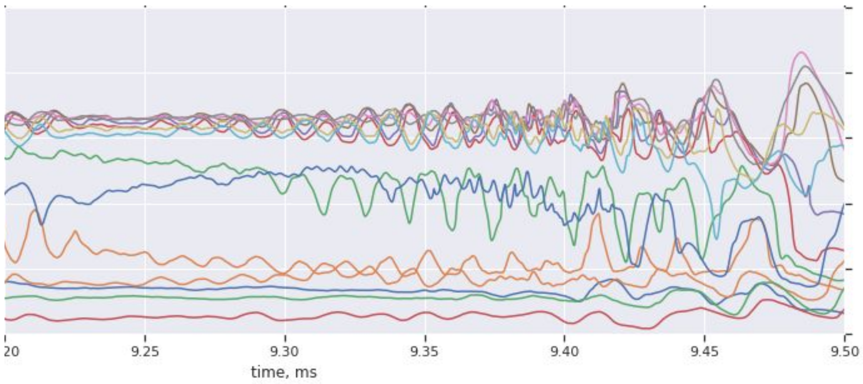
Plasma formed on each end, then fired into center vessel

Plasma confined by magnetic fields, heated/stabilized by neutral beams

Ports provide access for measurement devices

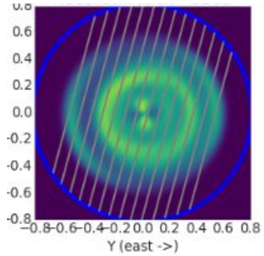
Goal: Learn to confine plasma long enough, at high enough temperatures, en route to net positive energy (in later machine)

Measurements in → Reconstructed plasma out

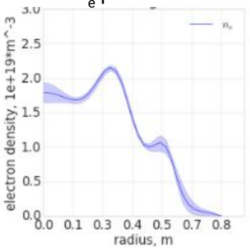


Interferometer traces

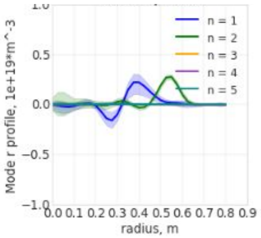
Cross section



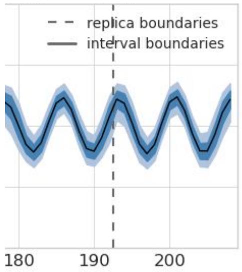
N_e profile



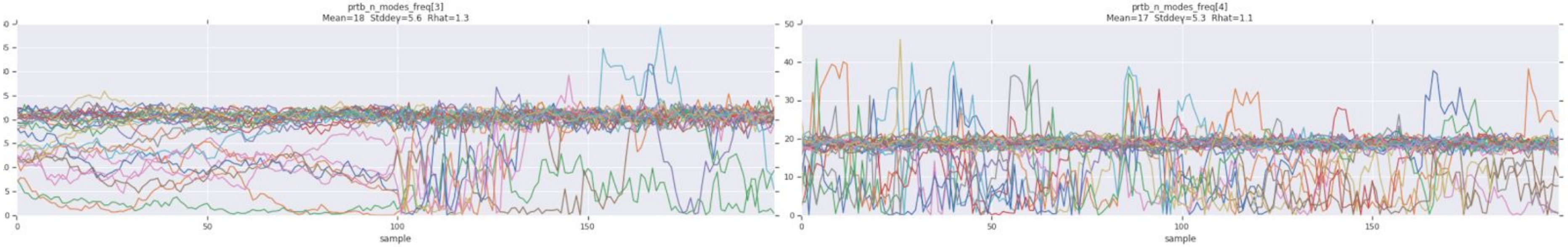
Mode profiles



Mode N=3 vs time

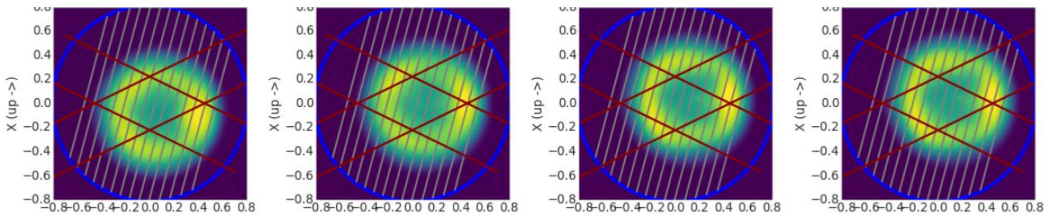


Reconstructions are Samples of Plasmas



Samples of random variables

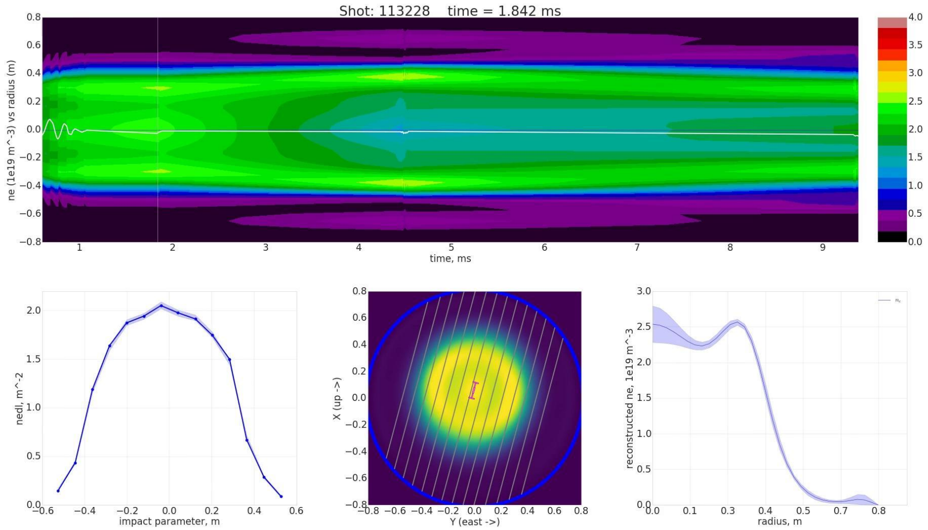
Map to samples of plasmas



Resolving Plasma Properties: The Center

Location parallel to lasers is *not well resolved* by Interferometer alone

Coupled SEE helps to resolve this

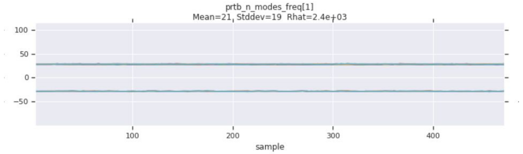


Blue dots are samples from posterior over plasma center

Resolving Plasma Properties: Mode Rotation

Rotation direction is *not resolved* by Interferometer alone

Coupled magnetic probes help to resolve this



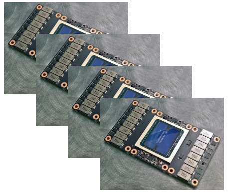
Bimodal: ± 28 kHz

Traces from Markov Chain

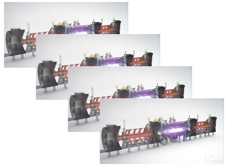
Unimodal: -28 kHz

Computation is *highly* parallelized

5000+ GPUs



1000+ experiments



On each GPU

Plasma density etc… represented by <u>many</u> 4-D Tensors, each of shape

`[n_samples, n_chains, n_times, n_events]`

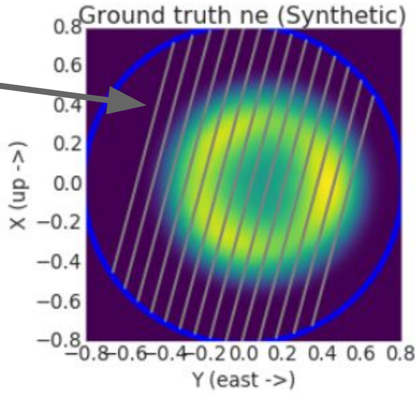
- `n_samples`: Number of samples from each chain
- `n_chains`: Number of Markov chains run in parallel
- `n_times`: Number of times handled by each GPU
- `n_events`: Dimension of each random variable

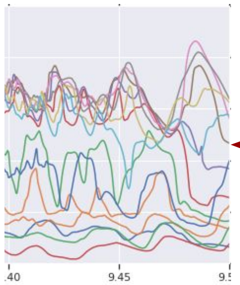
Example shape: `[450, 30, 40, 16]`

Some Bayesian Modeling Details

Modeling the Interferometer

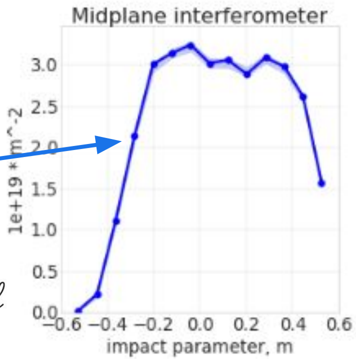
14 Lasers pass through plasma





Measure phase shift

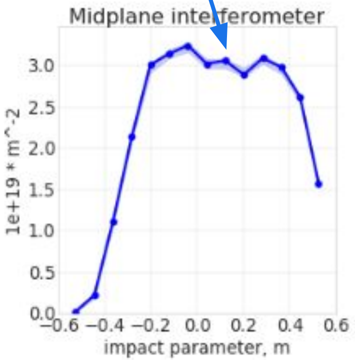
Phase shift translated to integrated density



$$\Delta\text{Phase}_i = C\int_{-1}^{1} N_e(x_i + \ell v)\,d\ell$$

Interferometer Forward Model

Phase shift translated to integrated density



Ideally, with N_e^{true} the *actual* plasma density,

$$\Delta\text{Phase}_i = C \int_{-1}^{1} N_e^{true}(x_i + \ell v) d\ell$$

We model density as $N_e = N_e(\xi)$, for $\xi \sim \mathcal{N}(0, I)$.

N_e is instantiated along the lines of integration only.

Our *Forward Model* for (phase) measurement $m = (m_1, \ldots, m_{14})$ is

$$m = AN_e + \sqrt{\sigma_{const}^2 + \sigma_{prop}^2 AN_e} \cdot \epsilon, \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, I)$$

$$(AN_e)_i \approx C \int_{-1}^{1} N_e(x_i + \ell v) d\ell$$

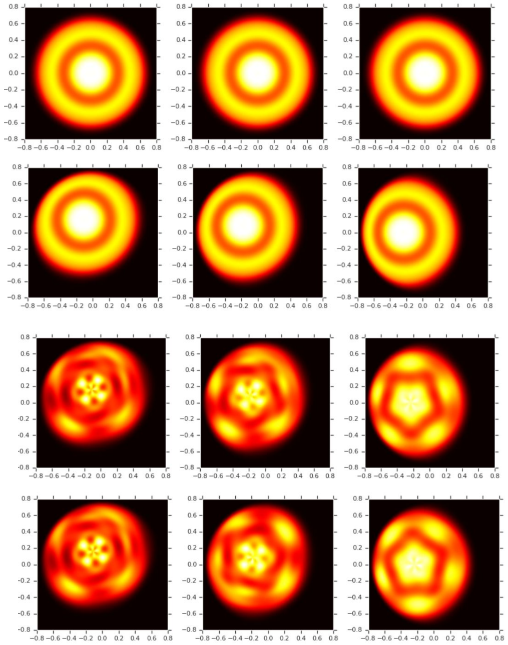
Model for Electron Density (N_e)

Our Prior is over these

$$N_e(r,\theta) = \log\left[1 + \exp\left\{\sum_{k=1}^{K} \xi_k u_k(r)\right\}\right], \quad \text{where} \quad \sum_{k=1}^{\infty} u_k(r) u_k(r') \to \exp\left\{-\frac{|r-r'|^2}{2(0.15)^2}\right\}, \quad \text{and} \quad \xi_k \sim \mathcal{N}(0,(0.1)^2)$$

$$(r\cos\theta, r\sin\theta) \mapsto (r\cos\theta - \delta_x, r\sin\theta - \delta_y), \quad \text{where} \quad \delta_x, \delta_y \sim \mathcal{N}(0,(0.1)^2)$$

$$N_e(r,\theta) \mapsto N_e(r,\theta)\left[1 + \text{Bound}_{(-1,1)}\left(\sum_{n=1}^{N} \eta_n \sin(n\theta)\right)\right], \quad \text{where} \quad \eta_n \sim \mathcal{N}(0, 1/n^2).$$



Now turn every random variable into a random process in time...

...plus a whole lot more...probably should have modeled some of the nuisance variables as noise instead.

Graphical Representation

Could have been lumped together with a more complex prior

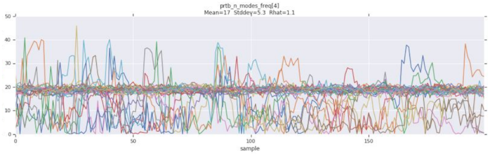
Latent
Prior = iid
Normal(0, 1)

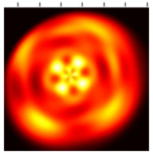
Transformations

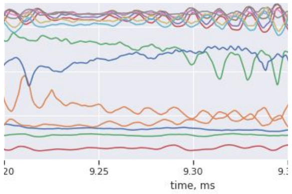
Plasma

Forward Model

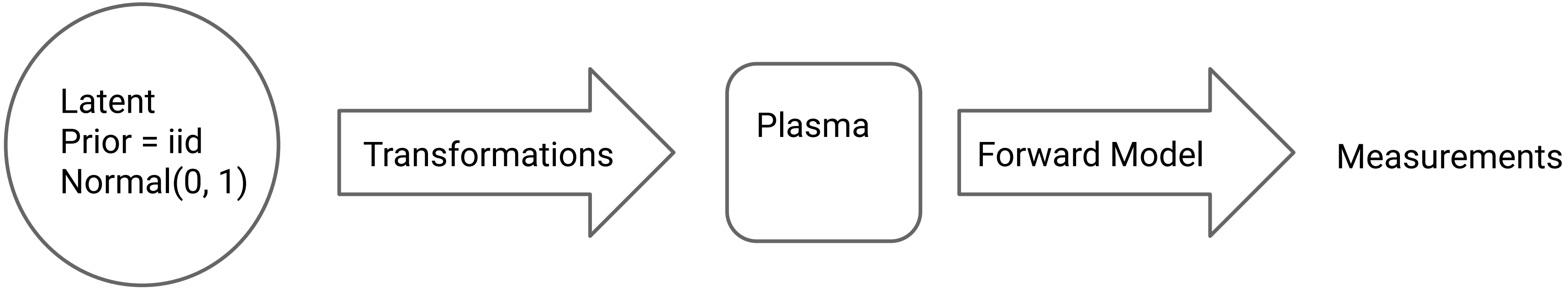
Measurements







Inverse Problems != {Generative Models, Stats, ML}



Unlike typical Generative Modeling

We care *only* about the latent (equivalently Plasma).

⇒ The Forward Model must be near perfect!

Unlike typical Statistical Modeling

For each latent, we have *exactly one* (multidimensional) measurement

⇒ Posterior predictive evaluation is impossible

Unlike typical Machine Learning

We have no golden data

⇒ Opportunities to learn are limited

See also [go/ip-not-gm](go/ip-not-gm)

Inference

MAP Estimates

$$z_{MAP} := \arg\max_z p(z)p(m \mid z) = \arg\max_z p(z \mid m)$$

MAP estimates:

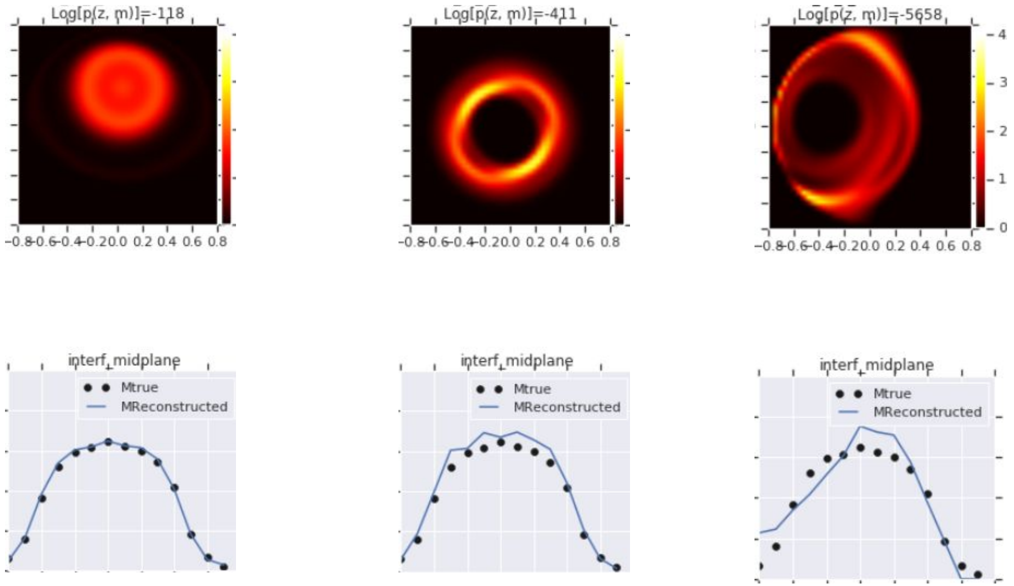
Attempt at a "best guess"

Warning:

Often finds "bad random modes"

⇒ Compute 30 estimates in parallel

(good use of TFP batch dimension capabilities)



Variational Inference : Could not make it work

Start with parameterized distribution $q(z;\phi)$. Then set

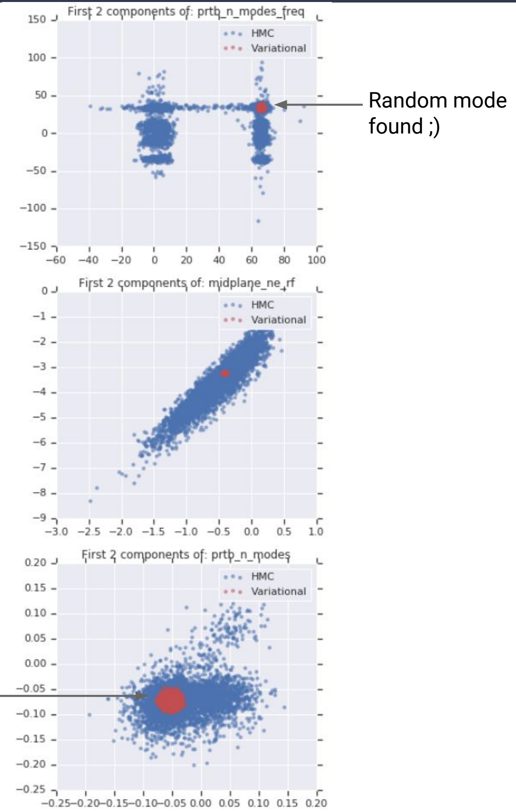
$$\phi^* := \arg\min_{\phi} \int \log\left[\frac{q(z;\phi)}{p(z,m)}\right] q(z;\phi) dz$$
$$= \arg\min_{\phi} KL\left[q(z;\phi)||p(z\,|\,m)\right].$$

- $KL[q||p] = 0 \Leftrightarrow q = p$
- Above loss function has a stable numerical approximation

- In most problems, there is no ϕ such that $q = p$
- Often under-estimates uncertainty

VI is “scared” of putting mass outside the extent of p(z)

Every time a compromise must be made, q(z) will error in this manner.

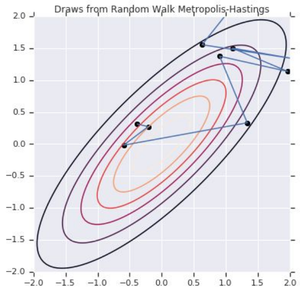


Sampling: Random-Walk Metropolis Hastings

Metropolis Hastings recipe to sample from $p(z)$

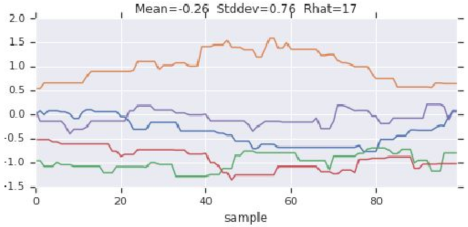
1. Initialize $z = z^0$
2. Propose a move $z \rightarrow y \sim q(y|z)$
3. Accept with probability $\min\left\{1, \frac{q(z|y)p(y)}{q(y|z)p(z)}\right\}$
    1. If Accept, set $z^1 = y$
    2. If Reject, set $z^1 = z^0$
4. Iterate...

Random Walk Metropolis-Hastings if $q(y|z) \sim \mathcal{N}(y; z, \sigma^2 I)$ is Gaussian



Random Walk behavior $\Rightarrow$ slowly mixing chains in higher dimensions
O(N) steps per "uncorrelated" sample

5 chains sampling a 50-dimensional Gaussian:
Pictured is the first component



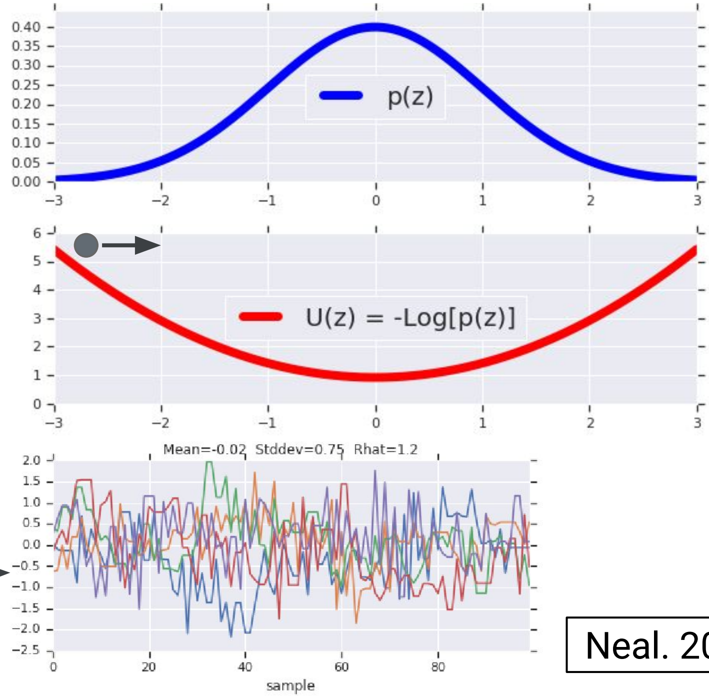
HMC: A proposal that scales well

Physics explanation

1. Let $U(z)$:= $-Log[p(z)]$ define a surface in R^N
2. Start a ball at z^0
3. Give the ball a Gaussian "kick"
4. Let the ball roll for time T, giving you the proposal

If well tuned, the "rolling" allows the proposal to travel a long distance.

$O(N^{1/4})$ steps per "uncorrelated" sample



Neal. 2012

HMC: A proposal that scales well

Increase dimension $\mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ by adding "momentum" ζ, and then...

1. Initialize $(z, \zeta) = (z^0, \zeta^0)$, where $\zeta^0 \sim \mathcal{N}(0, I)$
2. Define $H(z, \zeta) := -\log p(z) + \|\zeta\|^2/2$
3. Propose $(z(T), \zeta(T))$, the time-T (numerical) solution to the initial value problem:

$$\dot{z}(t) = \frac{\partial H}{\partial \zeta}, \quad z(0) = z^0,$$

$$\dot{\zeta}(t) = -\frac{\partial H}{\partial z}, \quad \zeta(0) = \zeta^0,$$

4. Accept with probability

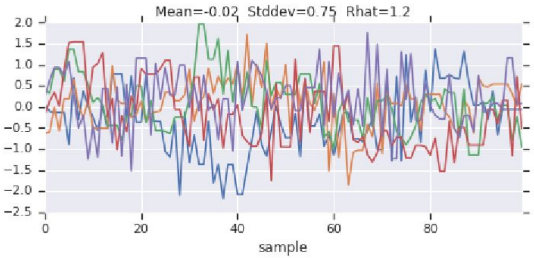
$$\min\left\{1, \exp\{H(z^0, \zeta^0) - H(z(T), \zeta(T))\}\right\}$$

If numerical integration were perfect, you would accept *every time*

This produces samples $[(z^0, \zeta^0), \ldots, (z^K, \zeta^k)]$ from $p(z, \zeta) \propto \exp\{-H(z, \zeta)\}$

The samples $(\zeta^0, \ldots, \zeta^K)$ may be discarded.

The samples $(z^0, \ldots, z^K)$ are from $p(z)$.



Neal. 2012

HMC Efficiency Tradeoff

Smaller numerical integration step size ⇒

- Lower integration error
- Higher Prob[Accept]

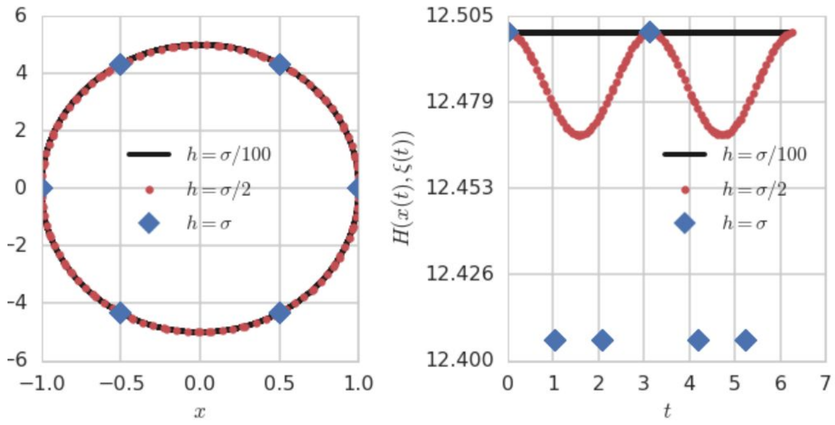
But also…

- number of steps needed

  ~ O(1 / step_size)

Rough best practice:

1. Adjust step_size until P[Accept] ≈ 0.68
2. Adjust num_leapfrog_steps

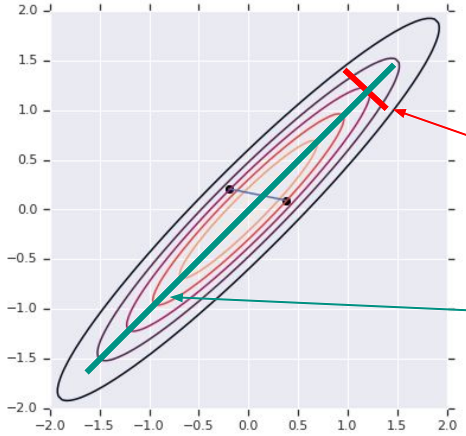


Integration error due to finite step size

Beskos. 2010

What are the optimal parameters?

Level sets of p(z) in 2-D



What are the optimal step size h^* and number of integration steps ℓ^*?

Assume...

- Gaussian target $p(z)$
- Eig(Covariance[Z]) is $\sigma_1^2 \geq \cdots \geq \sigma_N^2 > 0$

Then

- Must have stable integration along smallest scales
  $\Rightarrow h^* < 2\sigma_N$
- $T = h^* \ell^*$ must be large enough to traverse largest scales
  $\Rightarrow h^* \ell^* = c\sigma_1$

$\Rightarrow \ell^* > \frac{c}{2} \frac{\sigma_1}{\sigma_N}$

Suggested efficiency measure

The correct asymptotics are

$$h^* \propto \left(\sum_{i=1}^{n} \frac{1}{\sigma_i^4} \right)^{-1/4}, \qquad \boxed{\ell^* \propto \kappa := \left(\sum_{i=1}^{n} \frac{\sigma_1^4}{\sigma_i^4} \right)^{1/4}}$$

L. 2019

Kappa : Approximation valid for most spectra

$$\ell^* \propto \kappa := \left(\sum_{i=1}^{n} \frac{\sigma_1^4}{\sigma_i^4} \right)^{1/4}$$

The approximation is valid for spectra with "heavy enough" tails.

For example: Discrete low-pass filters with squared spectra

$$\sigma_k^2 = \frac{1}{1 + k^{2p}}, \quad k = 1, \ldots, n.$$

Or repeated blocks (c.f. Beskos 2010)

$$C = \mathrm{Diag}(B, \ldots, B_{\lfloor n/k \rfloor}), \quad B \in R^{k \times k}$$

Or discretizations of continuous operators, e.g.

$$(\Delta + \epsilon I) f(x) \to \begin{pmatrix} 2+\epsilon & -1 & 0 & \cdots & & \\ -1 & 2+\epsilon & -1 & 0 & \cdots & \\ 0 & -1 & 2+\epsilon & -1 & 0 & \cdots \\ \cdots & & & & & \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \cdots \end{pmatrix}$$

Fun Facts about Kappa

$$\ell^* \propto \kappa := \left(\sum_{i=1}^{n} \frac{\sigma_1^4}{\sigma_i^4} \right)^{1/4}$$

As n increases, we expect κ to increase as $\sim O(n^{1/4})$ (c.f. Beskos 2010).

The proof

- involves a Normal limit of integration error terms
- requires (in addition to Normality) σ_i decay slower than exponentially

  $\Rightarrow$ many terms contribute, and a CLT applies

κ can be written in terms of spectral and Schatten-4 norms

$$\kappa = \|L\|_2 \|L^{-1}\|_{S^4}, \quad \text{where} \quad \text{Covariance} = LL^T.$$

$\Rightarrow$ can be re-written in terms of traces

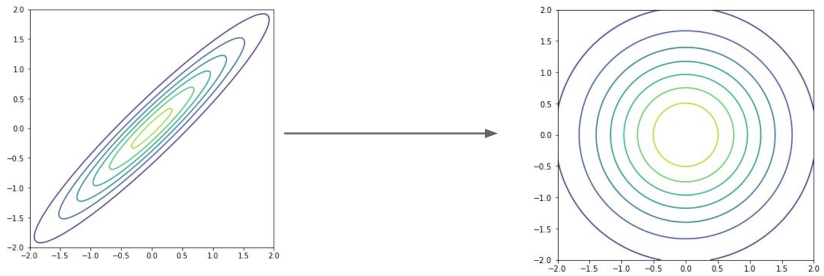
Preconditioning

$$\text{Suppose Covariance}(Z) = E[ZZ^T] = C = LL^T$$

Change the random variable you are sampling

$$Z \longrightarrow L^{-1}Z$$

This alters the shape of the PDF



...and minimizes κ

$$\kappa = \left(\frac{\sigma_1^4}{\sigma_1^4} + \frac{\sigma_1^4}{\sigma_2^4} + \cdots + \frac{\sigma_1^4}{\sigma_N^4} \right)^{1/4} \longrightarrow (1 + 1 + \cdots + 1)^{1/4} = N^{1/4}$$

Preconditioning

Practicalities

- You don’t know the covariance, so you must estimate it using...
  - samples
  - variational inference
  - `tf.hessians`
- Have to hope nonlinearities don’t mess things up!
- Nonlinear preconditioning methods exist, but are very very tricky

Hoffman. 2019

Parno. 2017

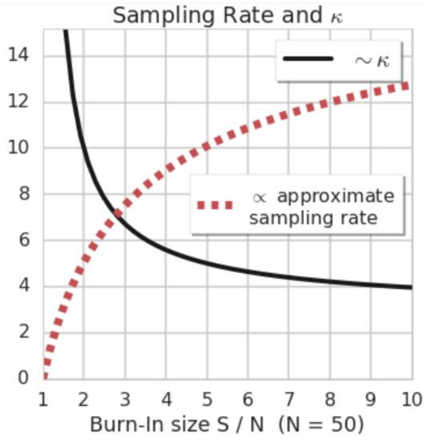
Connection to Random Matrix Theory

Lemma 5.1. *Suppose* $(X^1, \ldots, X^S)$ *are i.i.d., and HMC sampling of* $X \sim \mathcal{N}(0, C)$ *is preconditioned with the* S*-sample Cholesky factor* $\hat{\mathrm{L}}$*. Then the preconditioned* κ *follows the law of* $\kappa(C)$*, for* $C \sim InverseWishart(S, N)$*.*

Since the spectrum of large Wishart matrices follow the Marcenko-Pastur Law,

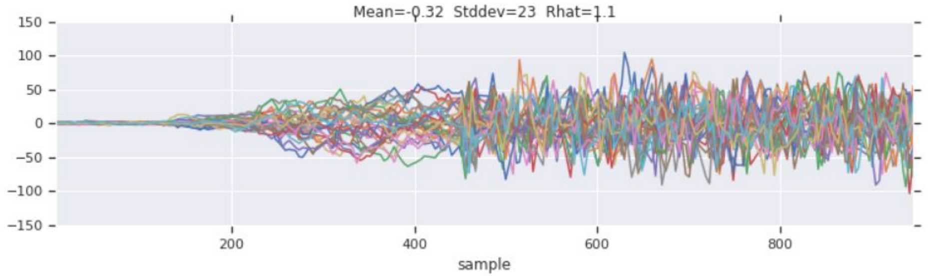
we have:

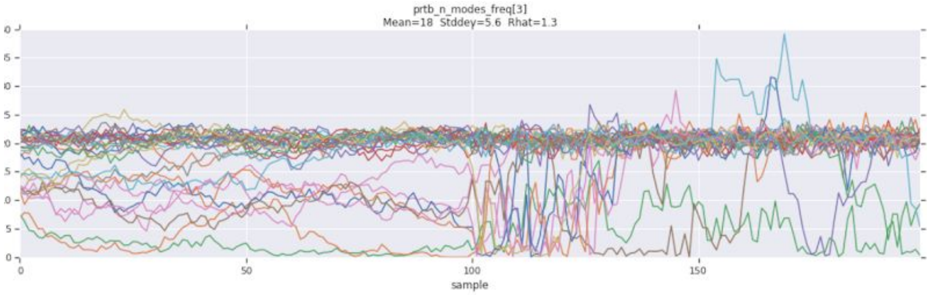
$$\kappa \approx N^{1/4} \frac{\left(1 + \frac{N}{S}\right)^{1/4}}{1 - \sqrt{\frac{N}{S}}}$$



Sampling Strategy : Iterative improvement

1. Find preconditioner L via Variational Inference
2. Use `tfp.mcmc.SimpleStepSizeAdaptation` to adapt h until P[Accept] ≈ 0.9
3. Draw ~ 25 samples from 30 parallel chains
4. Update L → Diag(Stddev(Z_{sample}))
   a. adapt step size again
5. Draw ~ 25 more samples
6. Update L → ?? Depending on estimated change in Kappa
   a. adapt step size again
7. Continue, until Rhat is small enough





Set L → Diag(Stddev)

Evaluation of Bayesian Reconstructions

First: Let's be realistic

Do we really sample from the "posterior"?

- Our prior is "reasonable", but is it really the marginal distribution over all possible plasmas?
  - hahahahhahahaha
- We model many effects, but plasmas are complex beasts and we do not model all
- We only have one measurement, of much smaller dimension than our unknowns.
- We *never* sample from the tails
  - takes too long to get samples
  - *by definition* you can't really validate them
- Will we ever know we're right about anything?
  - we have *zero* golden data

Responsible hypothesis generation

Debugger Commandments:

1. If a ~~human~~ physicist can infer interesting event X is likely from the raw data, so too shall the debugger
2. If two events, X and Y are both somewhat likely, the debugger shall indicate thus
3. The debugger shalt not send TAE on too many wild goose chases for effects it has *hallucinated*
4. The degree to which we achieve 1-3 shalt be exhaustively tested using synthetic data

Synthetic Plasma

No "ground truth" solution exists for plasma dynamics (can't solve for 10^{20} particles + Maxwell's equations).

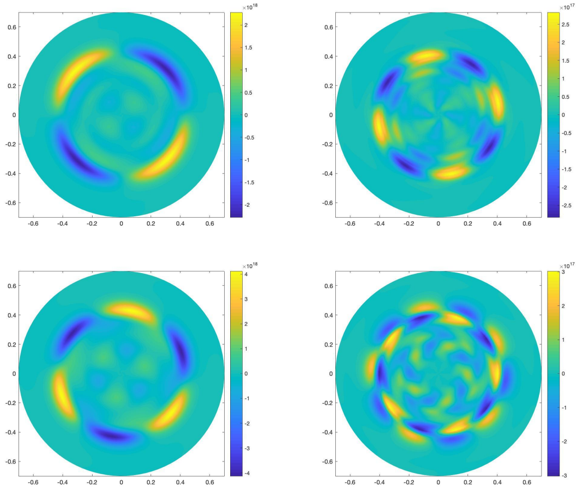
Approximate solutions from fluid/particle simulation can still be used to test the inference algorithm.
A physicist combines and modifies certain features from simulation data to make a "synthetic plasma".

Examples:
- Check for false positive / false negative of feature identification
- Evaluate impact of 3d effects on 2d reconstruction
- Investigate cases with statistical ambiguity

"What is the smallest density fluctuation that can be reconstructed?"

"Can the model identify both fast and slow feature dynamics?

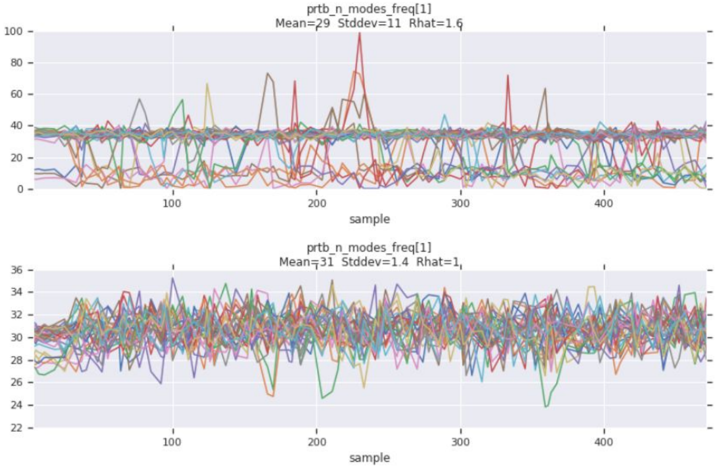
"How does aliasing present with high-frequency behaviors?"



MCMC Diagnostic : Rhat

Running parallel Markov Chains...

- makes efficient use of GPUs
- allows for the convergence diagnostic R-hat
  - tfp.mcmc.potential_scale_reduction



$$\hat{R} := \frac{\text{WithinChainVariance} + \text{BetweenChainVariance}}{\text{WithinChainVariance}}$$

$$= \frac{N_c^{-1} \sum_{n=1}^{N_c} \sigma_n^2 + (N_c - 1)^{-1} \sum_{n=1}^{N_c} (\mu_n - \mu)^2}{N_c^{-1} \sum_{n=1}^{N_c} \sigma_n^2}$$

If chains are mixing well

$$\hat{R} \to 1, \quad \text{as} \quad N_c \to \infty.$$

Generally, $\hat{R} < 1.1$ is "good enough" for us.

Thank You!

Contact: langmore@google.com

Beskos et al. 2010: Optimal tuning of the Hybrid Monte-Carlo algorithm (link)

Betancourt. 2018: A conceptual introduction to Hamiltonian Monte Carlo (link)

Hoffman et al. 2019: NeuTra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport (link)

Langmore et al. 2019: A condition number for Hamiltonian Monte Carlo (link)

Neal. 2012: MCMC Using Hamiltonian dynamics (link)

Parno, Marzouk. 2017: Transport map accelerated Markov chain Monte Carlo (link)

Verma et al. 2015: Large scale cluster management at Google with Borg (link)

Citations

Supplementary Material