

Practical ODE modelling

Ben Lambert¹

¹University of Oxford

Monday 8th February, 2021

Lecture outcomes

- ① Explain difficulty with inference for ODEs;
- ② Inference with PINTS and other libraries;
- ③ PINTS demo.

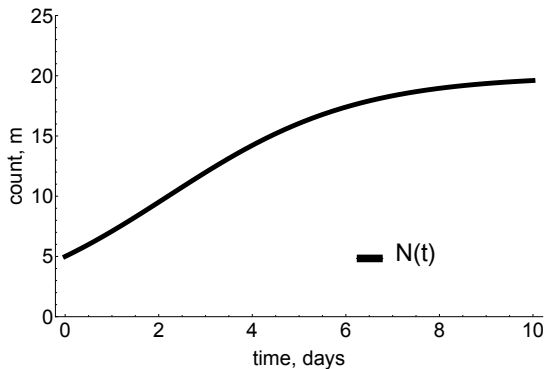
1 ODE inference refresh

2 Issues with ODE inference

3 Inference cycle

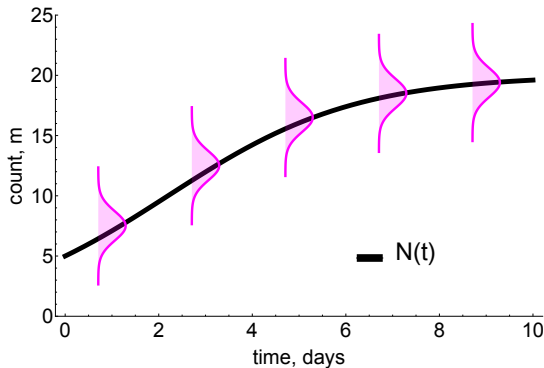
Setting up the inference problem: forward model

$$\frac{dy(t)}{dt} = f(y, t; \theta). \quad (1)$$



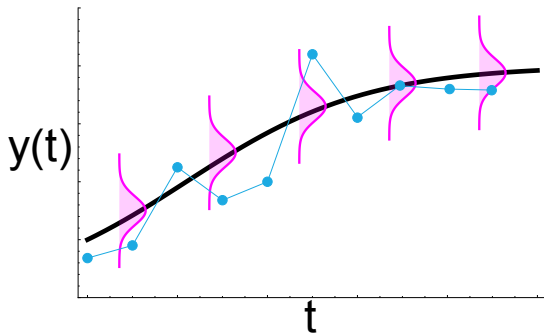
Setting up the inference problem: noise model

$$y^*(t) \stackrel{i.i.d.}{\sim} \mathcal{N}(y(t; \theta), \sigma). \quad (2)$$



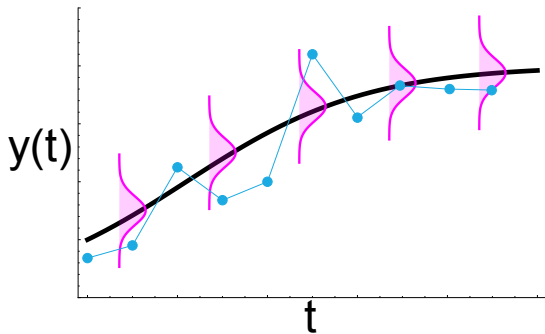
Setting up the inference problem: data

$$\mathcal{L} = \prod_{i=1}^S \mathcal{N}(y^*(t_i) | y(t_i; \theta), \sigma). \quad (3)$$



Setting up the inference problem: posterior

$$p(\theta|X) \propto p(\theta, \sigma) \prod_{i=1}^S \mathcal{N}(y^*(t_i)|y(t_i; \theta), \sigma). \quad (4)$$

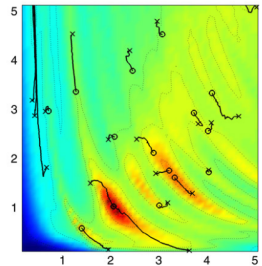
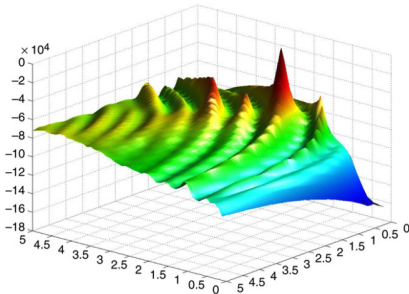


- 1 ODE inference refresh
- 2 Issues with ODE inference
- 3 Inference cycle

Why is inference for ODEs hard?

- ODEs (typically) require numerical solutions \implies expensive;
- Non-linearity \implies posterior distributions can be complex / unidentified.

Why is inference for ODEs hard?



from “Bayesian inference for differential equations”, Girolami (2008)

How to make your life easier

Before you start inference:

- Fake data simulation followed by inference: make simulated data as similar to real data characteristics as possible;
- Profile likelihood method to assess identifiability;
- General mathematical analysis: assess sensitivity of outputs to parameters.

- 1 ODE inference refresh
- 2 Issues with ODE inference
- 3 Inference cycle

How to fit model to data?

Read statistical literature

Journal of statistical research for statisticians, (3), 2020

vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 \Phi}{\partial t^2}$$

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit



Understand method?

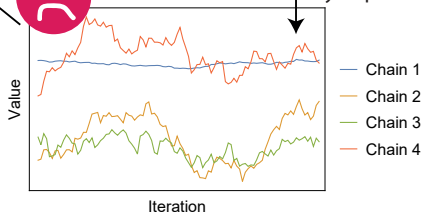
Code method

```
class NewMethod(GaussianMixture):  
    def __init__(self):  
        # not sure what these init code blocks do  
        x = [complex_function(a) for a in A]  
        y = [evaluate(x) for x in A]  
        # ...  
        return self  
    def complex_function(x):  
        # follows lines 2-35 in main text of ref [1]  
        # the line below may be a mistake?  
        x = another_complex_fn(x)  
        # ...
```



Try implementation

Method fails?



Why does this cycle of misery persist?

- Statistical literature is written by methods experts for other experts;
- Statistics papers do not often contain high quality pseudocode;
- Software accompanying papers is typically not professionally developed;
- Software is typically idiosyncratic;
- Different problems require different solutions but solutions currently require reinventing the wheel;
- Not enough knowledge sharing between applied researchers about “Which method works best?”.

What is PINTS?

PINTS: **P**robabilistic **I**nference for **N**oisy **T**ime **S**eries, which covers two broad categories of inference methods:

- Optimisation: single set of parameters returned;
- Sampling: many sets returned.

It's an open-source Python library available on Github.

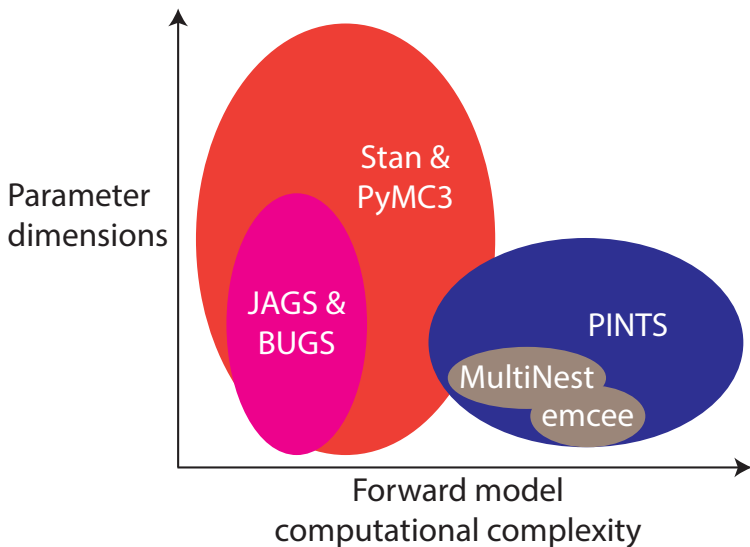
What does existing literature do?

- Most other inference software pin their hope on a single sampler:
 - BUGS and JAGS: Gibbs sampler;
 - PyMC3 and Stan: No U-Turn Sampler (NUTS);
- Other libraries prepackage forward model solution methods with sampling method.

How is PINTS different?

- PINTS not aligned to a single algorithm;
- PINTS is designed to interface with (other) probabilistic programming languages (for example, Stan);
- PINTS aimed at harder forward models: ODEs and PDEs;
- PINTS allows users freedom to use own forward model solution method.

PINTS niche



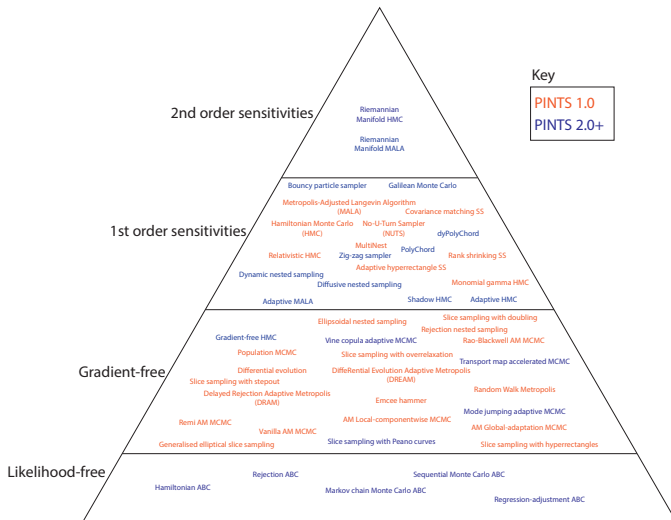
Deciding on an approach

- For cheap forward model \implies use Stan (you can always use PINTS' Stan interface if this doesn't work);
- For expensive forward models, potentially requiring bespoke solvers \implies use PINTS.

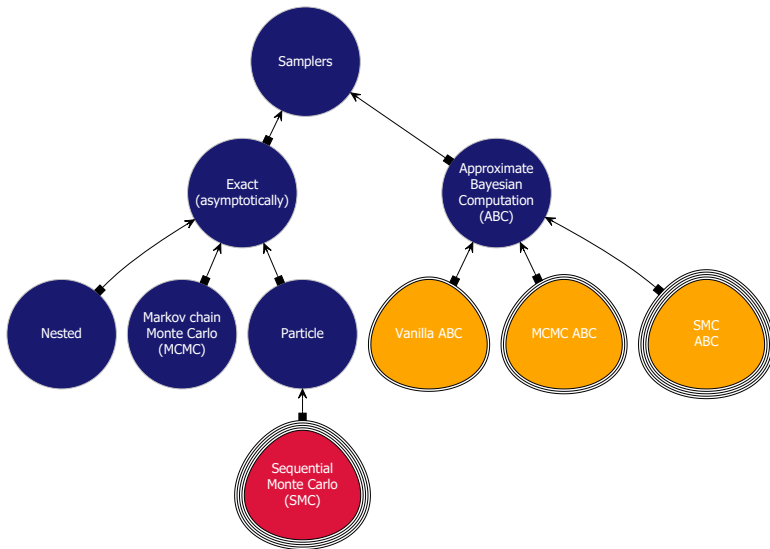
PINTS in detail

- Literature emphasis on developing new algorithms;
- Literature dense with hard-to-decipher algorithm details;
- PINTS aims to make this research **useful**:
 - Common, easy-to-use, interface for lots of methods;
 - Rigorous software development practices;
 - Many levels of testing: unit, functional, and so on;
 - Collaboration with statisticians working on various methods;
 - Benchmark problems;
 - Pseudocode and tutorial papers which explain algorithms and their ecosystem.

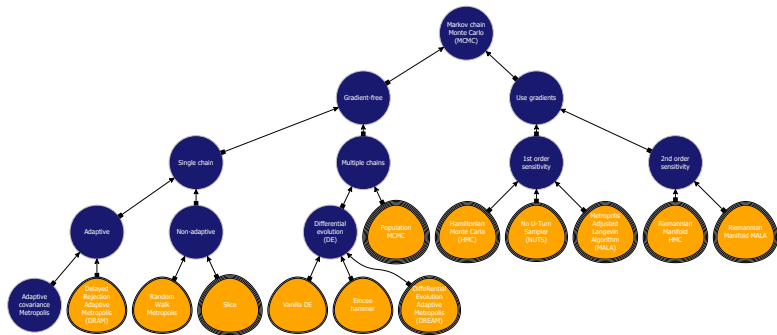
PINTS roadmap: samplers



Families of samplers: overall



Families of samplers: within MCMC



PINTS: optimisers

Optimisation more “solved” than sampling and have two families in PINTS:

- Gradient-free: CMAES, XNES, SNES, Nelder-mead, PSO, SHGO (planned);
- 1st order sensitivities: gradient descent, L-BFGS (planned).

Conclusions

- Inference for ODE models is generally hard;
- Stan and PINTS are your current best bet for inference.