SCIENCE
PASSION
TECHNOLOGY

Parameter learning with physics-consistent Gaussian Processes

J. Moser, C. Albert, S. Ranftl

MaxEnt Conference 2025

Gaussian Processes

$$f \quad GP(m(\cdot), k(\cdot,\cdot))$$

$$x \quad X : f(x) \quad N(m(x), k(x,x))$$

$$\begin{matrix} k(x_1,x_1) & \dots & k(x_1,x_N) \\ \dots & & \dots \\ k(x_N,x_1) & \dots & k(x_N,x_N) \end{matrix}$$

$$f \quad GP(m(\cdot), K(\cdot,\cdot))$$

$$f(x) \quad N(m(x), K(x,x))$$

$$\begin{matrix} k_{11}(x,x) & k_{12}(x,x) \\ k_{21}(x,x) & k_{22}(x,x) \end{matrix}$$

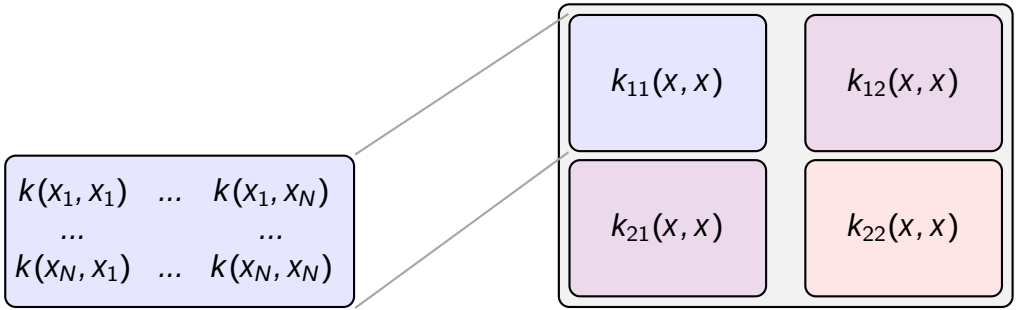
J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Gaussian Processes

$$f \quad GP(m(\cdot), k(\cdot,\cdot)) \qquad f \quad GP(m(\cdot), K(\cdot,\cdot))$$

$$x \quad X : f(x) \quad N(m(x), k(x,x)) \qquad f(x) \quad N(m(x), K(x,x))$$



J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Gaussian Processes [1]

$$f \quad GP(m(\cdot), k(\cdot,\cdot)) \qquad\qquad f \quad GP(m(\cdot), K(\cdot,\cdot))$$

$$x \quad X : f(x) \quad N(m(x), k(x,x)) \qquad\qquad f(x) \quad N(m(x), K(x,x))$$

posterior

$$f_{|X,Y} \quad GP(m_{|X,Y}(\cdot), k_{|X,Y}(\cdot,\cdot))$$

$$m_{|X,Y}(x) = m(x) - k(x,X)k(X,X)^{-1}Y$$

$$k_{|X,Y}(x,y) = k(x,y) + k(x,X)k(X,X)^{-1}k(X,y)$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Physics informed Gaussian Processes (PIGP) [2]

$$L_x f(x) = u(x)$$

$$K(x,y) = \begin{bmatrix} k_0(x,y) & L_y k_0(x,y) \\ L_x k_0(x,y) & L_x L_y k_0(x,y) \end{bmatrix}$$

$$\begin{bmatrix} f \\ u \end{bmatrix} \sim GP\left(0, K(\cdot,\cdot)\right)$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Physics informed Gaussian Processes (PIGP) [2]

$$L_x f(x) = u(x)$$

$$\begin{bmatrix} f \\ u \end{bmatrix} \quad GP\left(0, K(\cdot,\cdot)\right)$$

$$K(x,y) = \begin{bmatrix} k_0(x,y) & L_y k_0(x,y) \\ L_x k_0(x,y) & L_x L_y k_0(x,y) \end{bmatrix}$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Null-space based (NSB) Gaussian Processes [3]

$$L_x f(x) = u(x) \Leftrightarrow \begin{bmatrix} L_x & -1 \end{bmatrix} \begin{bmatrix} f \\ u \end{bmatrix} = 0$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Null-space based (NSB) Gaussian Processes [3]

$$A_x \xi = 0$$

$$A_x B_x = 0$$

B_x...Null-space matrix

$$\xi \quad GP(0, B_x B_y^T k_0(x, y))$$

$$K(x, y) =$$

$$k_0(x, y)$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

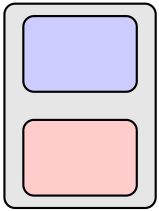
Null-space based (NSB) Gaussian Processes [3]

$$A_x \xi = 0$$

$$A_x B_x = 0$$

B_x...Null-space matrix

$\xi \quad GP(0, B_x B_y^T k_0(x, y))$

$K(x, y) =$  $k_0(x, y)$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Solving a complicated PDE

$$\nabla^2 f_1(x,y) + 30\cos(2x) f_1(x,y) + 50(y+1)^3 f_2(x,y) = 50e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09}}$$

$$\nabla^2 f_2(x,y) + 10 f_1(x,y) + f_2(x,y) = 0$$

$$A\xi = \begin{bmatrix} \nabla^2 + a & b & -1 \\ c & \nabla^2 + d & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ 50e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09}} \end{bmatrix} = 0$$

Macaulay2 [4], OreModules [5]...

$$B = \begin{bmatrix} d + \nabla^2 \\ -c \end{bmatrix}$$

$$-bc + ad + (a+d)\nabla^2 + (\nabla^2)^2$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Solving a complicated PDE

$$\partial^2 f_1(x,y) + 30\cos(2x) f_1(x,y) + 50(y+1)^3 f_2(x,y) = 50e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09}}$$

$$\partial^2 f_2(x,y) + 10 f_1(x,y) + f_2(x,y) = 0$$

$$A\xi = \begin{bmatrix} \partial^2 + a & b & -1 \\ c & \partial^2 + d & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 50e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09}} \end{bmatrix}$$

$$B = \begin{bmatrix} d + \partial^2 \\ -c \end{bmatrix}$$

$$-bc + ad + (a+d)\partial^2 + (\partial^2)^2$$

Macaulay2 [4], OreModules [5]...

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Solving a complicated PDE

$$\nabla^2 f_1(x,y) + 30\cos(2x) f_1(x,y) + 50(y+1)^3 f_2(x,y) = 50 e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09}}$$
$$\nabla^2 f_2(x,y) + 10 f_1(x,y) + f_2(x,y) = 0$$

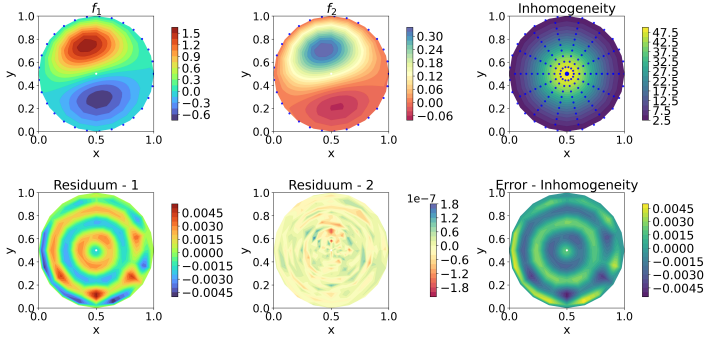
- Build kernel
- Condition GP on inhomogeneity and boundary conditions
- Add nugget (= likelihood standard deviation): $\sigma \approx 0.008$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics
Package available at: https://github.com/moserjo/PCGP

Solving a complicated PDE

$$\nabla^2 f_1(x,y) + 30\cos(2x) f_1(x,y) + 50(y+1)^3 f_2(x,y) = 50e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09}}$$

$$\nabla^2 f_2(x,y) + 10 f_1(x,y) + f_2(x,y) = 0$$

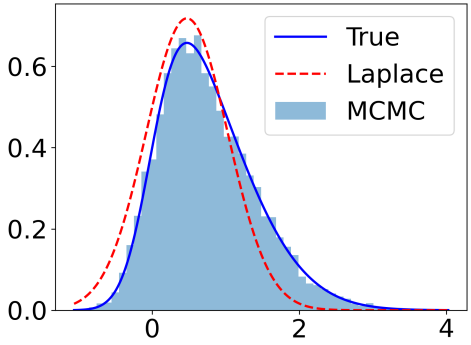


J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Distribution of hyperparameters

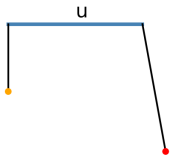
- Parameters in equation
- Parameters in kernel
- Hyperparameters of GP
- Gradient-based optimization → Laplace approximation
- MCMC sampling

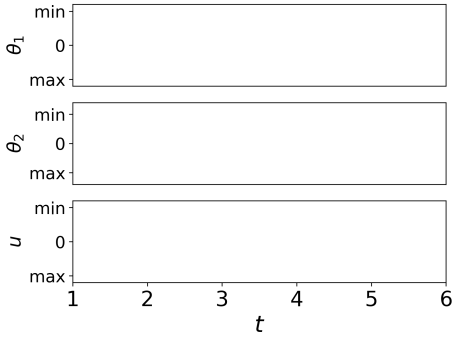


J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics
Package available at: https://github.com/moserjo/PCGP

Parameter estimation: Bipendulum

$$_1\frac{d^2}{dt^2}\theta_1(t) + g\theta_1(t) = u(t) \qquad _2\frac{d^2}{dt^2}\theta_2(t) + g\theta_2(t) = u(t)$$



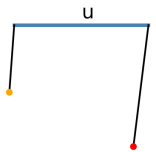


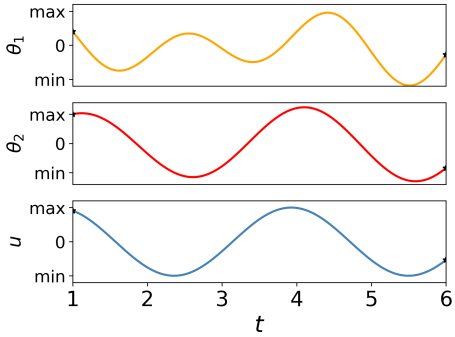
J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Parameter estimation: Bipendulum

$$1\frac{d^2}{dt^2}\theta_1(t) + g\theta_1(t) = u(t) \qquad 2\frac{d^2}{dt^2}\theta_2(t) + g\theta_2(t) = u(t)$$





J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Comparison of PIGP and NSB

$$ {}_1\frac{d^2}{dt^2}\theta_1(t) + g\theta_1(t) = u(t) \qquad {}_2\frac{d^2}{dt^2}\theta_2(t) + g\theta_2(t) = u(t) $$

PIGP

$$ L = \begin{bmatrix} {}_1\partial_t^2 + g & 0 \\ 0 & {}_2\partial_t^2 + g \end{bmatrix} $$

$$ B_{PIGP} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ {}_1\partial_t^2 + g & 0 \\ 0 & {}_2\partial_t^2 + g \end{bmatrix} $$

NSB

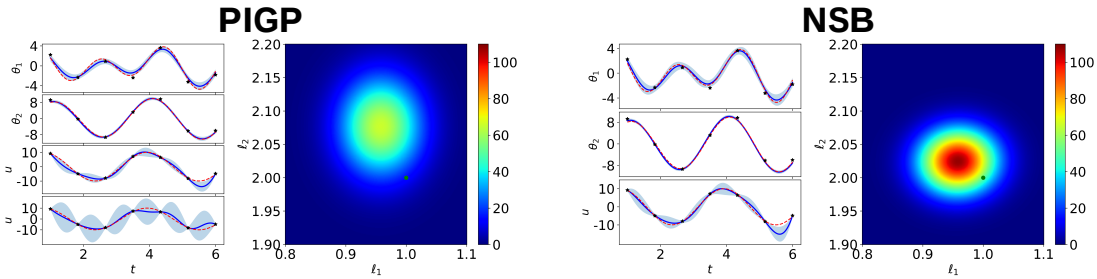
$$ A = \begin{bmatrix} {}_1\partial_t^2 + g & 0 & -1 \\ 0 & {}_2\partial_t^2 + g & -1 \end{bmatrix} $$

$$ B = \begin{bmatrix} & - {}_2\partial_t^2 - g & \\ & - {}_1\partial_t^2 - g & \\ - {}_1 {}_2\partial_t^4 - g & {}_1\partial_t^2 - g & {}_2\partial_t^2 - g^2 \end{bmatrix} $$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Comparison of PIGP and NSB

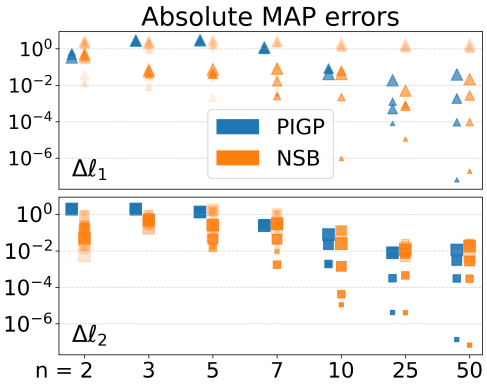


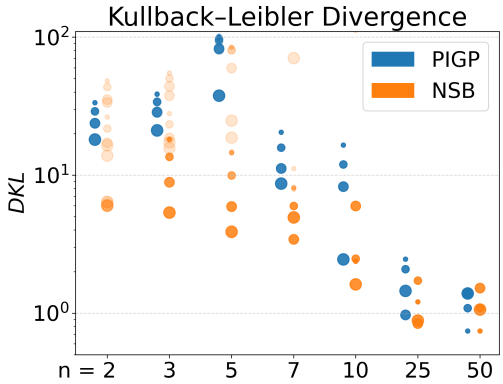
J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Comparison of PIGP and NSB

$$D_{KL}(P||Q) := \sum_x P(x) \log \left[\frac{P(x)}{Q(x)} \right]$$





J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Application

- GP scales badly with much data: problem for timeseries
- No direct way of learning non-constant hyperparameters
- We don't need much data: solve both problems with window approach and as a bonus: seamlessly integrate prior assumptions on parameter behaviour!

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics
Package available at: https://github.com/moserjo/PCGP

Example: Predictive maintenance

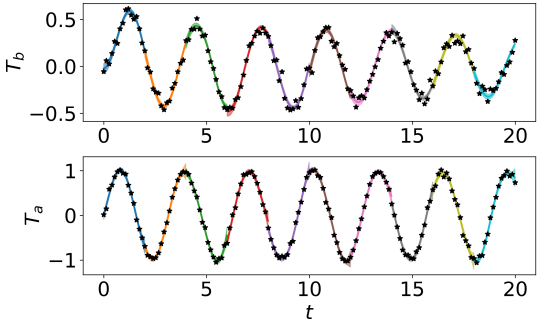
$$\frac{dT_b}{dt} = R^{-1}\left(T_a(t) - T_b(t)\right)$$

J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Example: Predictive maintenance

$$\frac{dT_b}{dt} = R^{-1}\left(T_a(t) - T_b(t)\right)$$

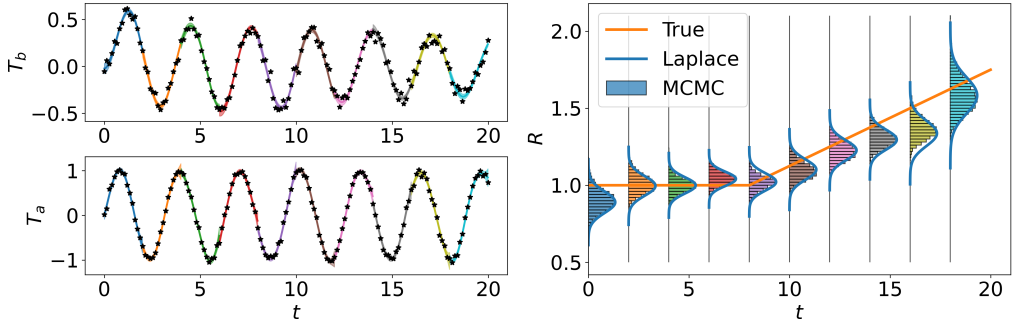


J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP

Example: Predictive maintenance

$$\frac{dT_b}{dt} = R^{-1}\left(T_a(t) - T_b(t)\right)$$



J. Moser, C. Albert, S. Ranftl, Institute of Theoretical Physics and Computational Physics

Package available at: https://github.com/moserjo/PCGP