NEPTUNE UKAEA TECHNICAL: UQ

Ed Threlfall (Joseph Parker)

NEPTUNE Workshop (The Cosener's House, Abingdon)

6 September 2022

UK Atomic Energy Authority

Smooth function fitting: GPs vs Multivariate Adaptive Regression Splines (MARS)

- **Introduction:** overview of GPs and MARS, description of test, metrics. Scaling – splines cheap O(N); GP $O(N^3)$.
- **Smooth functions**
- **Smooth functions with noise**
- **Conclusion**

- **Additional:**
  - **Nektar++ slot convection**
  - **Smallab**

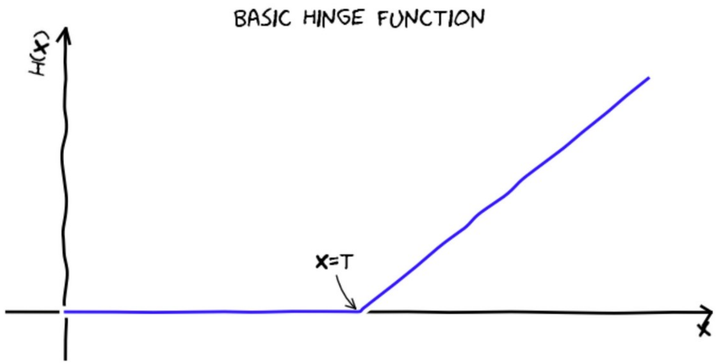
ExCALIBUR

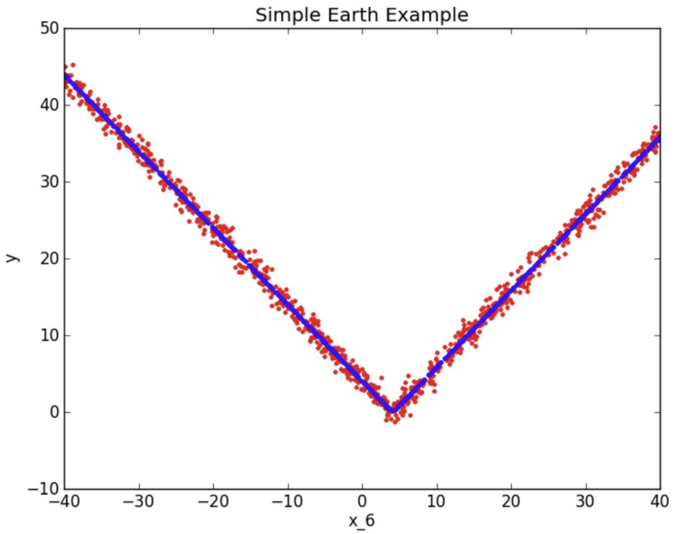
MARS

J.H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, pages 1–67, 1991.

- **Multivariate Adaptive Regression Splines**
  - **Hinge functions used in fit – can handle discontinuous derivative.**
  - **Fit is linear combination of global basis functions; each term is product of constant, linear function, hinge functions. (Num terms, product size are params.)**

$$\mathrm{h}(x-t) = [x-t]_+ = \begin{cases} x-t, & x > t \\ 0, & x \le t \end{cases}$$



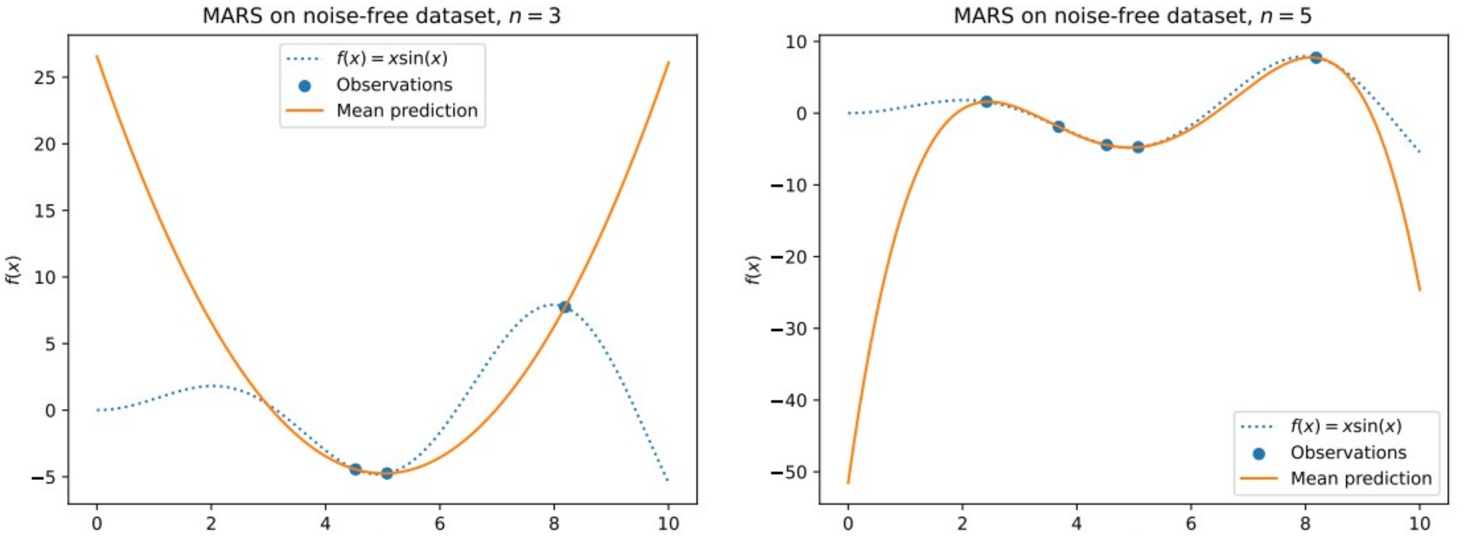


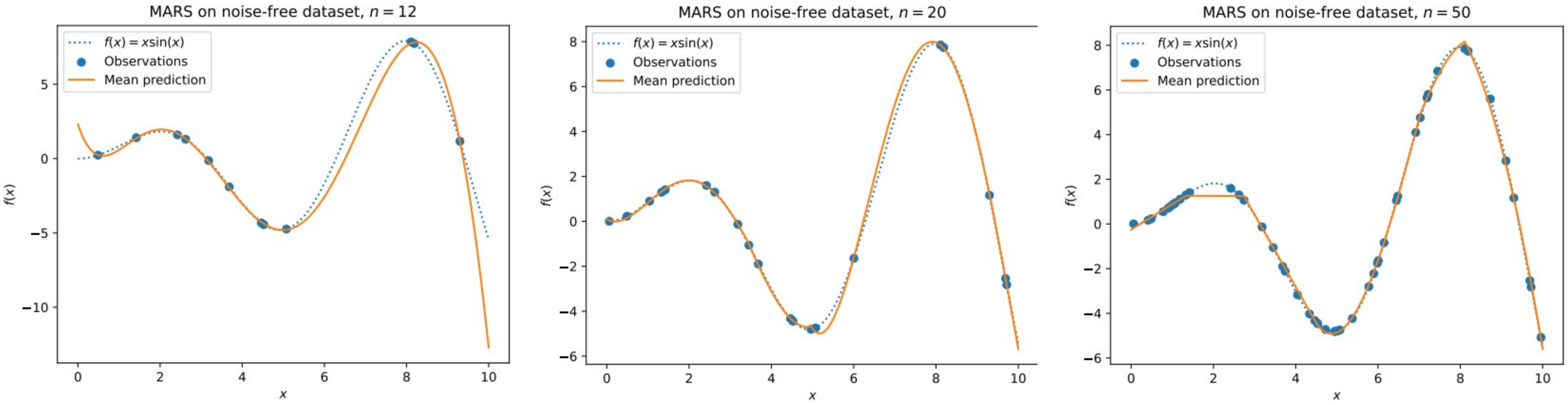
- **See** https://contrib.scikit-learn.org/py-earth/index.html (is source of these figures).

Fitting a smooth function with MARS

• , random sampling points

- : pretty bad
- : reasonable fits, but requires many more points for good fit
- Can be dubious when there are gaps in data (see)

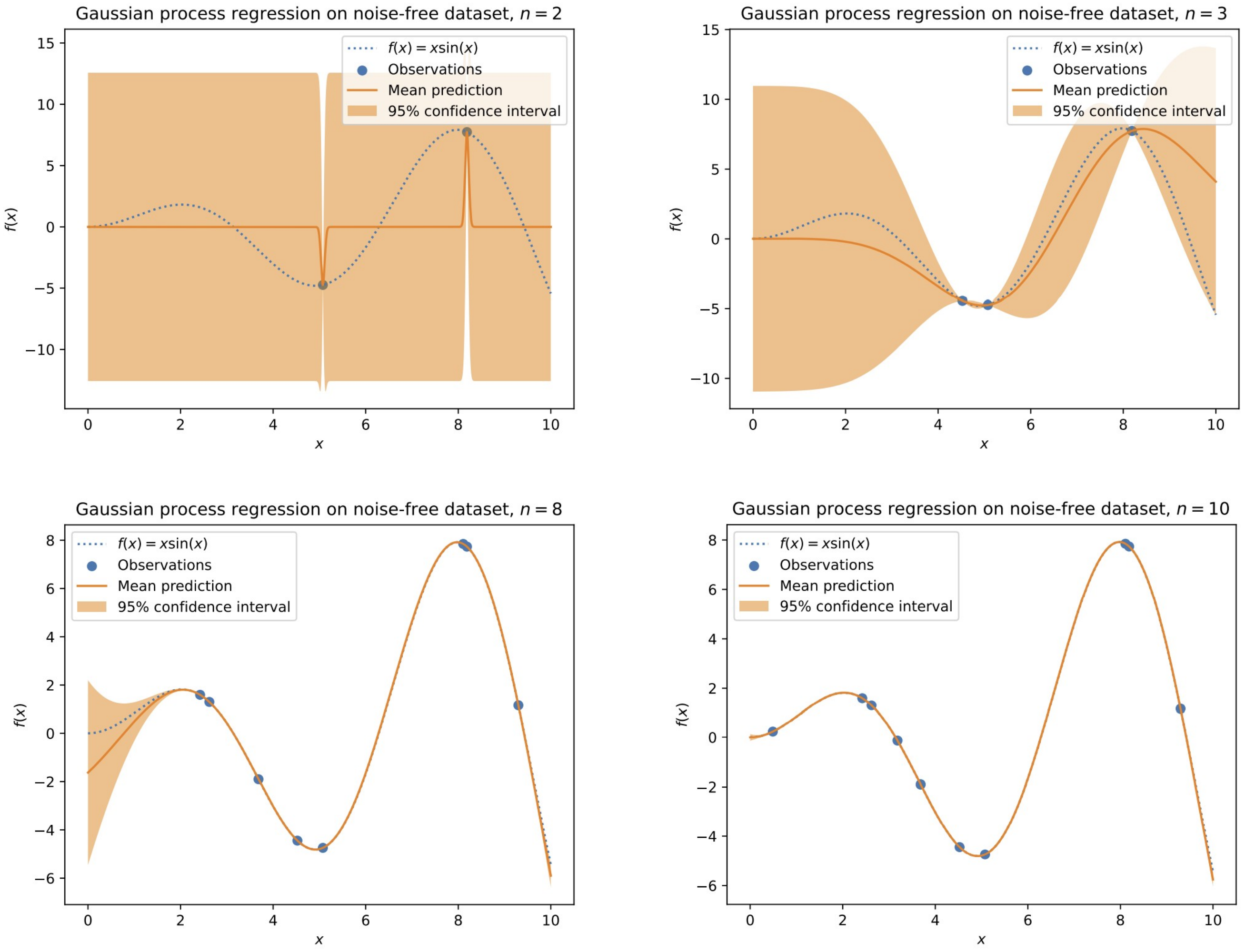




Fitting a smooth function with GPs

- , random sampling points*

- : meaningless…
- : poor but meaningful, bad wher extrapolating
- : good within training range
- : near-perfect by eye



GP has info about own uncertainty … MARS is able to do, but less clear; may be only implemented in R version of package.
Cheap to do perturbed ensemble ...

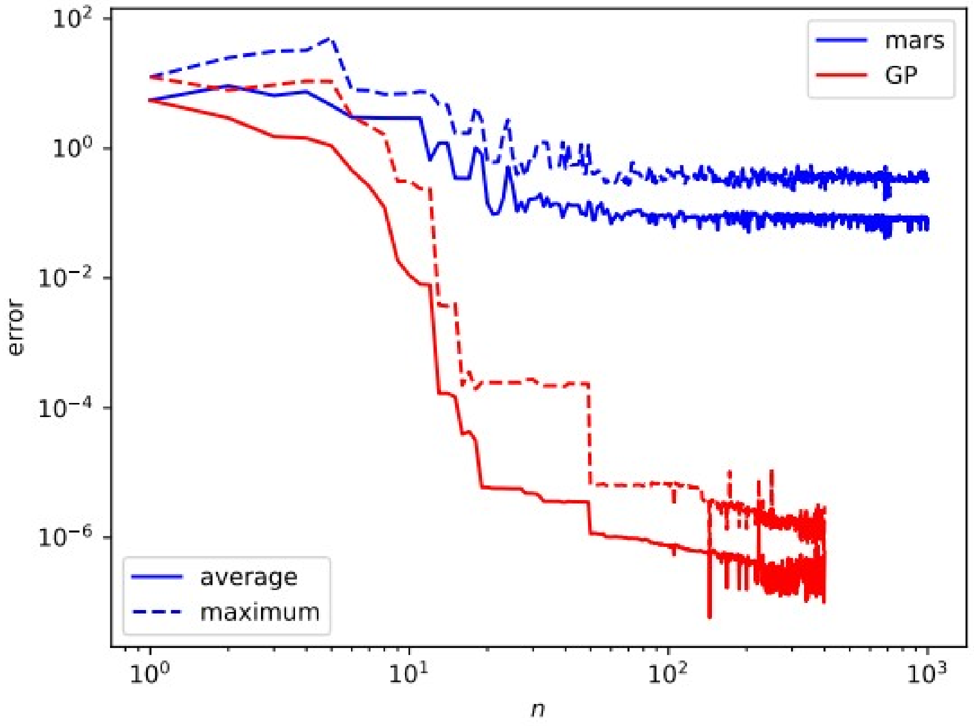
*Test cases adapted from SKLearn examples:

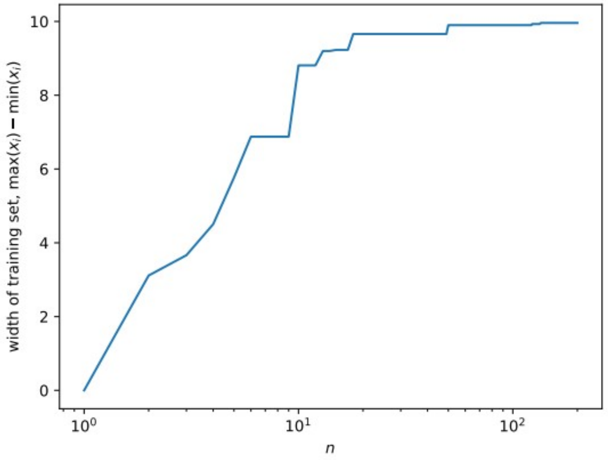
https://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_noisy_targets.html

Comparative error: GPs vs MARS

- , random sampling points*

- GPs have smaller error and converge more quickly than MARS
- Jumps in convergence correspond to making the training set wider (more interpolation, less extrapolation)
- Can we make convergence quicker by choosing training points adaptively?



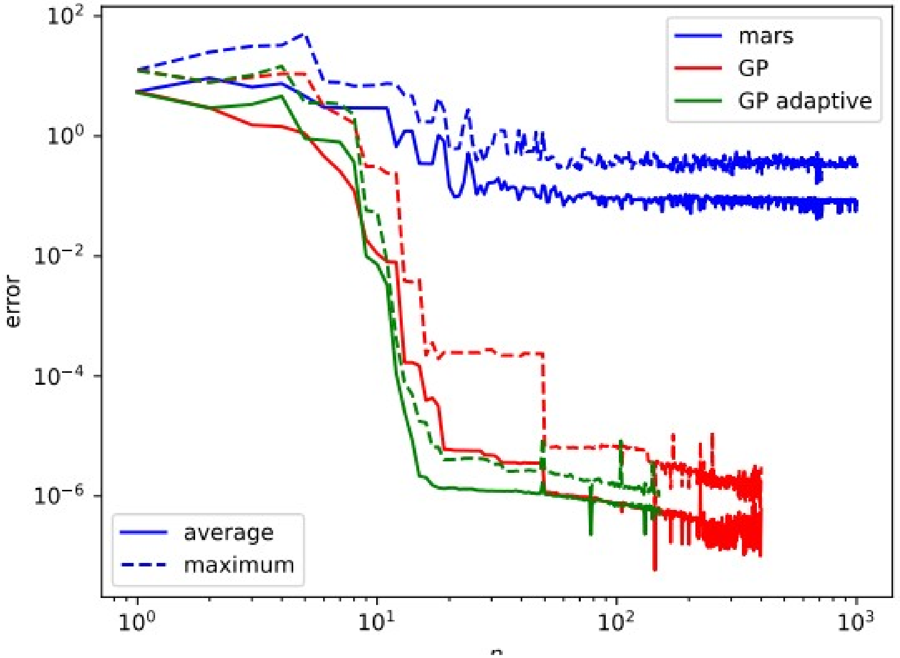


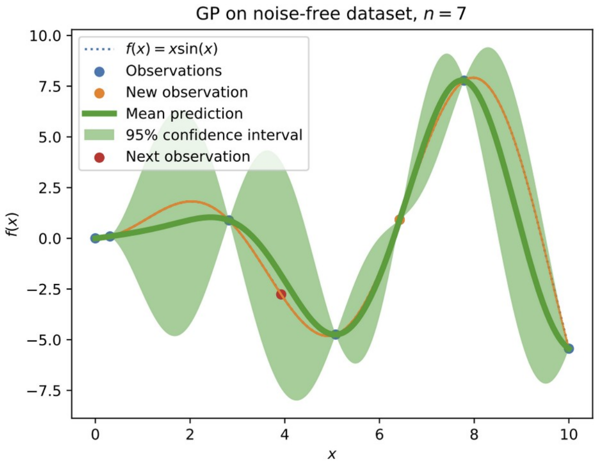
*Test cases adapted from SKLearn examples:
https://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_noisy_targets.html

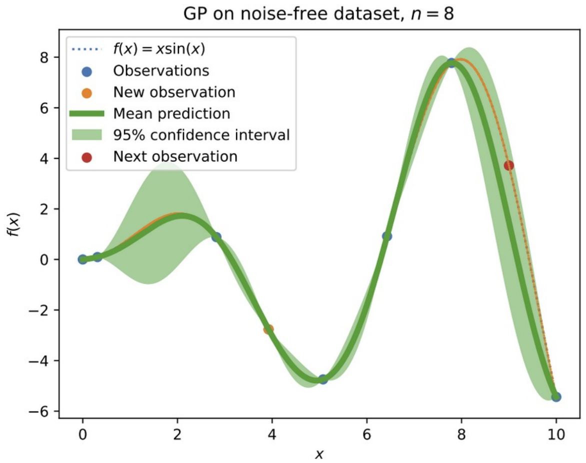
GPs with adaptive sampling

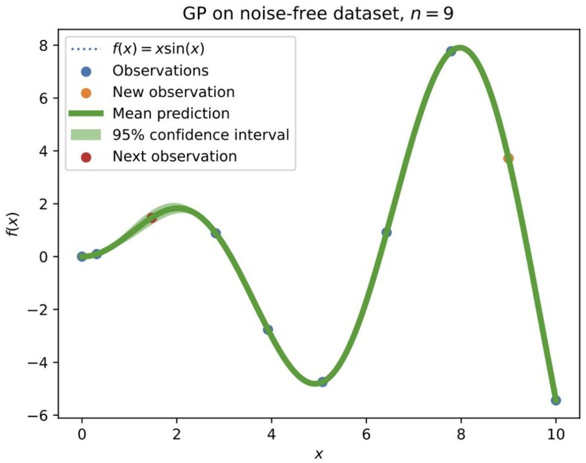
- , adaptive sampling

- GPs give confidence intervals of fit, choose sampling point where CI is largest
- Speeds convergence, though effect not very large with smooth functions
- Lower bound of error no point using more than ~20 points





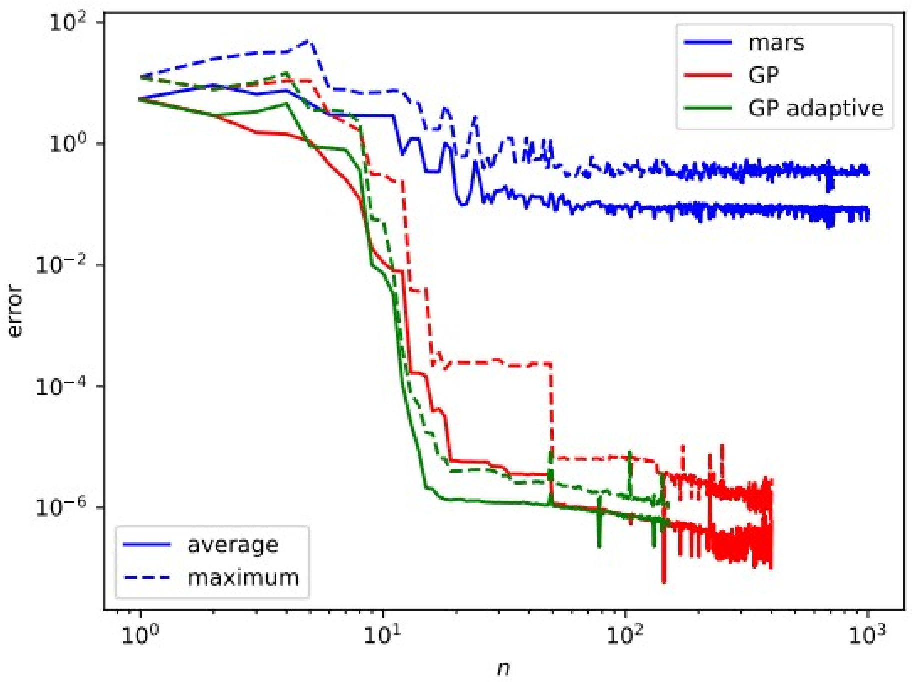


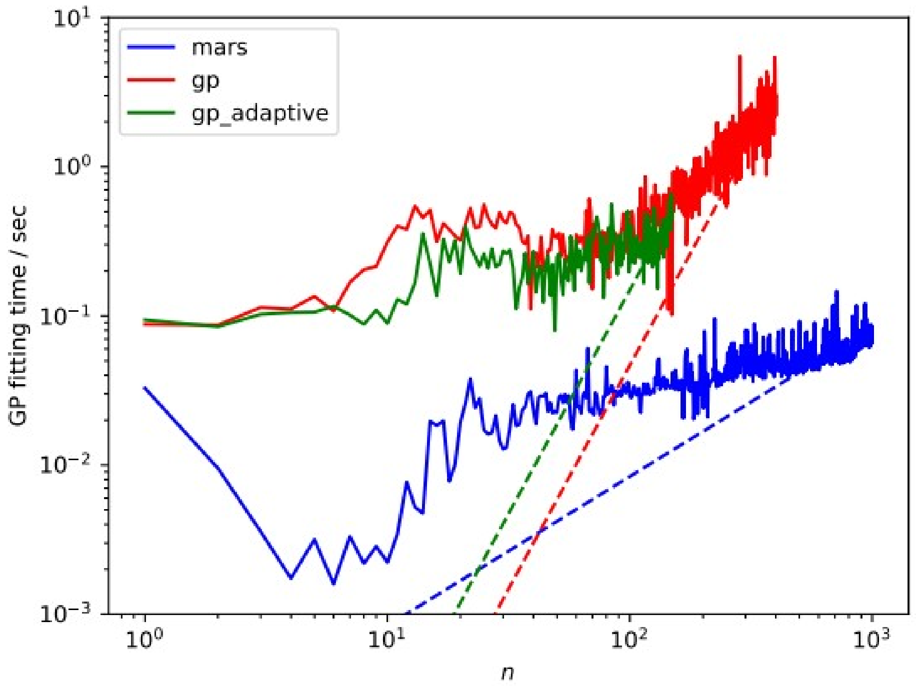


Comparative run times

- , adaptive sampling

- GPs converge very quickly, but take longer to run
- This is not a concern when GP requires few points
- How does noisy data change this picture?

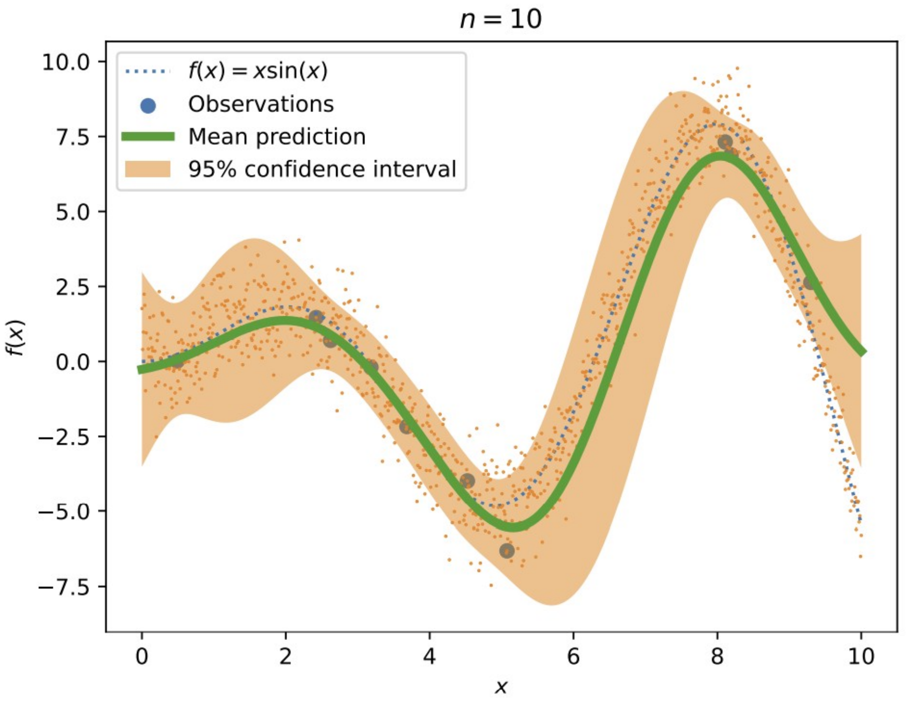


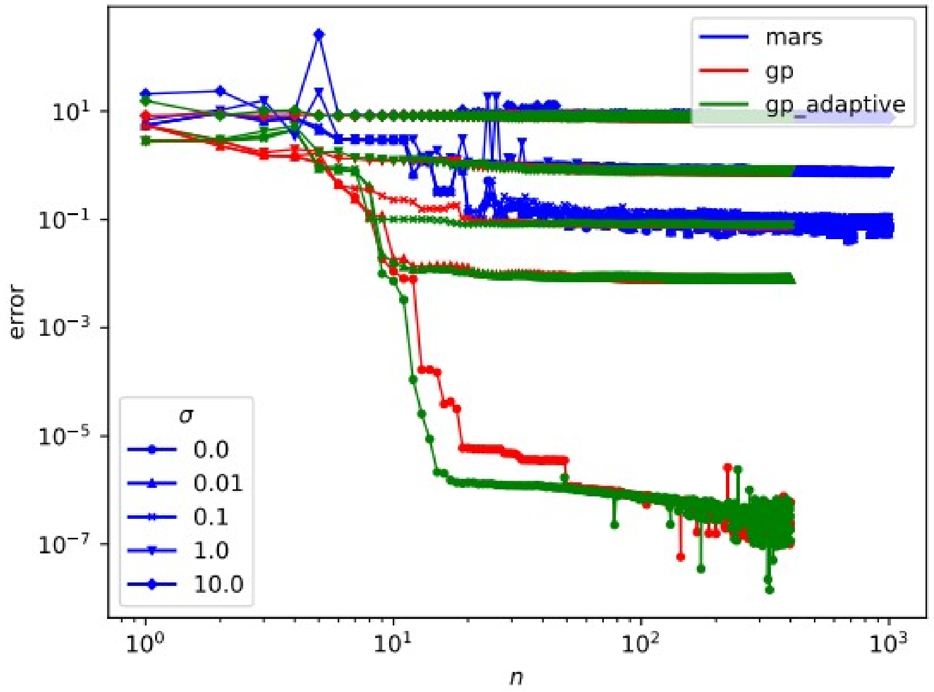


Fitting a smooth function with noise

- **where , random sampling points**

- Do NOT allow GPs to interpolate! Set kernel parameter , i.e. generally need to approximate variance of data's random noise.
- No change to MARS (it doesn't interpolate)





- Noise level sets a lower bound for achievable error
- For noise achieves good fit… use MARS as cheaper

- Caveat: single performance metric studied.

Conclusion

- **Gaussian Processes vs. MARS**

GPs:

- Work:
- Excellent fit to smooth functions
- Noise and bad fits around jumps
- Provides error estimates for fit, makes adaptive sampling easy
- Handle discontinuities with manual detection of jumps and one GP for each continuous section
- With adaptive sampling and jump detection, get very fast convergence

MARS:

- Work:
- Reasonable fit to smooth functions, but has an error floor, never converges to zero error
- No error estimate for fit, no adaptive sampling studied

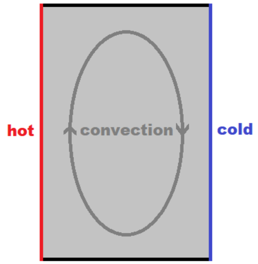
Noise:

- Imposes an error floor,
- Neither GPs nor MARS can converge
- … but MARS is much faster

- See UKAEA report Selection of Techniques for Uncertainty Quantification, ([Documents/CD-EXCALIBUR-FMS0063-M5.2.pdf at main · ExCALIBUR-NEPTUNE/Documents (github.com)](#)).

Nektar++ slot convection (reprise)

- 2D heat transfer problem:

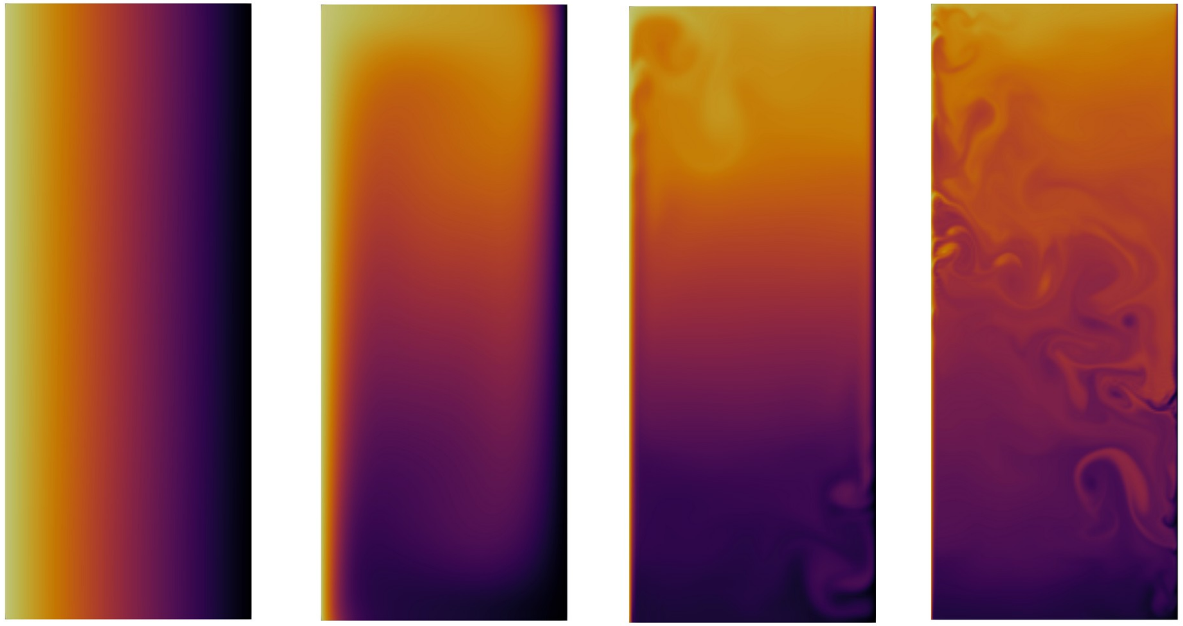


$$\frac{1}{Pr}\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + Ra\, T\, \hat{\mathbf{y}} + \nabla^2\mathbf{u}$$
$$\left(\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T\right) = \nabla^2 T$$
$$\nabla\cdot\mathbf{u} = 0.$$

- Parameter: Rayleigh number : strength of buoyancy force. (; air.)
- Phenomenology: increased gives increased heat transfer rate … diffusion, then steady convection, then 2D turbulence driven by large intermittent boundary fluctuations (cf. tokamak plasma edge).
- Goals: study of system as turbulent heat transport proxy; also investigate capability of Nektar++.

heat flux results vs. literature

Ra	Barletta *et al* (2005)	Lo *et al* (2005)	*Nektar++* (2021)
10^3	1.118	1.118	1.118
10^4	2.245	2.243	2.245
10^5	4.520/4.522	4.519	4.522
10^6	–	8.823	8.826
10^7	–	16.641	16.532

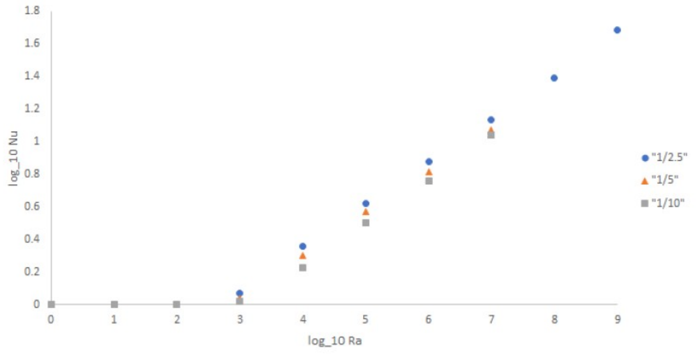


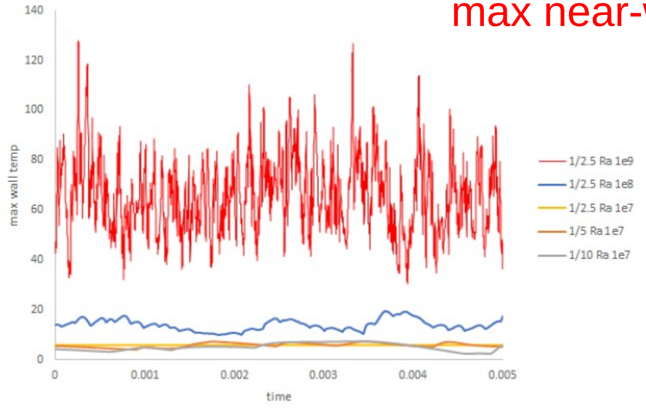
temperature fields for , ,

Nektar++ slot convection (reprise)

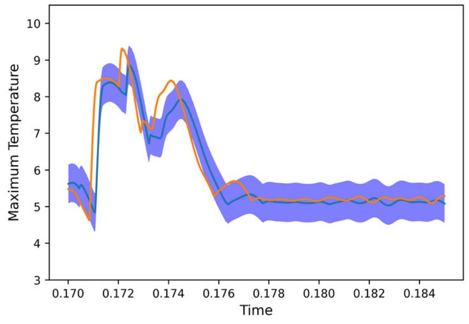
- Small modifications to Nektar++ to compute heat flux, and properties of wall-wave instability.
- Heat flux (Nusselt number): power laws (three different cavity aspect ratios)…

power laws





max near-wall temp fitted by GP (*EasyVVUQ*)

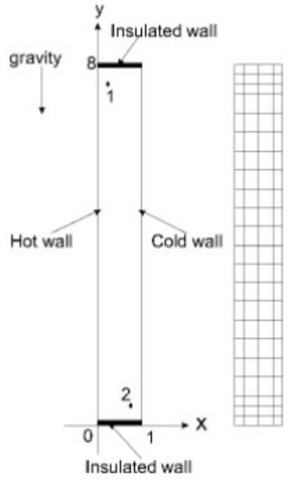


- Wall-wave instability: time series for maximum local temperature and position of hottest point (these were used with *EasyVVUQ* – constructed GP surrogate during hackathons).

- Main conclusions: -refinement advised in preference to -refinement (more efficient). Large- dynamics mandates small time step in solver (also tried out different schemes for `SpectralVanishingViscosity` solver option – did not see a big improvement).

Smallab

- Interesting problem for DA, GP-ROM surrogate ...
- Nektar++ convection problems are relevant (interesting features e.g. transition from time-periodic behaviour to turbulent chaos).
- MIT benchmark problem (2D 8:1 ratio cavity, Rayleigh numbers close to turbulent transition), then move on to 3D examples based on Smallab experiments (Dom Buta, Wayne Arter) - implemented in FabNEPTUNE. 2D case easy as accessible to single PC – turbulent 3D not. Video of Dom Buta initial results presentation available.

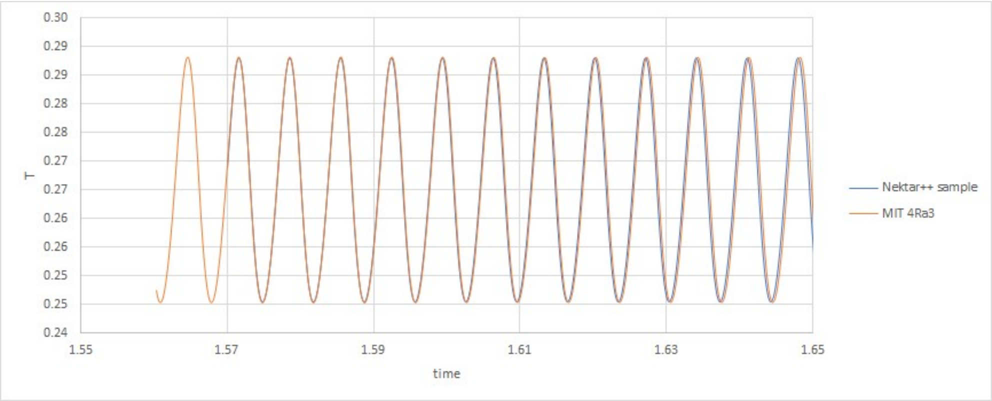


2D MIT benchmark

(QoIs)

1. The x-velocity at point 1
2. The temperature at point 1
3. The Nusselt number along both sides of the wall in an integral valued function,

$$Nu(t) = \frac{1}{H}\int_0^H \left|\frac{\partial \Theta}{\partial x}\right|_{x=0,W} dx$$



Smallab apparatus (Buta)

From Elder, *Laminar free convection in a vertical slot*, J. Fluid Mech (1965) vol.23, pt.1