Uncertainty quantification for models based on plasma profiles

Christopher G. Albert

November 20, 2018

NMPP, Max-Planck-Institut für Plasmaphysik, 8574 Garching, Germany
(formerly: MST1, Fusion@OEAW, TU Graz, 8010 Graz, Austria)

Ringberg Theory Seminar 2018

Motivation

- Many of us run models based on plasma profiles, e.g. from ASDEX Upgrade
- Example setup from my experience:
  1. 2D Equilibrium from CLISTE
  2. Non-axisymmetric 3D equilibrium from VMEC
  3. Neoclassical toroidal viscous torque profile from NEO-2
- How reliable are computed results based on uncertain data?

Current Helmholtz project (*RedMod*, many partners, Udo von Toussaint, Tom Tyranowski and me at IPP) should use and extend methods of model complexity reduction and **uncertainty quantification** that are applicable in a variety of settings.

Common workflow

1. Take experimental plasma profiles
2. Choose parametrization for profile
3. Fit profile parameters
4. Run our model at best fitted parameters
5. Publish

More honest workflow

1. Take experimental plasma profiles
2. Choose parametrization for profile
3. Fit profile parameters
4. Run our model **at different variations within parameter uncertainty**
5. **Analyze uncertainty propagation**
6. Publish

Tools for uncertainty quantification, interpolation and surrogate (proxy) models

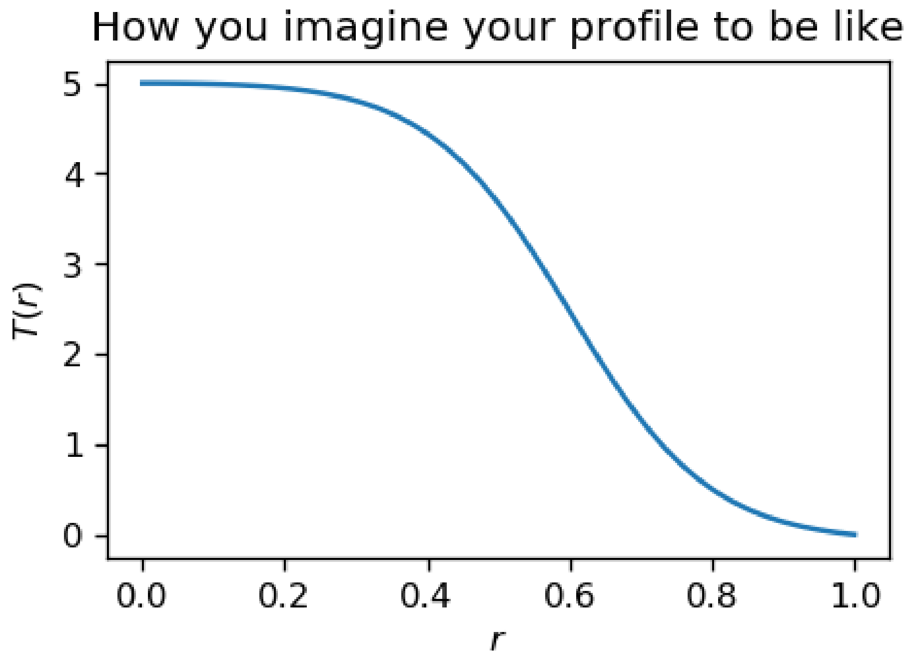
- Python libraries for rapid testing
  - **NumPy/Matplotlib/SciPy** for numerics, plotting, regressions, splines
  - **ChaosPy** for uncertainty quantification (Polynomial Chaos)
  - **rbf** and **scikit-learn** for meshless interpolation on unstructured point clouds and construction of surrogate models with uncertainties (Gaussian Processes)

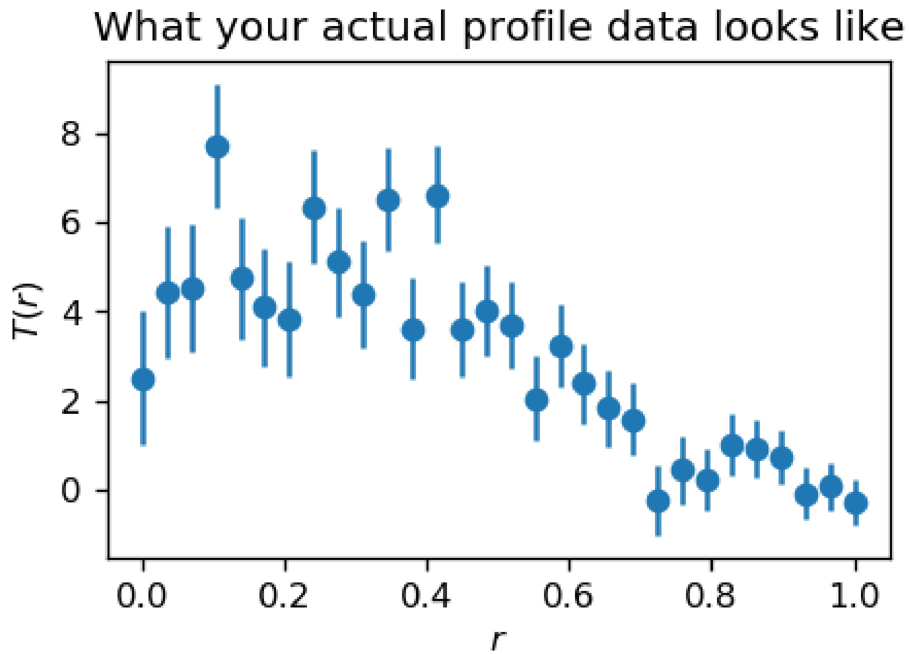
- In-house Fortran codes for **high performance** (Roland Preuss)
  - **Polynomial Chaos** Expansion with MPI parallelization (Jalal Lakhlili)
  - **Gaussian Process** regression with adaptive choice of sampling points

R Preuss, and U von Toussaint. "Global Optimization Employing Gaussian Process-Based Bayesian Surrogates." Entropy 20.3 (2018): 201.

1. Take experimental plasma profiles

Let's assume you are a theoretician:





For testing, we have invented this artificial temperature profile $T(r)$ over radius r. For reality see *R Fischer, et al. "Integrated data analysis of profile diagnostics at ASDEX Upgrade." Fusion science and technology 58.2 (2010): 675-684.*

2. Choose parametrization for profile

- Here we use two free parameters:

  1. $u = T(r = 0)$ is the temperature on the magnetic axis
  2. v parameterizes the pedestal radial position

```
def tprof(r, u, v): w = 5.0 # pedestal width (fixed)
  Tnorm = tanh(w*(v - r)) + r*w/cosh(w*v)**2 # tanh-like
  Tnorm0 = tanh(w*v) # Tnorm at r=0
  Tnorm1 = tanh(w*(v - 1)) + w/cosh(w*v)**2 # Tnorm at r=1
  return u/(Tnorm0 - Tnorm1)*(Tnorm - Tnorm1) # scale with u
```

- Temperature $T(r = 1)$ at the outer boundary must vanish

- Radial derivatives dT/dr must vanish on axis at $r = 0$ (depends on choice of r)

3. Fit profile parameters

- Usual least-squares regression assumes (multivariate) normal distribution for parameters $\rightarrow$ Gaussian probability density. Use this one for simplicity.
  - Alternative: Bayesian regression to estimate actual distribution

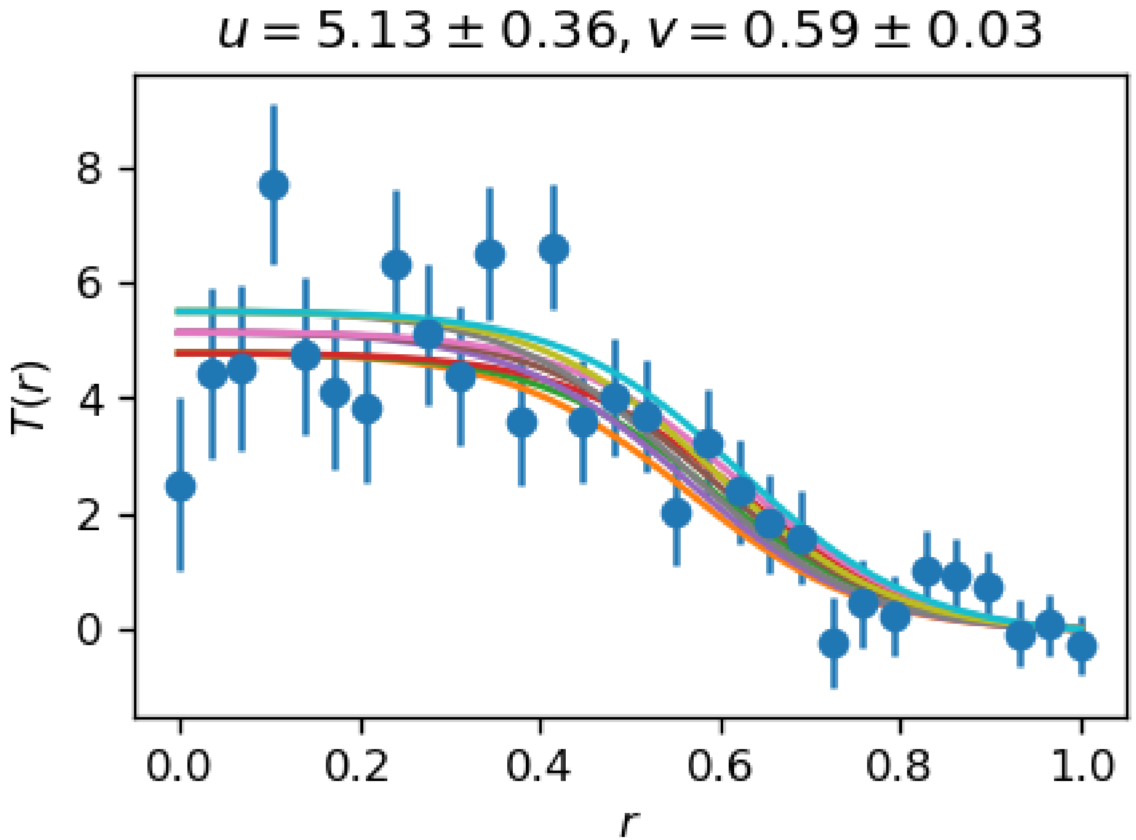
- If too many and/or correlated parameters, reduce by PCA

- For non-Gaussian and correlated parameters, use Rosenblatt transform

See, e.g., *A Boucher, and D Roussos. "Block simulation of multiple correlated variables." Mathematical Geosciences 41.2 (2009): 215-237.*

3. Fit profile parameters: results

This is what our fit looks like:

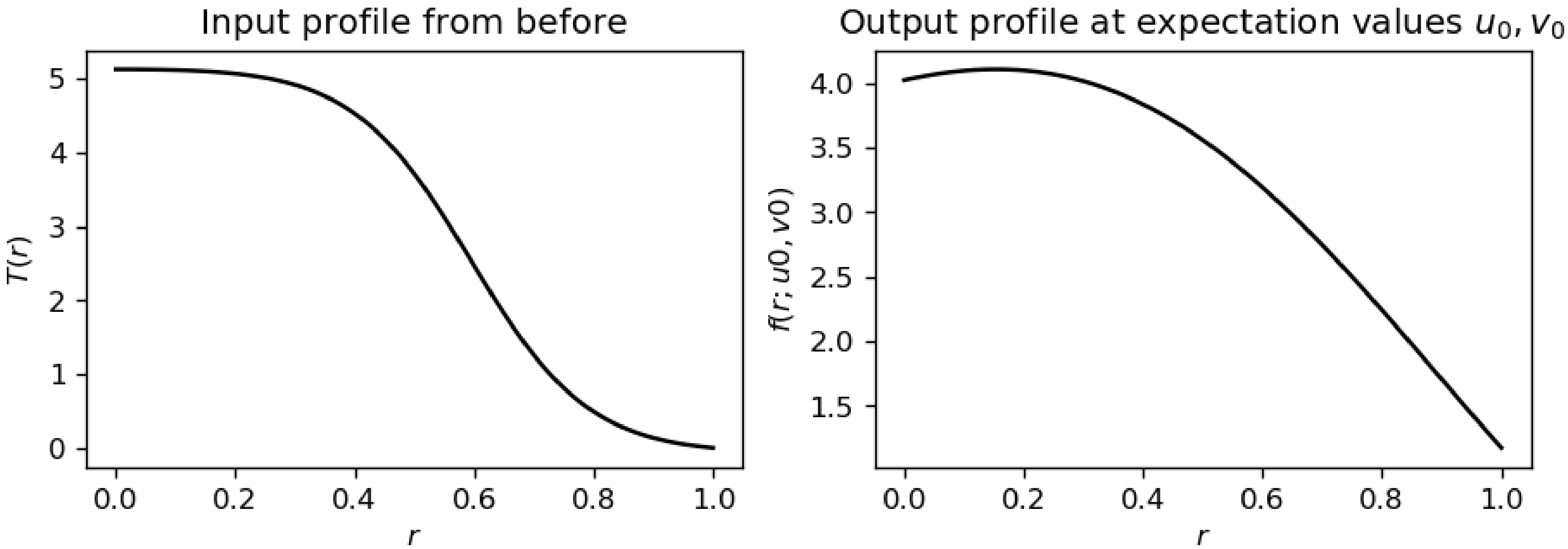


Multiple fitted curves indicate possible profiles within parameter standard deviation.

4. Run our model at best fitted parameters

Assume we have a model $f(r, T(r))$ that, for a given profile, yields a certain result,

$$f(r, T(r; u, v)) \equiv f(r; u, v).$$

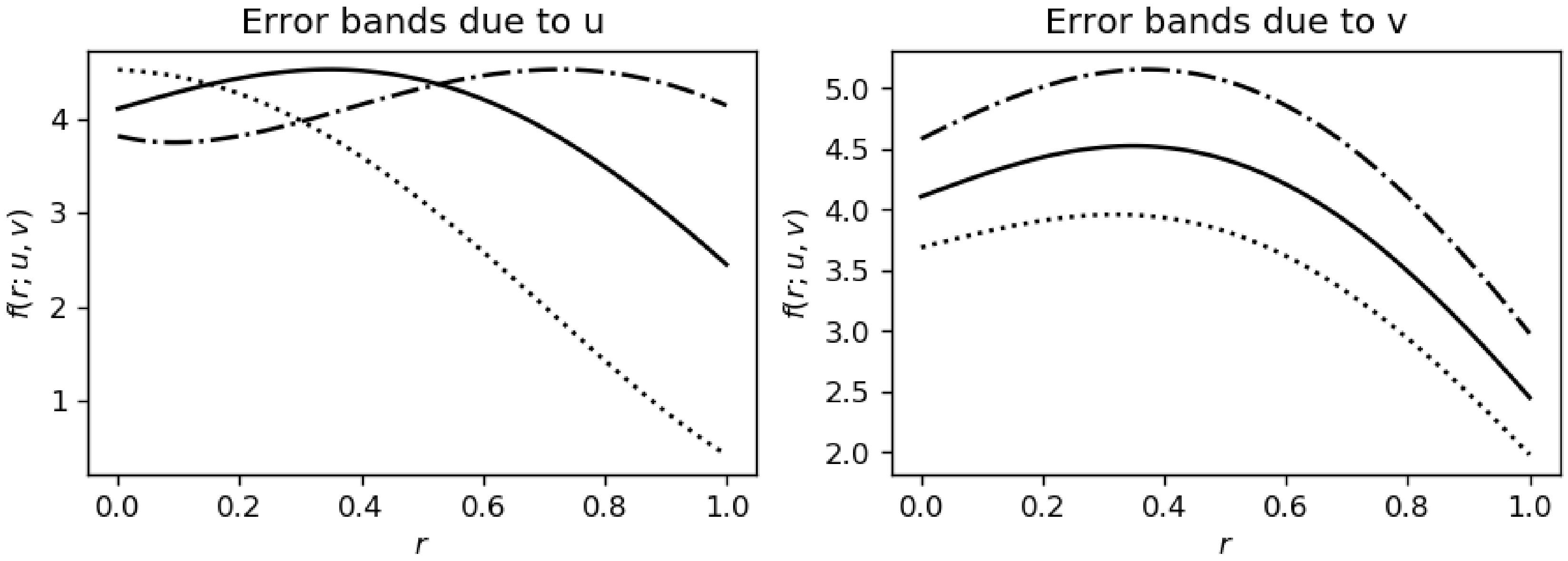


"Looks like a plateau at $r = 0.2$. This finally explains reduced anomalous transport."

4a. Run our model at some distance from best parameters

Experimental uncertainties can lead us to doubt the reliability of our result. First try: Add error bars by running the model at standard deviation distance,

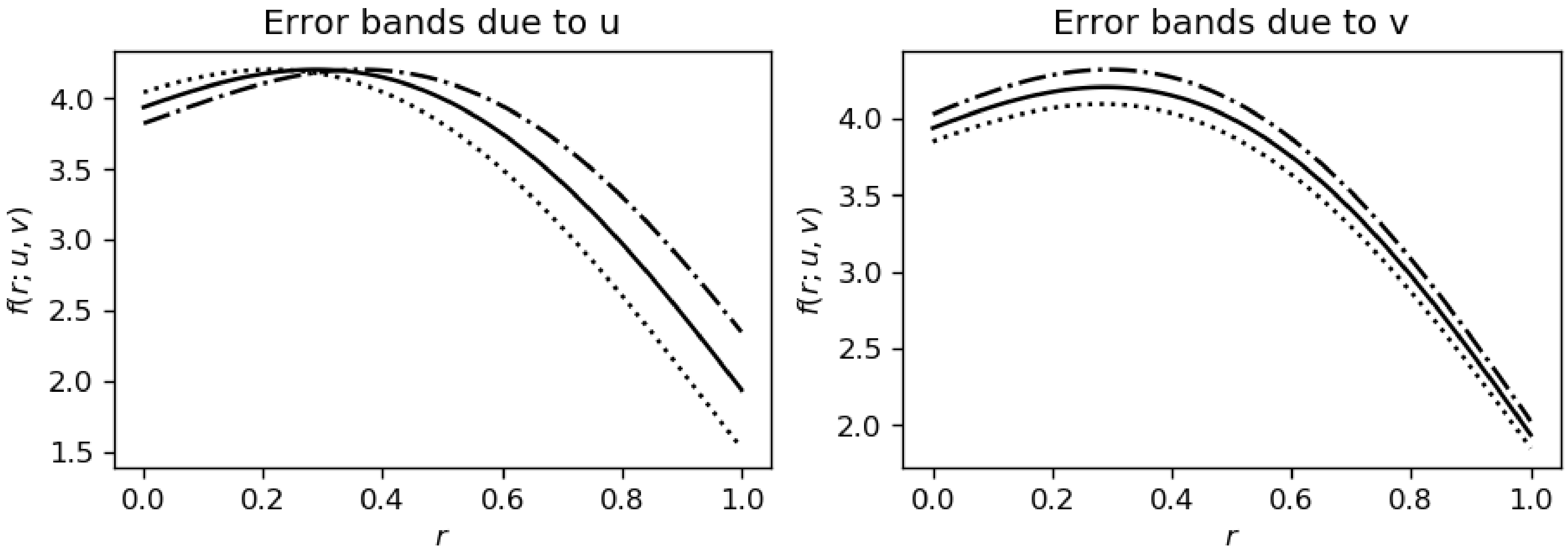
$$f(r; u_0 + \Delta u, v_0), f(r; u_0 - \Delta u, v_0), f(r; u_0, v_0 - \Delta v), f(r; u_0, v_0 + \Delta v)$$



There is quite some **uncertainty**, looks reasonable in v, but **not at all** in u.

4a. Run our model at some distance from best parameters

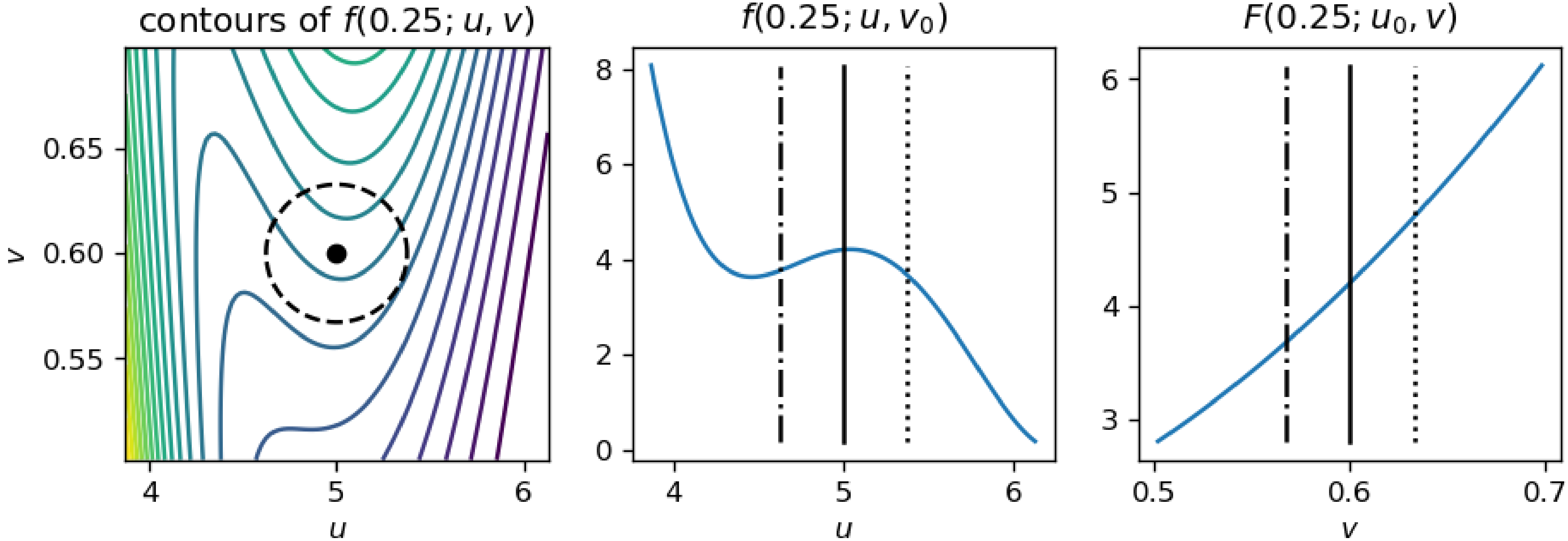
Let’s try again, but this time with 5 times smaller uncertainties in u, v.



Reducing input error from 10% to 2% we see that error band due to u **switches sign** at $r \approx 0.25$. It **fails** at 10% input error because we have left the **linear range**. To investigate, look at **actual** dependency of f on parameters u, v.

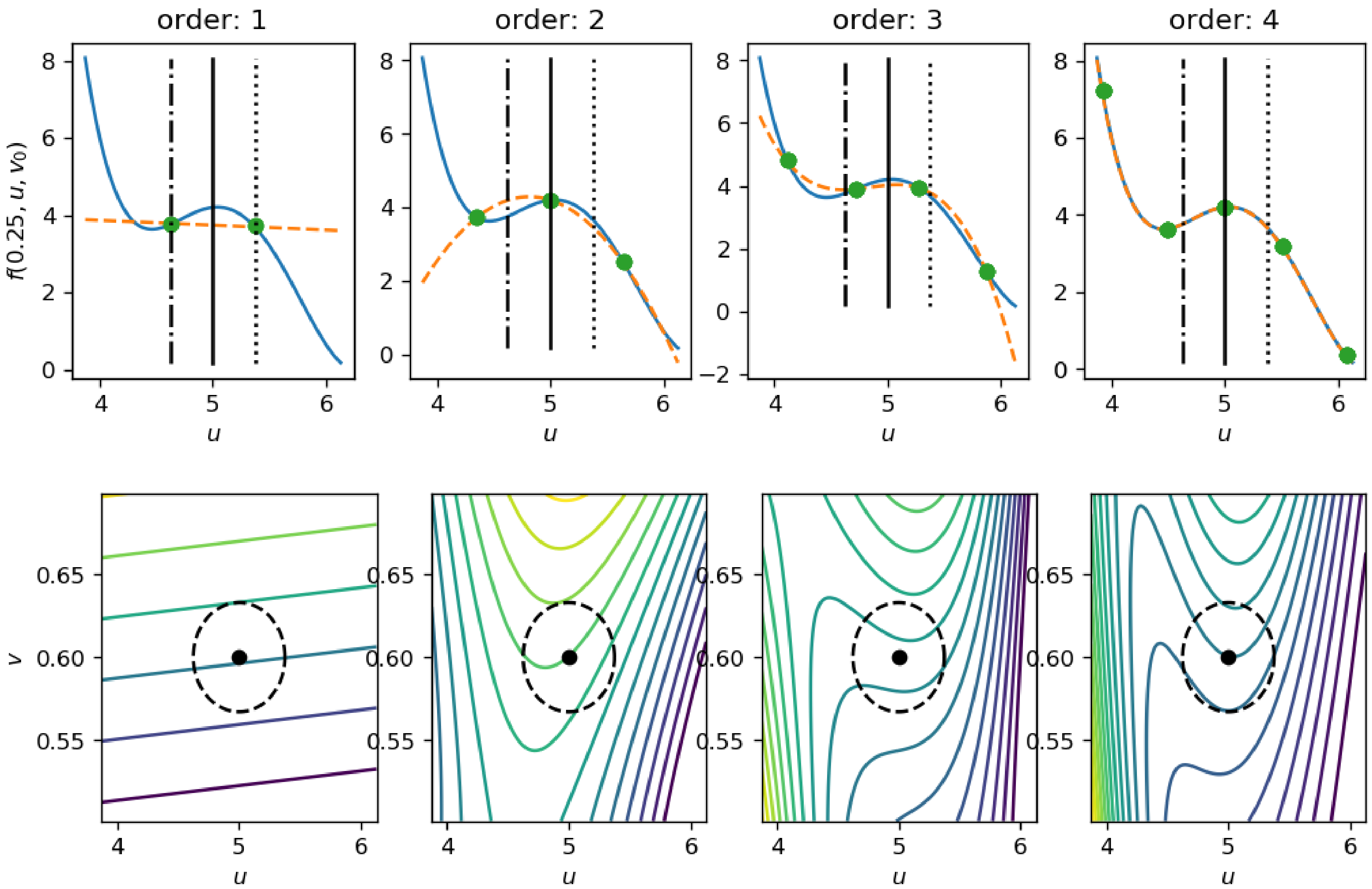
What has gone wrong?

Of course I've maliciously designed f such that it depends on u, v in a **non-linear** way. (Here we have used the Rosenbrock function)



Here we hit a **local maximum** that explains switch in sign of error bands. There is no reason why it can't be **even worse** in practice. Still f depends on u, v in a nice **continuous** way $\rightarrow$ can **expand** in **polynomials**.

Expansion in polynomials



Polynomial Chaos Expansion

How do (moments of) distribution in parameters (u, v) propagate to $f(r; u, v)$?

Polynomial chaos expansion uses analytical expressions when parameters (u, v) are fed to polynomial approximations to f.

There exists a variety of polynomials suited for different parameter distributions. Generally, **probability density function** $p(x)$ of parameters is equal to **weight function** under which polynomials are orthogonal. e.g. **Hermite** polynomials

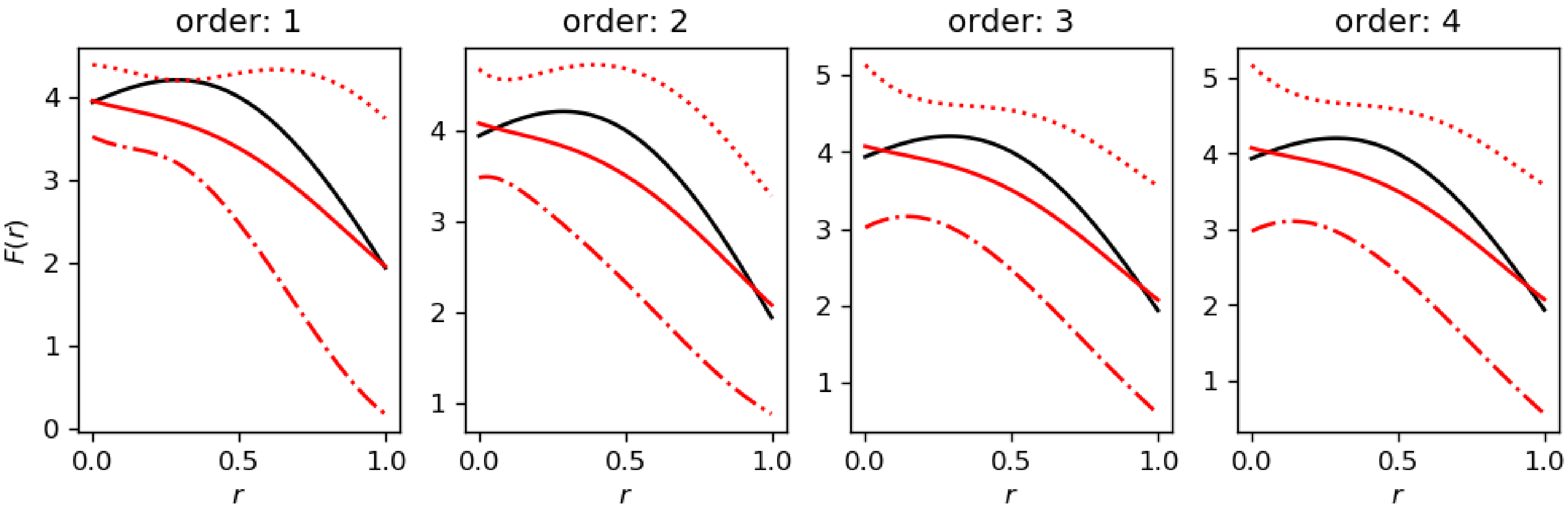
$$\int_{-\infty}^{\infty} H_m(x) H_n(x) e^{-x^2} du \propto \delta_{mn}$$

are to be used with **Normal** distribution with Gaussian probability density function.

N E Owen. "A comparison of polynomial chaos and Gaussian process emulation for uncertainty quantification in computer experiments." PhD Thesis (2017).

Polynomial Chaos Expansion: results

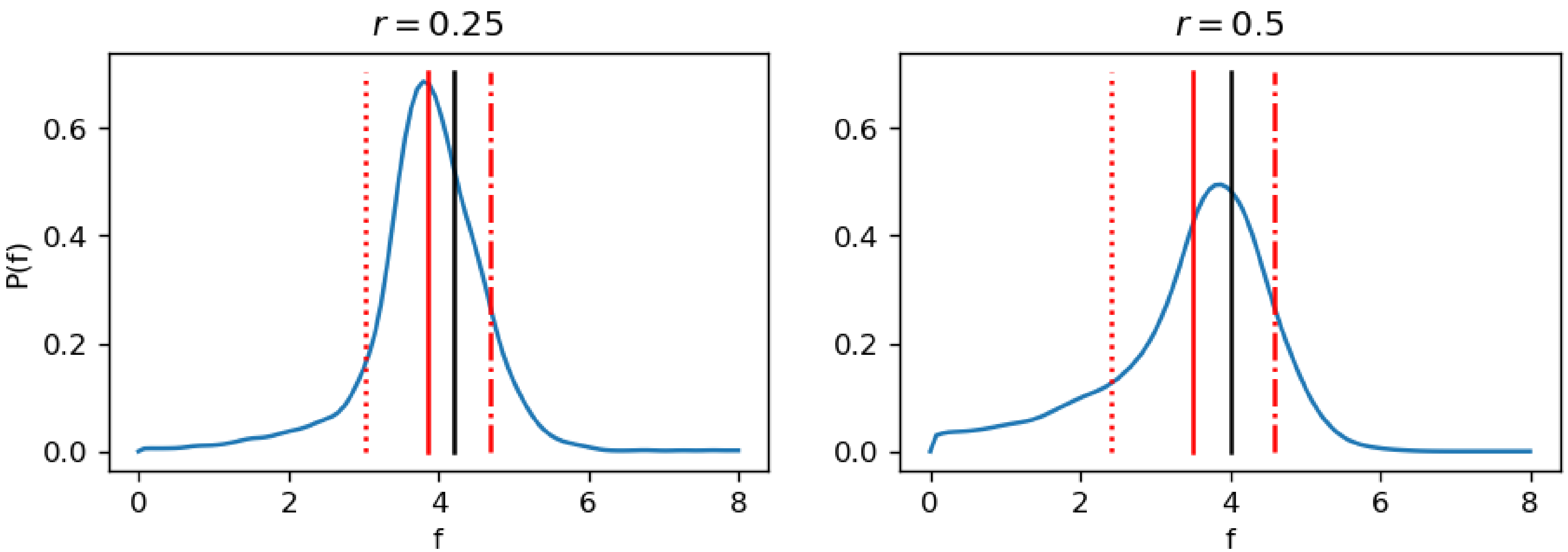
1st order: 4 evaluations, 2nd order: 9 evaluations,
3rd order: 16 evaluations, 4th order: 25 evaluations



Black: original model result, **Red:** expectation value (solid) and std.dev. from PCE

1. **Expectation value** of f **different** from run at expectation values of (u_0, v_0).
2. Here, expectation value accurate already in **1st order**, error bands at order ≥ 3.

Polynomial Chaos Expansion: explanation



Blue: Probability density function for model result f. **Red:** Expectation value of f $\pm$ std.dev. by PCE. **Black:** Model result at expected parameters (u_0, v_0) from fit. Nonlinearity of f in fit parameters u, v leads to **shifted non-Gaussian** distribution of f and therefore directly and indirectly to **different best guess**.

Conclusion

1. Being honest to ourselves we **should only believe** conclusions from modeling results from **more** than a **single run** on fitted plasma profiles.

2. In our example with 2 free parameters already **4 runs** allow for use of **PCE** for **good expectation** values and **16 runs** for **quantification of uncertainties**.

3. Keep number of **free parameters small**, as it appears in **exponent** of run count.

Slides and Python example available on https://itp.tugraz.at/~ert/teaching/

Outlook

- Apply to existing set of codes CLISTE+VMEC+NEO-2 for computation of NTV and impurity transport in ASDEX Upgrade with 3D perturbations.
- Extend to structure-preserving properties of GEMPIC, Hamiltonian codes, Cross-disciplinary applications within the Helmholtz RedMod project.
- If you have experience or would like to apply UQ methods, contact us!

Thank you for your attention, have a nice discussion and lunch!