Improved profile fitting and uncertainty quantification: applications to impurity and momentum transport

M.A. Chilenski, M. Greenwald, Y. Marzouk,*
N.T. Howard, A.E. White, J.E. Rice and J.R. Walk

Acknowledgements: M. Barnes, I. Abel

MIT PSFC/Alcator C-Mod
*MIT Aero/Astro, Uncertainty Quantification Group

September 10, 2014

Supported by USDoE award DE-FC02-99ER54512. Supported in part by a DOE SCGF fellowship, administered by ORISE-ORAU under contract DE-AC05-06OR23100.

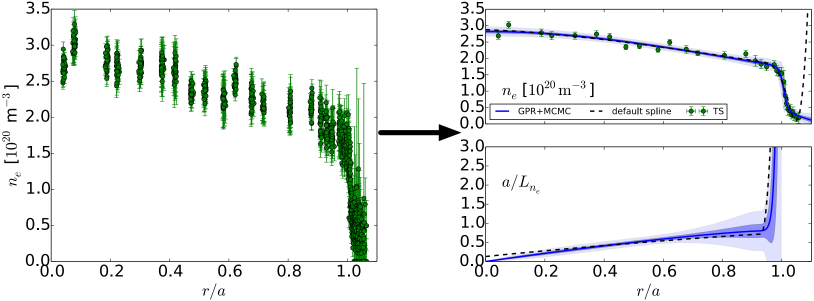
Outline: better profiles $\rightarrow$ better UQ $\rightarrow$ better physics

- Many experimental quantities are not measured directly, instead are computed with complex codes.
- Need profile fits, need to propagate uncertainties efficiently.
- Existing techniques have substantial shortcomings.
- New technique has been developed, applied to several cases:
  - Uncertainty quantification (UQ) of **gradient scale lengths**
  - UQ of L-mode **impurity transport coefficients**
  - Exploration of the connection of second derivatives with **momentum transport/intrinsic rotation**

Better profile fitting leads to better uncertainty quantification (UQ) of experimental quantities, more trust in results.

Profile fitting is fundamental to plasma data analysis...

Raw data: many samples at discrete points in space — Averaged and smoothed

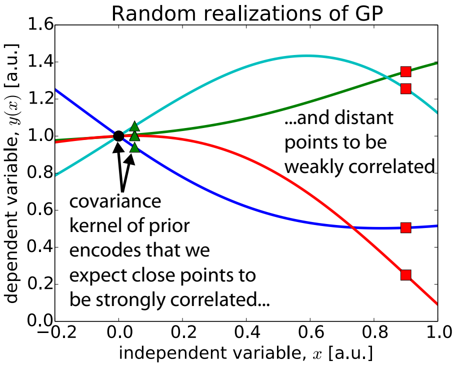


...but traditional spline methods have major shortcomings

- Analytical forms for uncertainty are cumbersome to compute, often an additional Monte Carlo step must be performed.
- Selection of properties nontrivial, often ends up being manual.
- Use of a point estimate for properties can end up hiding **substantial** uncertainty, particularly in the gradient.
- Has issues fitting *whole* profile without incorporating explicit functional form (mtanh, etc.).

Gaussian process regression (GPR) overcomes the shortcomings of splines

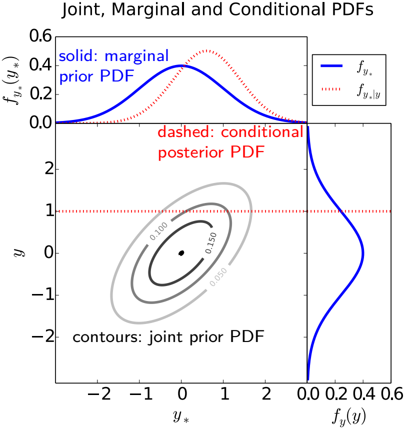
- Established statistical/machine learning technique [1].
- Expresses profile in terms of (spatial) covariance of multivariate normal (MVN) distribution.
- Selection of fit properties is **automated** and **statistically rigorous**.

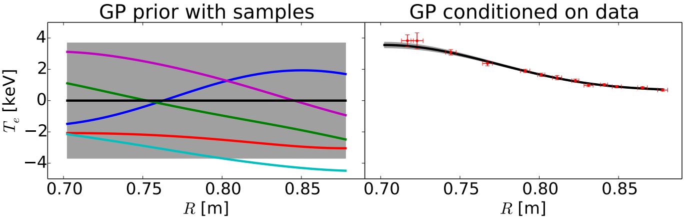


[1] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

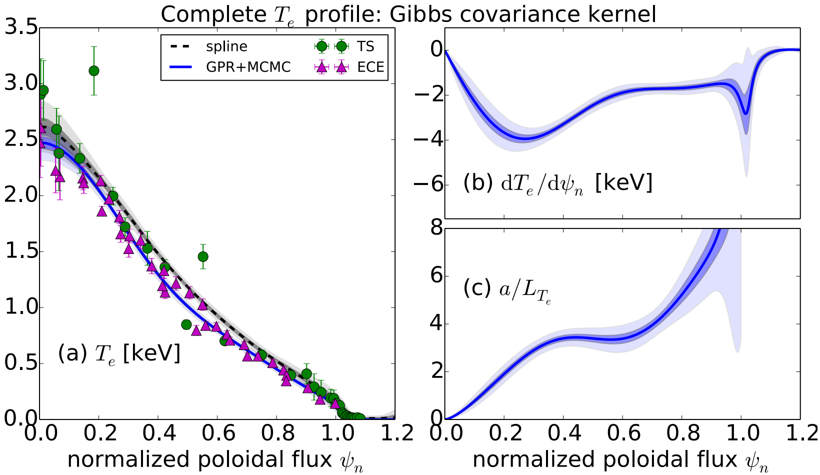
GPR: a probabilistic method to fit profiles

- Create multivariate normal *prior distribution* that sets smoothness, symmetry, etc.
- *Condition* on observations to yield the fit, including uncertainty estimate.
- Distribution can include derivatives, line integrals, volume averages, etc.





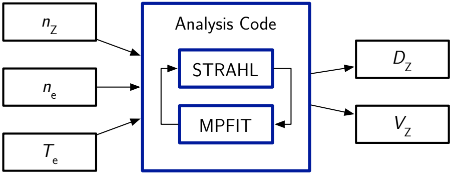
Application of GPR to C-Mod data delivers improved estimates of profile gradients, uncertainties [1]



[1] M.A. Chilenski et al. (2014), submitted to Nucl. Fusion.

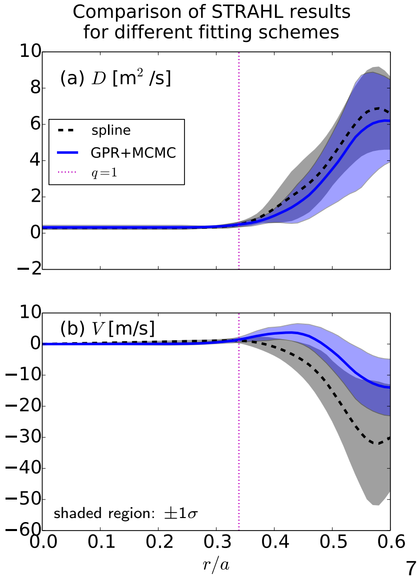
Preprint: PSFC report PSFC/JA-14-22.

GPR+Monte Carlo sampling has been applied to obtain uncertainties in experimental impurity transport coefficients

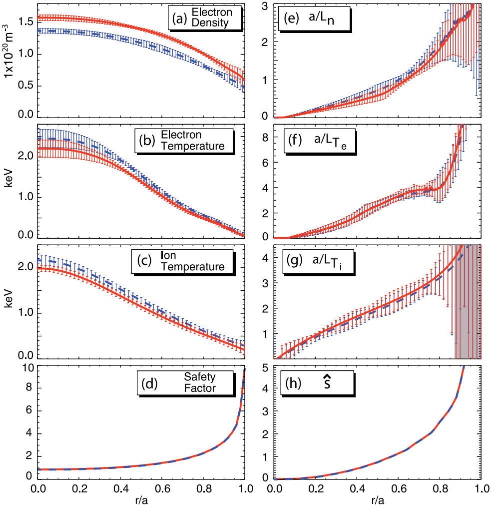


- Used 80 samples from GPR fits for T_e, n_e.
- D agrees with previous result obtained using splines.
- V differs, likely due to the 12% discrepancy in the T_e fits.

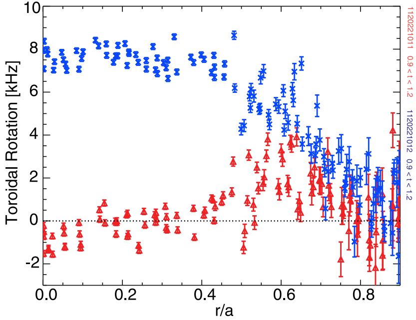
Previous work: N.T. Howard et al. (2012), Nucl. Fusion **52**, 063002



Open question: intrinsic rotation

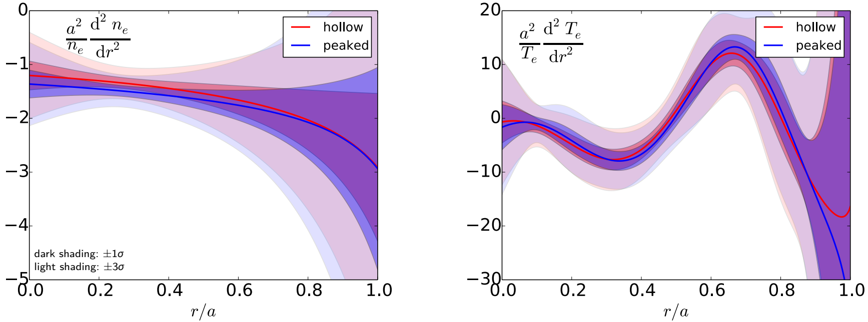


A.E. White et al. (2013), Phys. Plasmas **20**, 056106



- No external momentum input.
- Slight increase in density.
- Negligible change in gradients/turbulence drive.
- *Dramatic change in rotation profile: peaked to hollow.*

GPR enables computation of second derivatives:
More detailed physics is necessary to explain change in rotation



- Some speculation that second derivatives influence momentum transport [1].
- But, very little difference is observed between second derivative, normalized second derivative in these discharges.

More physics necessary to explain dramatic change in rotation.

[1] M. Barnes et al. (2013), PRL **111**, 055005

Advanced profile fitting enables better code validation, exploration of new physics

- Advanced profile fitting techniques improve the **credibility** and **efficiency** of uncertainty propagation through analysis codes.
- GPR has been used to fit plasma profiles and propagate uncertainties through a calculation of transport coefficients.
- GPR has been used to compute first and second derivatives *and their uncertainties*.
- Open-source software is available: github.com/markchil/gptools
- Paper has been submitted to Nuclear Fusion. Preprint: PSFC report PSFC/JA-14-22.

Better profile fitting leads to better UQ of experimental quantities, more trust in results.

Backup slides

1. Gaussian processes
2. gptools

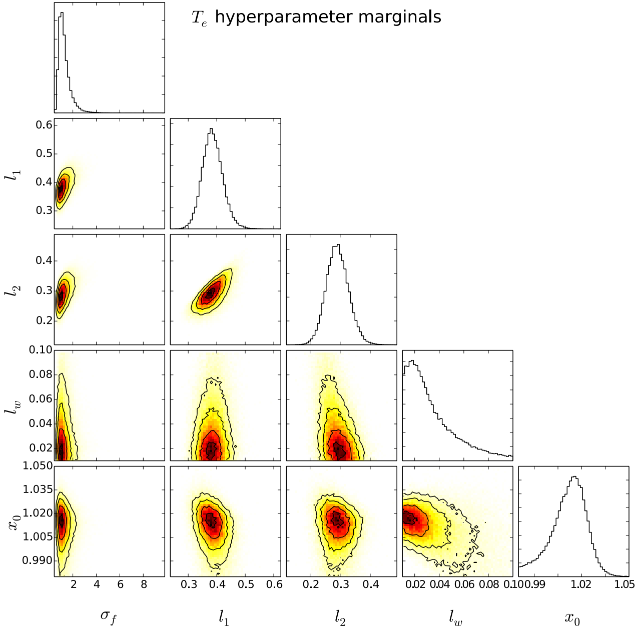
The hyperparameters can be estimated by maximizing the likelihood

Likelihood of the training data given k with hyperparameters $\boldsymbol{\theta} = [\sigma_f,\ \ell,\ \ldots]$:

$$\ln p(\boldsymbol{y}|\mathrm{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}[\mathrm{K} + \Sigma_{\mathrm{n}}]^{-1}\boldsymbol{y} - \frac{1}{2}\ln|\mathrm{K} + \Sigma_{\mathrm{n}}| - \frac{n}{2}\ln 2\pi$$

- Maximize with respect to hyperparameter vector $\boldsymbol{\theta}$.
- Local maxima: different possible interpretations of the data. E.g., noisy and long-ℓ versus precise and short-ℓ
- Compare likelihoods to select the most appropriate kernel.

Full treatment of hyperparameters uses MCMC integration

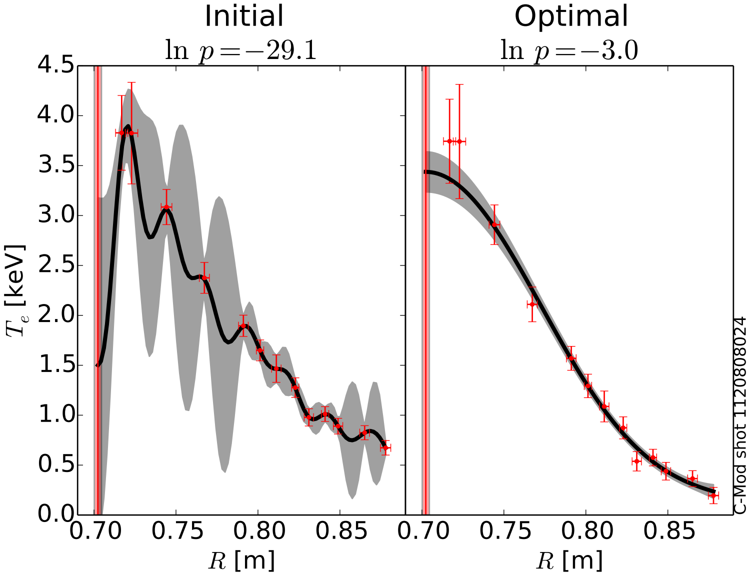


Point estimate misses substantial uncertainty in gradient

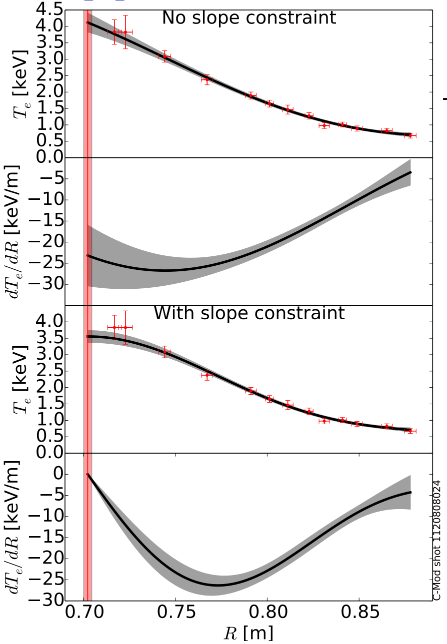
Median relative uncertainties over $0 \leq \psi_n \leq 1$

Quantity	y	y'	a/L_y
n_e, MAP	1.2%	6.0%	6.0%
n_e, MCMC	1.4%	8.4%	8.3%
T_e, MAP	1.3%	3.7%	4.0%
T_e, MCMC	1.4%	5.4%	5.6%

Bad versus good choices for the hyperparameters have a large effect on the likelihood



Getting gradients *and their uncertainties* is straightforward



The derivative of a GP is a GP:

$$\mathrm{cov}\left(y_i,\ \frac{\partial y_j}{\partial x_{dj}}\right) = \frac{\partial k(\boldsymbol{x}_i,\ \boldsymbol{x}_j)}{\partial x_{dj}}$$

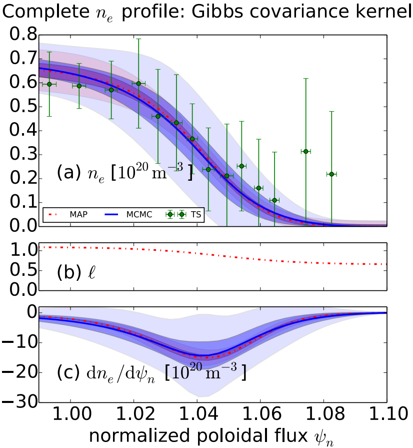
$$\mathrm{cov}\left(\frac{\partial y_i}{\partial x_{di}},\ \frac{\partial y_j}{\partial x_{dj}}\right) = \frac{\partial^2 k(\boldsymbol{x}_i,\ \boldsymbol{x}_j)}{\partial x_{di}\,\partial x_{dj}}$$

- Derivative equality constraint: just add a datapoint!
- Derivative predictions: predictive distribution contains the uncertainty.

Capturing the pedestal requires a non-stationary kernel

Gibbs kernel: ℓ is an arbitrary function of $\boldsymbol{x}$

$$k_{\mathrm{G}}(\boldsymbol{x},\,\boldsymbol{x}') = \sigma_f^2 \left(\frac{2\ell(\boldsymbol{x})\ell(\boldsymbol{x}')}{\ell^2(\boldsymbol{x}) + \ell^2(\boldsymbol{x}')}\right)^{1/2} \exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{\ell^2(\boldsymbol{x}) + \ell^2(\boldsymbol{x}')}\right)$$



- Length scale:

$$\ell = \frac{\ell_1 + \ell_2}{2} - \frac{\ell_1 - \ell_2}{2} \tanh \frac{x - x_0}{\ell_{\mathrm{w}}}$$

- Handled ℓ_1, ℓ_2, ℓ_{w} and x_0 by maximizing $\ln p$ and with MCMC.

gptools: An extensible, object-oriented Python package for multivariate GPR including gradients

- Available GPR codes lack one or more critical features:
  - Ability to both constrain and predict gradients.
  - Straightforward way to draw random samples.
- gptools was written to meet these needs:
  - Object-oriented structure.
  - Interface for easy data fusion and application of constraints.
  - SE, Gibbs, Matérn and RQ kernels *with support for arbitrary orders of differentiation*.
- Available on GitHub: www.github.com/markchil/gptools

gptools contains two classes for performing GPR

GaussianProcess

k : Kernel
nk : Kernel
X
n
y
err_y

add_data(X, y, err_y, n)
optimize_hyperparameters()
predict(X_star)
draw_sample(X_star)

Kernel

num_dim
params
fixed_params
param_bounds

__call__(Xi, Xj, ni, nj)
set_hyperparams(new_params)

gptools: An extensible, object-oriented Python package for multivariate GPR including gradients

```python
import gptools

# Create kernel:
k = gptools.SquaredExponentialKernel(1)
# Create GP:
gp = gptools.GaussianProcess(k, X=R_mid, y=Te, err_y=err_Te)
# Impose zero slope constraint at magnetic axis:
gp.add_data(R_mag, 0, n=1)
# Optimize hyperparameters:
gp.optimize_hyperparameters()

# Make a prediction of the value:
R_star = scipy.linspace(R_mag, R_mid.max(), 100)
Te_fit, Te_std = gp.predict(R_star)
# Make a prediction of the gradient:
gradTe_fit, gradTe_std = gp.predict(R_star, n=1)
```

gptools implements a very general form of GPR

$$f\left(\begin{bmatrix}\boldsymbol{M}_*\\ \boldsymbol{M}\end{bmatrix}\right)=\mathcal{N}\Bigg(\begin{bmatrix}\mathrm{T}_* & 0\\ 0 & \mathrm{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{\mu}(\mathrm{X}_*)\\ \boldsymbol{\mu}(\mathrm{X})\end{bmatrix},$$
$$\begin{bmatrix}\mathrm{T}_* & 0\\ 0 & \mathrm{T}\end{bmatrix}\begin{bmatrix}\mathrm{K}(\mathrm{X}_*,\mathrm{X}_*) & \mathrm{K}(\mathrm{X},\mathrm{X}_*)\\ \mathrm{K}(\mathrm{X}_*,\mathrm{X}) & \mathrm{K}(\mathrm{X},\mathrm{X})\end{bmatrix}\begin{bmatrix}\mathrm{T}_*^T & 0\\ 0 & \mathrm{T}^T\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & \Sigma_M\end{bmatrix}\Bigg)$$

$$\ln\mathcal{L}=-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln\left|\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M\right|$$
$$-\frac{1}{2}(\boldsymbol{M}-\mathrm{T}\boldsymbol{\mu}(\mathrm{X}))^T(\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M)^{-1}(\boldsymbol{M}-\mathrm{T}\boldsymbol{\mu}(\mathrm{X}))$$

$$f(\boldsymbol{M}_*|\boldsymbol{M})=\mathcal{N}\big(\mathrm{T}_*\boldsymbol{\mu}(\mathrm{X}_*)+\mathrm{T}_*\mathrm{K}(\mathrm{X}_*,\mathrm{X})\mathrm{T}^T(\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M)^{-1}(\boldsymbol{M}-\mathrm{T}\boldsymbol{\mu}(\mathrm{X})),$$
$$\mathrm{T}_*\mathrm{K}(\mathrm{X}_*,\mathrm{X}_*)\mathrm{T}_*^T-\mathrm{T}_*\mathrm{K}(\mathrm{X}_*,\mathrm{X})\mathrm{T}^T(\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M)^{-1}\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X}_*)\mathrm{T}_*^T\big)$$

- Supports data of arbitrary dimension $\boldsymbol{x}\in\mathbb{R}^n$.
- Supports explicit, parametric mean function $\mu(\boldsymbol{x})$: can perform nonlinear regression with GP fit to residuals.
- Supports arbitrary linear transformations T, T_* of inputs, outputs: can perform tomographic inversions constrained with point measurements.
- Supports noise of arbitrary structure Σ_M on observations.