UQ update: Fitting Models to Langmuir Probe Data

Khachik Sargsyan[1], Tiernan Casey[1], Habib Najm[1], Timothy Younkin[2]

[1]Sandia National Laboratories, Livermore, CA
[2]Oak Ridge National Laboratory, Oak Ridge, TN

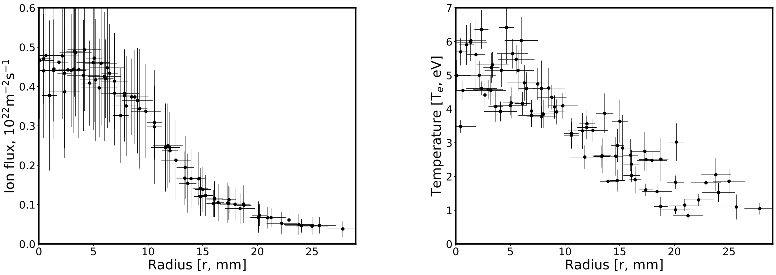
PSI2 Meeting
May 23, 2018

Outline

- Langmuir probe data and initial UQ goal
- Fitting parametric model to data
- Bayesian viewpoint
  - Noise assumptions
  - Markov chain Monte Carlo
  - Model selection
- Some results
  - Basis choice, zero-derivative constraint
  - Error-in-variable models
  - Moment-matching likelihood
- Summary and work-in-progress

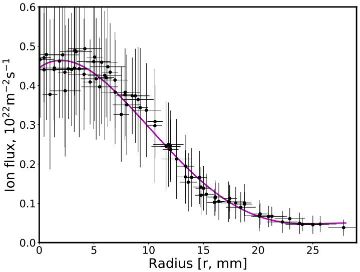
PISCES-A He(+1) Plasma Profiles Measured By Reciprocating Langmuir Probe

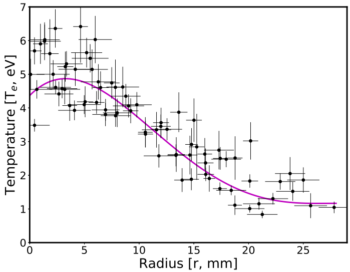
- Probe data consists of 5 probe shots (or plunges)
- Each point is a measurement (no averaging)
- Horizontal error bars: uncertainty in position during plunge
- Vertical error bars: fitting uncertainty



PISCES-A He(+1) Plasma Profiles Measured By Reciprocating Langmuir Probe

- Probe data consists of 5 probe shots (or plunges)
- Each point is a measurement (no averaging)
- Horizontal error bars: uncertainty in position during plunge
- Vertical error bars: fitting uncertainty

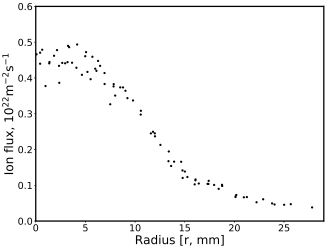




Build uncertain representation (a.k.a. joint PDF) of the fit to feed forward model (GITR, Xolotl)

Fitting parametric model to data: least squares

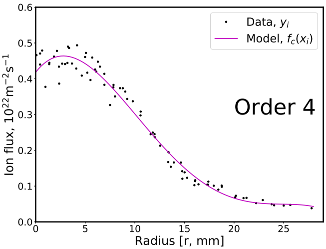
- Given data (x_i, y_i) for $i = 1, \ldots, N$



Fitting parametric model to data: least squares

- Given data (x_i, y_i) for $i = 1, \ldots, N$
- Given parametrized model form $f_c(x)$
- Tune c, such that $y_i \approx f_c(x_i)$
- Least-squares

$$\underset{c}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - f_c(x_i))^2$$

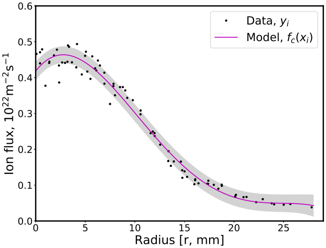


- Linear parametrization (basis expansion)... $$f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$$
- ... allows analytical answer $c = (P^T P)^{-1} P^T y$, where $P_{ik} = \Psi_k(x_i)$

Fitting parametric model to data: least squares

- Given data (x_i, y_i) for $i = 1, \ldots, N$
- Given parametrized model form $f_c(x)$
- Tune c, such that $y_i \approx f_c(x_i)$
- Least-squares

$$\underset{c}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - f_c(x_i))^2$$



- Linear parametrization (basis expansion)... $$f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$$
- ... allows analytical answer $c = (P^T P)^{-1} P^T y$, where $P_{ik} = \Psi_k(x_i)$
- ... with covariance information $\Sigma_c \propto (P^T P)^{-1}$

Bayesian viewpoint of fitting $y_i \approx f_c(x_i)$

- Bayes' formula $$p(\mathcal{M}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M})p(\mathcal{M})}{p(\mathcal{D})}$$
- Data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$
- Model $\mathcal{M} \equiv c$
- Rewrite Bayes' formula

$$\overbrace{p(c|y)}^{\text{Posterior}} = \frac{\overbrace{p(y|c)}^{\text{Likelihood}} \quad \overbrace{p(c)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}}$$

- Prior $p(c)$: expert knowledge, or uninformative
- Posterior $p(c|y)$: updated 'knowledge' of c, given data y
- Likelihood $L(c) = p(y|c)$: key, noise/error model, encapsulates assumptions about data collection
- Evidence $p(y)$: not important for parameter (coeff. c) estimation; crucial for model selection (e.g. poly order)

Bayesian least squares $\equiv$ Gaussian noise assumption

- Gaussian likelihood:

$$L(c) = p(y|c) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f_c(x_i))^2}{2\sigma^2}\right)$$

- Data noise size σ either given by data expert, or inferred with c as a *hyperparameter*
- For linear models: $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$, we have analytically available Gaussian posterior, with mean $\mu_c = (P^T P)^{-1} P^T y$ and $\Sigma_c = \sigma^2 (P^T P)^{-1}$, exactly as in deterministic least-squares
- This simple formulation highlights importance of noise assumption:

Least-squares assumes Gaussian i.i.d. noise with constant st. dev.

Posterior sampling via Markov chain Monte Carlo (MCMC)

$$\overbrace{p(c|y)}^{\text{Posterior}} = \frac{\overbrace{p(y|c)}^{\text{Likelihood}} \quad \overbrace{p(c)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}}$$

- In general, when model is not linear or noise is not Gaussian, there is little alternative to MCMC
- MCMC is a search procedure in parameter space leading to a stochastic process with a stationary distribution $p(c|y)$
- Given samples from posterior, one can interrogate it further
  - Estimate PDF with KDE
  - Compute moments
  - Build functional representation, such as PC
  - Pipe it to the next model as an input

Model selection via Bayes Factor

$$\overbrace{p(c|y)}^{\text{Posterior}} = \frac{\overbrace{p(y|c)}^{\text{Likelihood}} \quad \overbrace{p(c)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}}$$

- MCMC only requires posterior evaluation up to proportionality constant, $p(c|y) \propto p(y|c)p(c)$
- Evidence $p(y)$ is not important for parameter estimation
- Evidence is marginal likelihood (i.e. likelihood integrated w.r.t. prior)

$$p(y|M) = \int p(y|c)p(c)dc$$

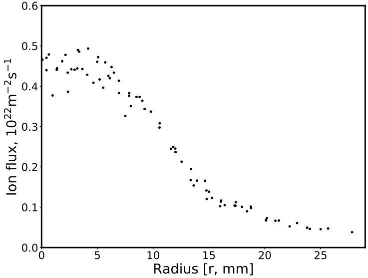
- It is crucial for model selection via Bayes factors

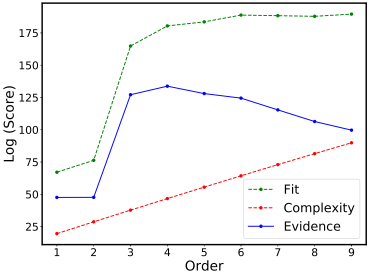
$$\text{BF}(M_1, M_2) = \frac{p(y|M_1)}{p(y|M_2)}$$

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



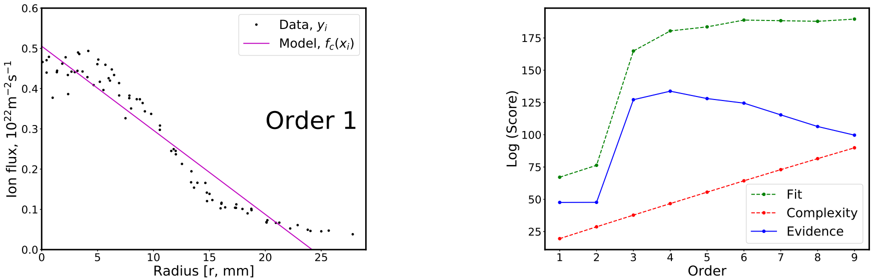


- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)

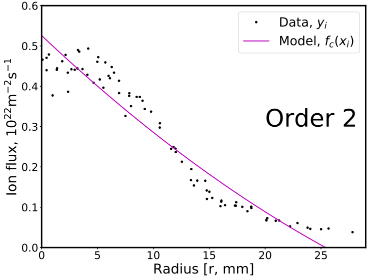


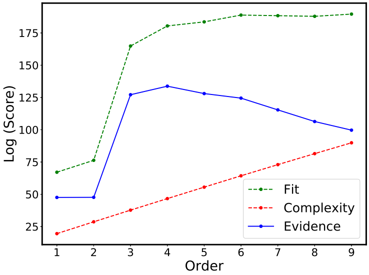
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



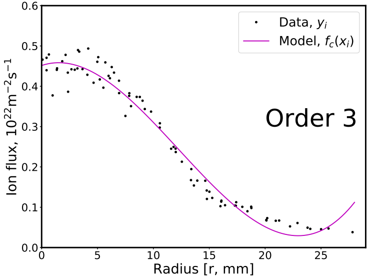


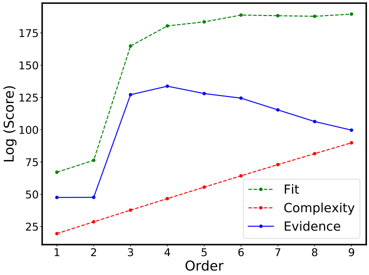
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



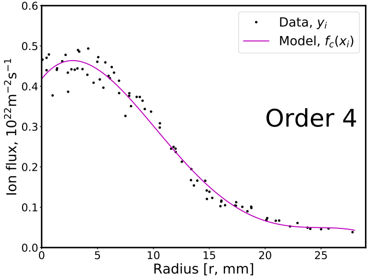


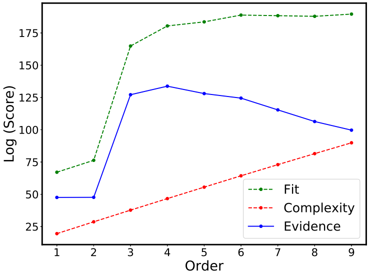
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



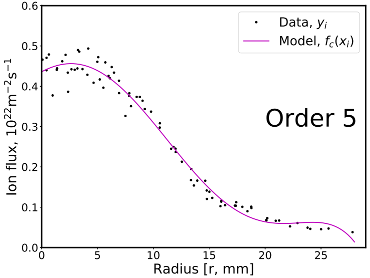


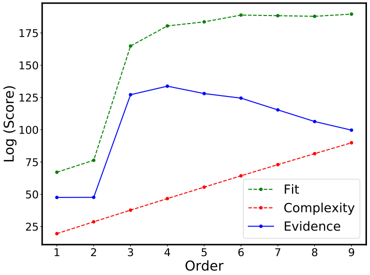
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



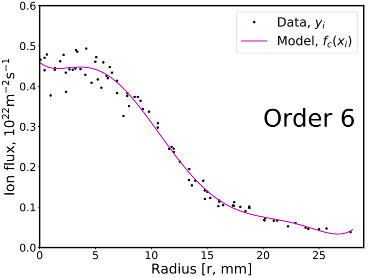


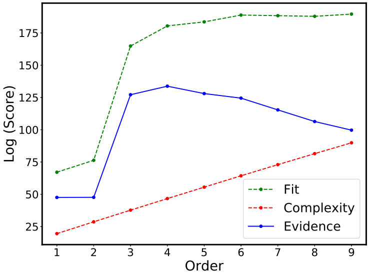
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



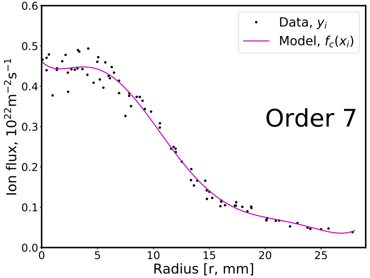


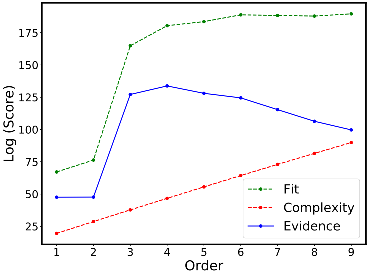
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



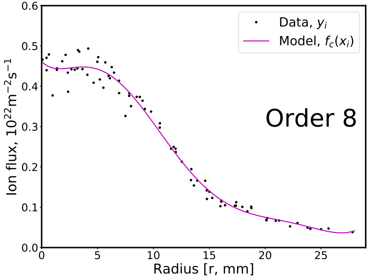


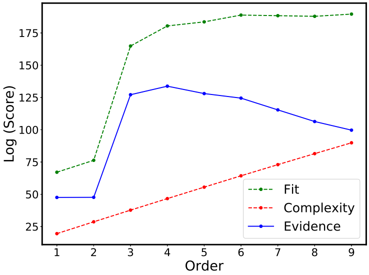
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

log(Evidence) = log(Fit) - log(Complexity)



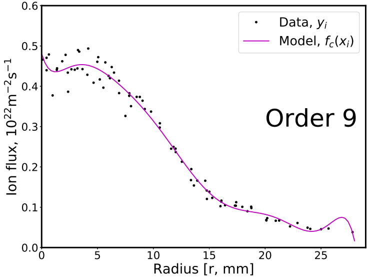


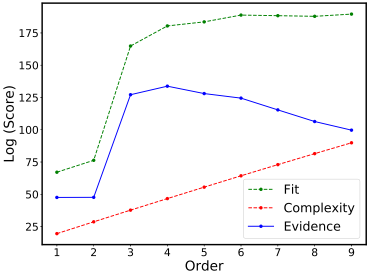
- Caveat: evidence is often difficult to compute

Poly order as alternative models: BF$(M_p, M_q) = \frac{p(y|M_p)}{p(y|M_q)}$

- Evidence $p(y|M_K)$ for K-th order model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$
- Encapsulates Occam's razor or the "law of parsimony"

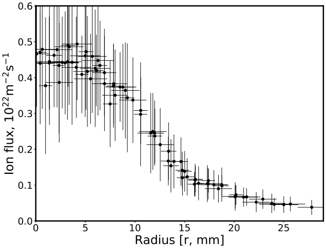
log(Evidence) = log(Fit) - log(Complexity)

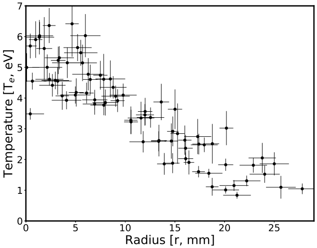




- Caveat: evidence is often difficult to compute

Back to Langmuir probe data

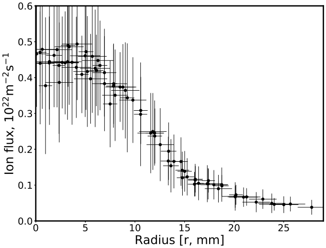


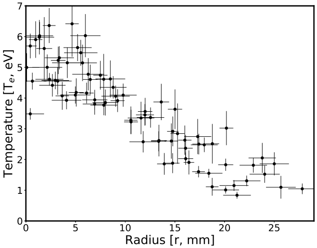


Three paths:

- Ignore correlations for now and fit individual QoIs independently
  - Done. See next few slides.
- Get the raw measurements behind this data, use (hierarchical) Bayesian inference with raw data
  - Formulation nearly ready. Some questions remain.
- In case raw data is not available, employ maximum-entropy methods to propose hypothetical underlying data sets
  - Not needed yet.

Back to Langmuir probe data

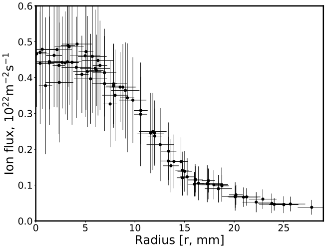


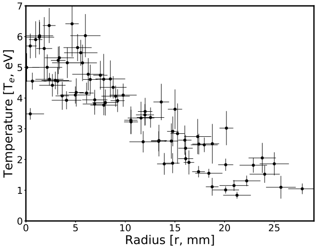


Three paths:

- Ignore correlations for now and fit individual QoIs independently
  - Done. See next few slides.
- Get the raw measurements behind this data, use (hierarchical) Bayesian inference with raw data
  - Formulation nearly ready. Some questions remain.
- In case raw data is not available, employ maximum-entropy methods to propose hypothetical underlying data sets
  - Not needed yet.

Back to Langmuir probe data





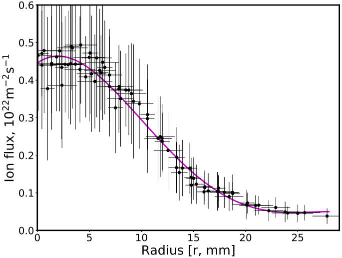
Three paths:

- Ignore correlations for now and fit individual QoIs independently
  - Done. See next few slides.
- Get the raw measurements behind this data, use (hierarchical) Bayesian inference with raw data
  - Formulation nearly ready. Some questions remain.
- In case raw data is not available, employ maximum-entropy methods to propose hypothetical underlying data sets
  - Not needed yet.

Independent modeling of fitted data

A few improvements first: recall the model $f_c(x) = \sum_{k=0}^{K} c_k \Psi_k(x)$

- Basis choice: use Legendre polynomials (orthogonal on $[-1, 1]$) instead of monomials (1, x, x^2, x^3, ...)

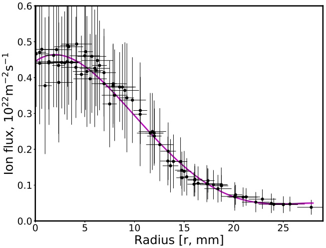


$$\Psi_0(x) = 1$$
$$\Psi_1(x) = x$$
$$\Psi_2(x) = (3x^2 - 1)/2$$
$$\Psi_3(x) = (5x^3 - 3x)/2$$

*orthogonality makes coeff. inference better conditioned

- Scale input from $r \in [0, 29]$ to $x \in [-1, 1]$, essentially arriving at scaled Legendre polynomials $L_k(r) = \Psi_k(x)$
- Zero-derivative on one end: the highest-order coefficient is completely determined by the lower-order ones
- Positivity constraint: work with logarithms (not impl. yet)

Error-in-variable model [perhaps outdated]



- True $\tilde{x}_i$ is 'hidden' behind observed x_i
- ξ_i is uniform, η_i is normal

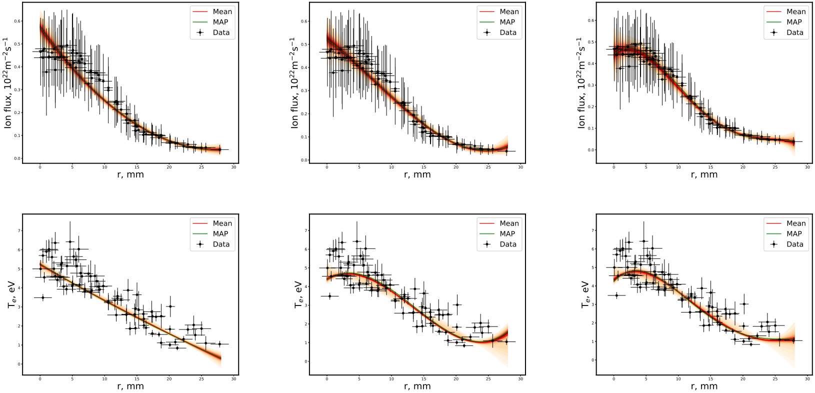
$$\begin{cases} x_i = \tilde{x}_i + \sigma_i^x \xi_i, \\ y_i = f_c(\tilde{x}_i) + \sigma_i^y \eta_i. \end{cases}$$

- Option 1: infer c only
  - Need uncertainty propagation for likelihood construction
  - Use Polynomial Chaos (story for another day)
- Option 2: infer c and $\tilde{x}$
  - Pseudo-marginal MCMC

$$\begin{aligned} p(c,\tilde{x}|\mathcal{D}) \propto \quad & \propto \; p(\mathcal{D}|c,\tilde{x}) \quad p(c)p(\tilde{x}) \\ & = \; p(y|x,\tilde{x},c)p(x|\tilde{x},c) \quad p(c)p(\tilde{x}) \\ & = \; p(y|\tilde{x},c)p(x|\tilde{x}) \quad p(c)p(\tilde{x}) \\ & \propto \; p(c|y,\tilde{x})p(y|\tilde{x})p(\tilde{x}|x) \\ & \propto \; p(c|y,\tilde{x})p(\tilde{x}|x,y) \end{aligned}$$

Error-in-variable model [perhaps outdated]

Order 2 | Order 3 | Order 4



But our assumptions were wrong (see next slide)!

Modeling noise is critical

- Turns out the vertical errorbars are not data noise, but are a result of a fitting process
- We need to produce polynomial models that are representative of given vertical errorbars
- Horizontal errorbars are not ‘measurement’ errors either!

In lieu of raw data, need to be careful about
the errorbars and noise assumptions

Moment/PDF matching noise model

- Lift the model from deterministic to stochastic

$$\begin{aligned} f_c(x) &= c_0 + c_1\Psi_1(x) + c_2\Psi_2(x) + c_3\Psi_3(x) + \\ &+ \left[d_0 + d_1\Psi_1(x) + d_2\Psi_2(x) + d_3\Psi_3(x)\right]\xi \end{aligned}$$

- Zero-derivative constraint $c_3 = l(c_0, c_1, c_2), d_3 = l(d_0, d_1, d_2)$
- Object of inference $c = (c_0, c_1, c_2, d_0, d_1, d_2)$

- Match moments, or better,
  Kullback-Leibler divergence between model and data

$$KL(p_1||p_2) = \int \log\left(\frac{p_1}{p_2}\right) dp_1 \overset{\text{Gauss.}}{=} \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

- Use approximate likelihood $\log L(c) = -KL(p_y||p_f) - KL(p_f||p_y)$

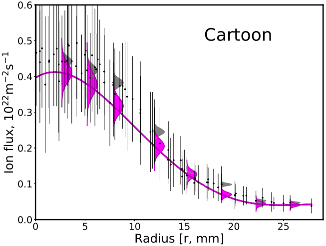
Moment/PDF matching noise model

- Lift the model fro

$f_c(x)$ ... $c_3\Psi_3(x) +$

$+ d_3\Psi_3(x)]\,\xi$

- Zero-derivative c ... $= l(d_0, d_1, d_2)$
- Object of inferen



- Match moments, or better,
  Kullback-Leibler divergence between model and data

$$KL(p_1||p_2) = \int \log\left(\frac{p_1}{p_2}\right) dp_1 \overset{\text{Gauss.}}{=} \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

- Use approximate likelihood $\log L(c) = -KL(p_y||p_f) - KL(p_f||p_y)$

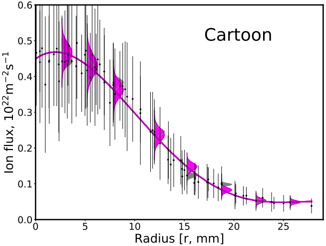
Moment/PDF matching noise model

- Lift the model fro

$f_c(x)$ ... $c_3\Psi_3(x) +$

$+ d_3\Psi_3(x)]\,\xi$

- Zero-derivative c ... $= l(d_0, d_1, d_2)$
- Object of inferenc

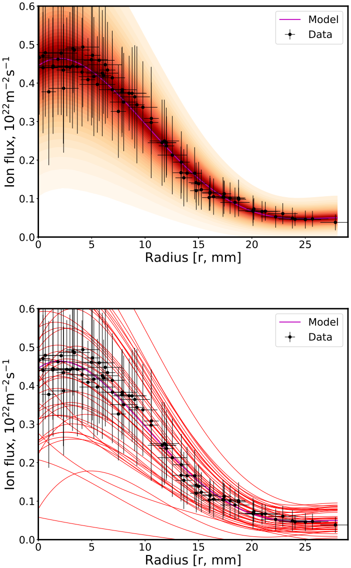


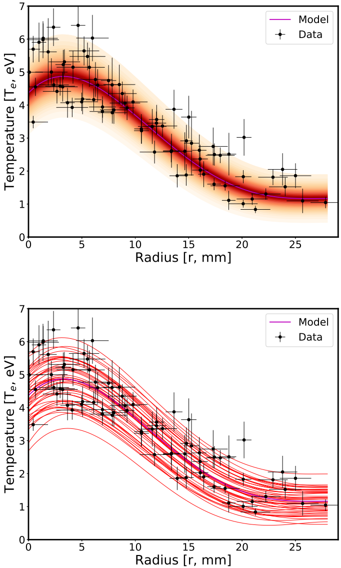
- Match moments, or better,
  Kullback-Leibler divergence between model and data

$$KL(p_1||p_2) = \int \log\left(\frac{p_1}{p_2}\right) dp_1 \overset{\text{Gauss.}}{=} \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

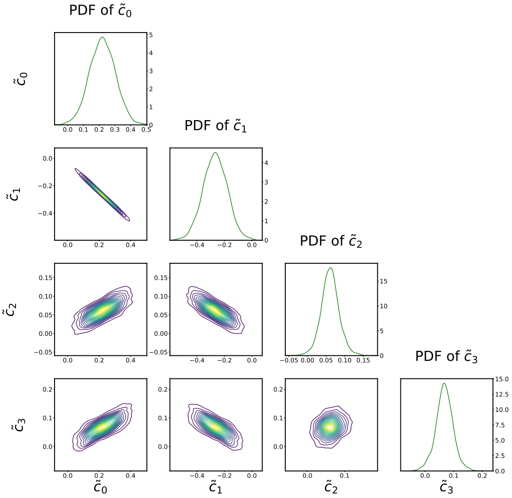
- Use approximate likelihood $\log L(c) = -KL(p_y||p_f) - KL(p_f||p_y)$

Moment/PDF matching noise model: resulting fits





Moment/PDF matching noise model: joint samples on poly. coeffs



Recall the model:

$$\underbrace{[c_0 + d_0\xi]}_{\tilde{c}_0} + \underbrace{[c_1 + d_1\xi]}_{\tilde{c}_1}\Psi_1(x) + ...$$

where ξ is standard normal, and c_i's and d_i's are represented by posterior samples via MCMC

The best option is to use the raw data

- All good, but we had to make a few assumptions/approximations
- Uncertainties in the process of producing fitted data are ignored
- As a consequence, correlations are not accounted for
- An extreme example - density is perfectly correlated with flux and temperature!

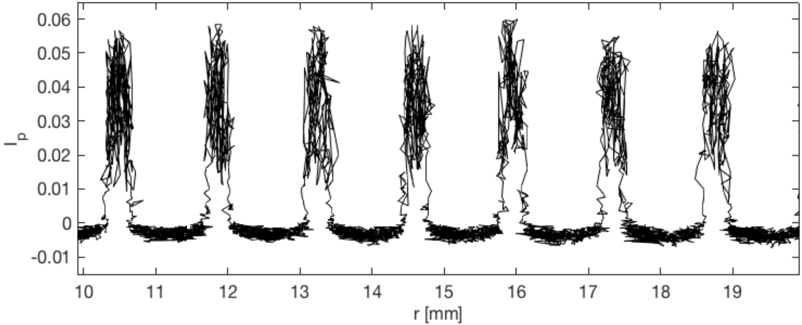
$$n_e \propto \frac{I_{is}}{\sqrt{T_e}}$$

- Using raw data would allow to put the measurement error assumptions where they belong, at the 'lowest' level

* Without raw data, we could employ maxEnt arguments to 'propose' datasets consistent with the fitted data, and treat it with a multi-stage Bayesian method [Najm et. al., IJUQ, 2014]

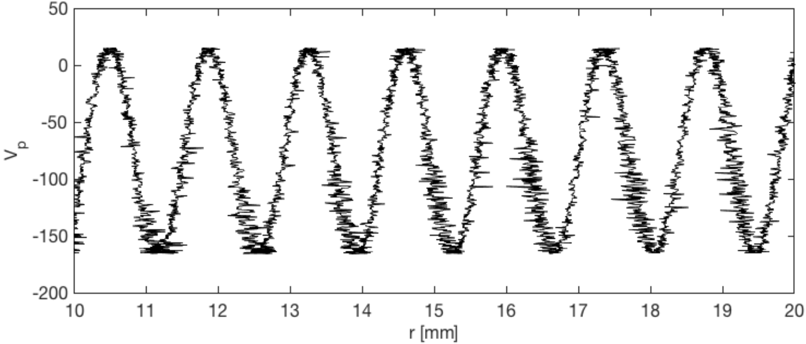
Formulate inference bottom-up, using raw data (work-in-progress)

The underlying data is for current/voltage as probe moves, followed by the I-V exponential fit to extract I_{is} and T_e, and computing (deterministically!) density $n_e \propto I_{is}/\sqrt{T_e}$.



Formulate inference bottom-up, using raw data (work-in-progress)

The underlying data is for current/voltage as probe moves, followed by the I-V exponential fit to extract I_{is} and T_e, and computing (deterministically!) density $n_e \propto I_{is}/\sqrt{T_e}$.



Formulate inference bottom-up, using raw data (work-in-progress)

The underlying data is for current/voltage as probe moves, followed by the I-V exponential fit to extract I_{is} and T_e, and computing (deterministically!) density $n_e \propto I_{is}/\sqrt{T_e}$.

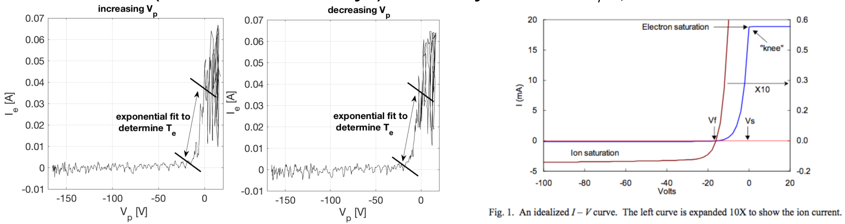


Fig. 1. An idealized $I-V$ curve. The left curve is expanded 10X to show the ion current.

Fig.1 from
[Francis Chen, Mini-Course on Plasma Diagnostics, IEEE-ICOPS meeting, June 5, 2003]

Summary

- General Bayesian machinery for fitting models to data
  - Flexibility to incorporate noise/error assumptions
  - Besides parameter estimation, it provides model selection machinery
- PISCES-A Langmuir Probe Data: three options:
  - [Done] Independent fitting with processed data
  - [In progress] Fit with raw data, retains correlations and builds on lower-level noise assumptions
  - [Not needed yet] Data space exploration using MaxEnt principle if raw data unavailable
- Any of above mechanisms provide posterior samples of fit parameters (polynomial coefficients)
  - Add to the list of uncertain inputs for GITR/Xolotl
  - Perhaps represent them with Polynomial Chaos (PC) expansions
  - Forward propagation of uncertainties with PC