

反问题与数据同化简介

黄政宇

北京大学北京国际数学研究中心

北京大学国际机器学习研究中心



本堂课大纲

➤ 课程内容简介

- 贝叶斯反问题 (Bayesian inverse problems)
- 数据同化 (Data assimilation)
- 扩散模型 (Diffusion model)

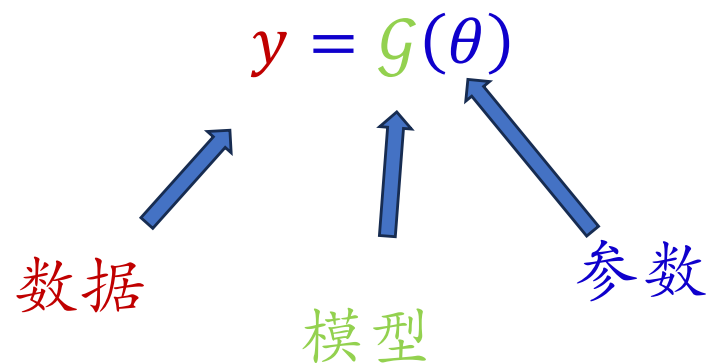
➤ 课程要求

- 作业
- 期末报告



反问题

➤ 抽象的数学形式



找到最“好”的 θ

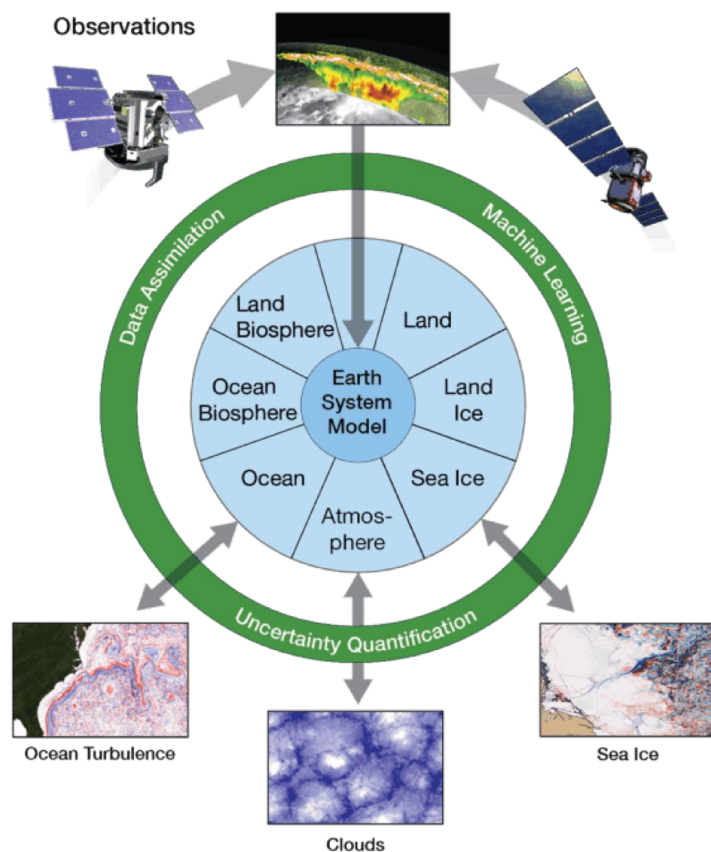


反问题

➤ 应用：气候模拟校准

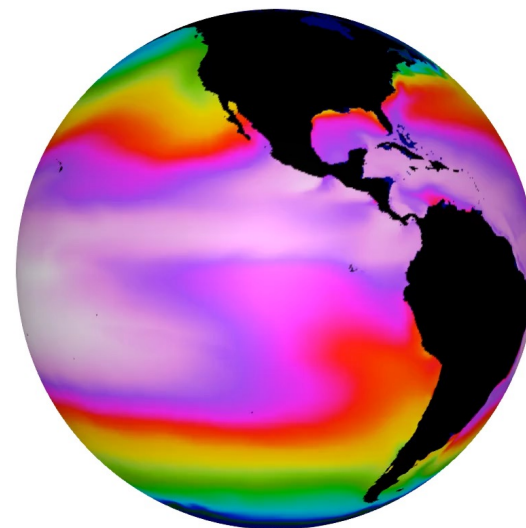
$$y = g(\theta)$$

数据 y



湍流、冰川等模型参数 θ

模型 g



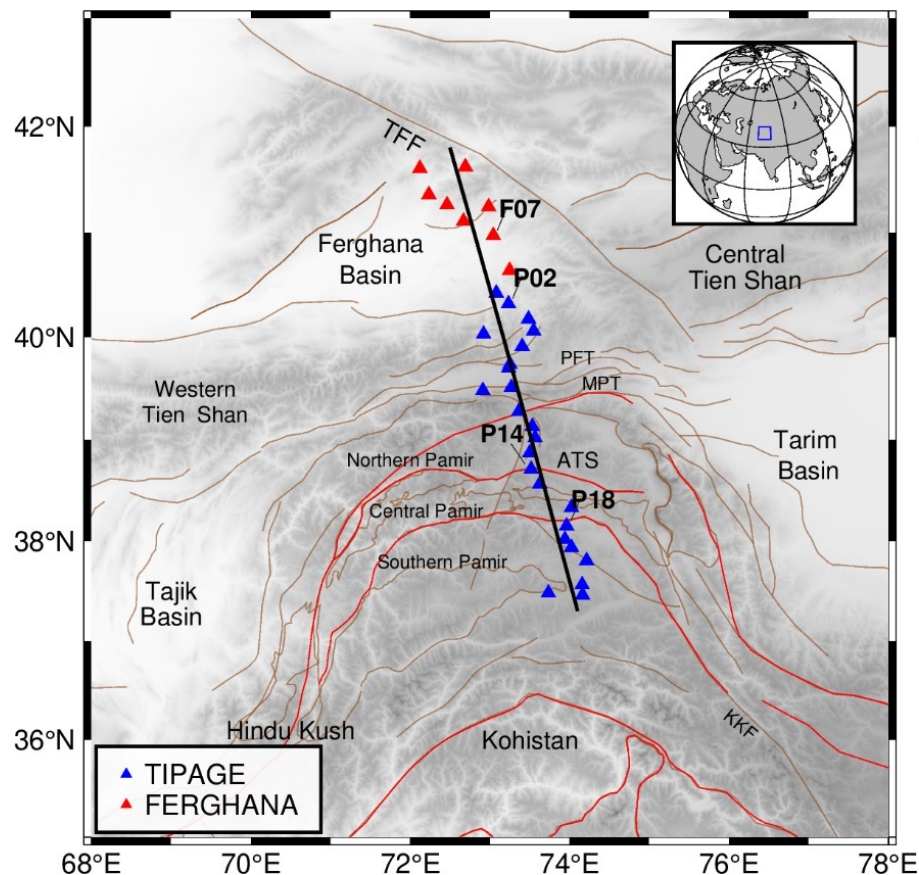


反问题

➤ 应用：地质勘探

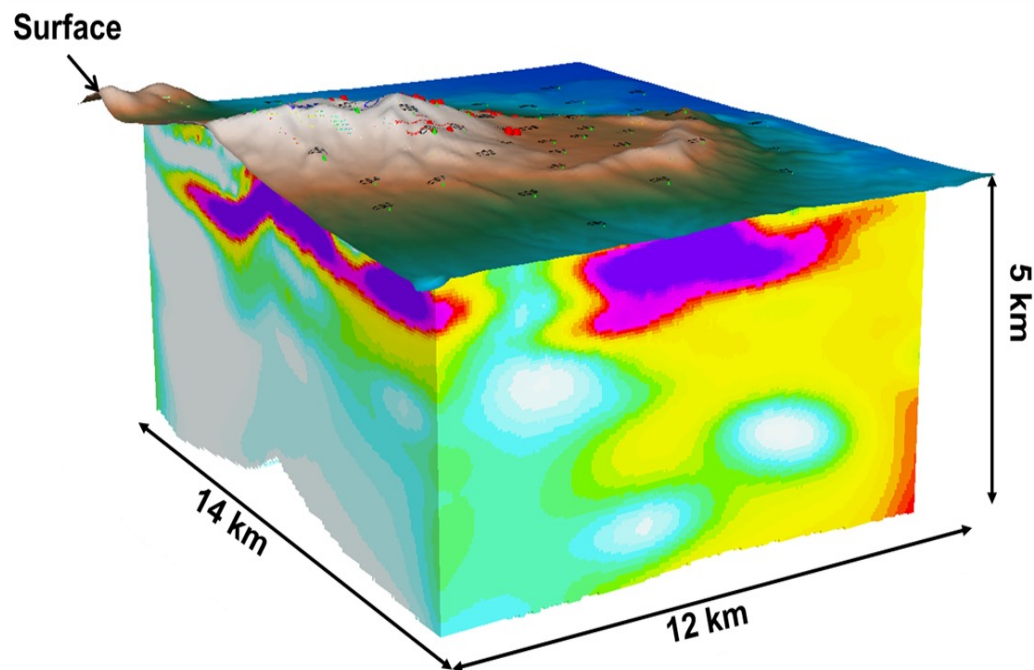
$$y = g(\theta)$$

数据 y



模型 g

地质参数 θ



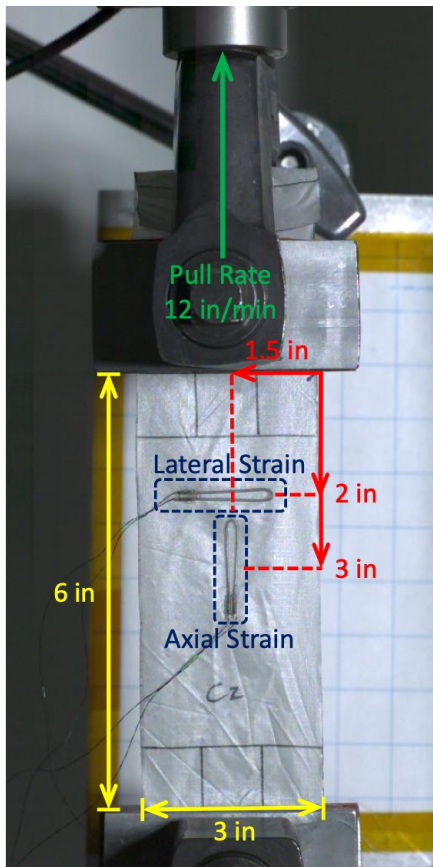


反问题

➤ 应用：材料力学

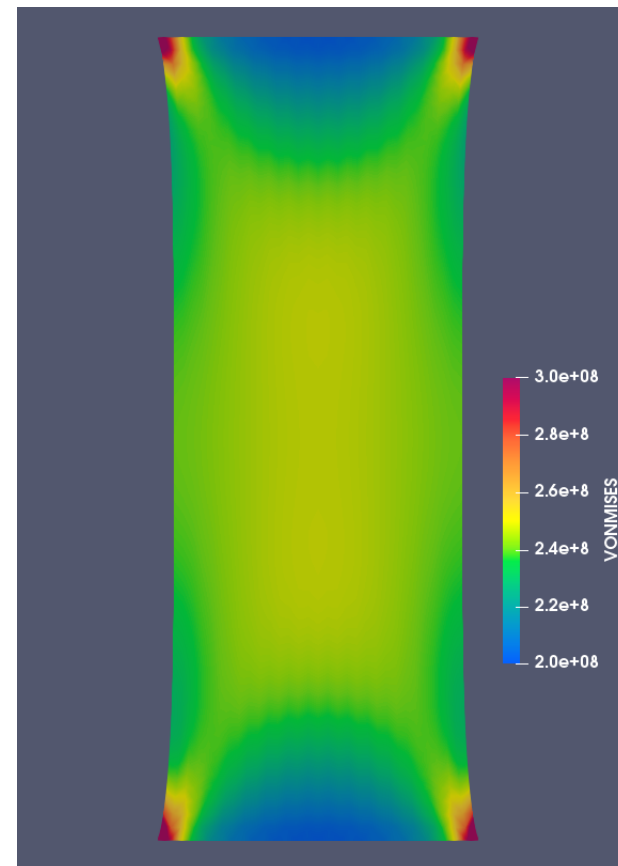
$$y = g(\theta)$$

数据 y



材料参数 θ

模型 g





反问题

➤ 应用：数字孪生

$$y = g(\theta)$$

数据 y

Making Bones: Air Force to develop 'digital twin' of B-1 for damage prediction

By [Jared Morgan](#)

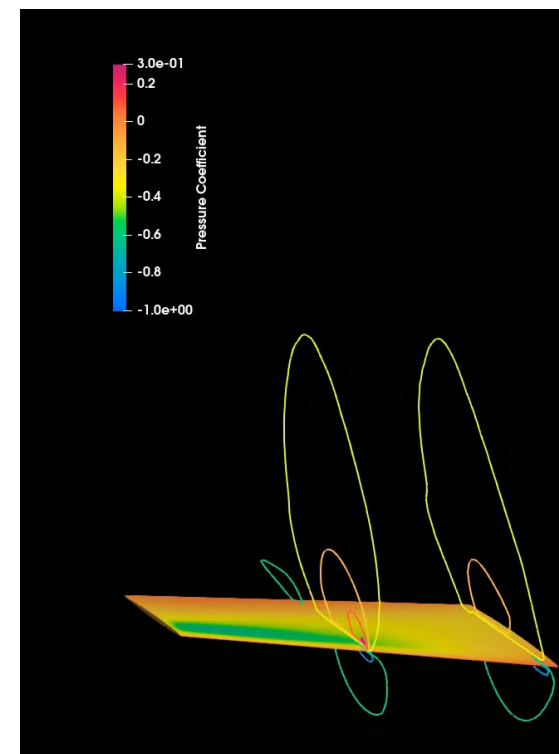
Oct 10, 2020



Two B-1B Lancers, assigned to the 28th Bomb Wing from Ellsworth Air Force Base, S.D., conduct a flyover before landing at Andersen Air Force Base, Guam. (Airman 1st Class Christina Bennett/Air Force)

模型 g

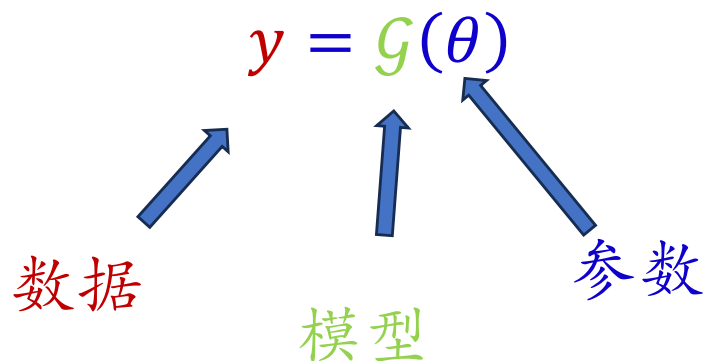
损伤参数 θ





反问题

➤ 抽象的数学形式



$$\theta = [\theta_1; \theta_2] \quad g(\theta) = \theta_1 + \theta_2$$

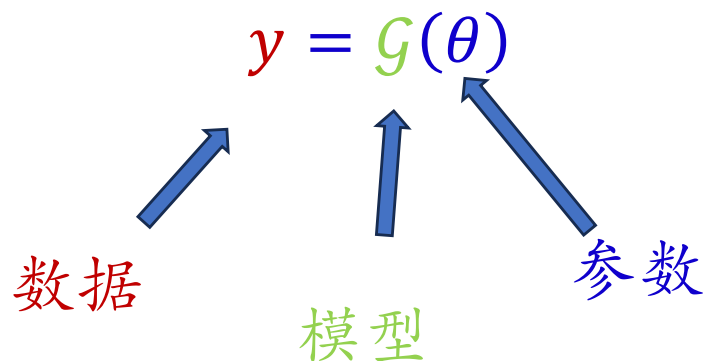
$$y = 1$$

θ 是多少？



反问题

➤ 抽象的数学形式



$$\theta = [\theta_1; \theta_2] \quad g(\theta) = \theta_1 + \theta_2$$

$$y \propto \mathcal{N}(1, 0.1^2)$$

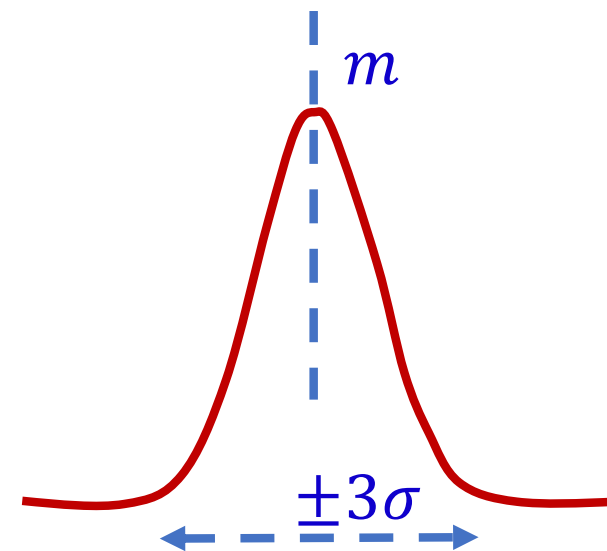
θ 是多少 ?



高斯误差

➤ 高斯分布

$$\mathcal{N}(x; m, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - m)^2}{2\sigma^2}\right)$$



$$\mathcal{N}(x; m, C) = \frac{1}{Z} \exp\left(-\frac{1}{2} (x - m)^T C^{-1} (x - m)\right)$$

$$Z = \sqrt{|2\pi C|} = (2\pi)^{N_{\theta}/2} \sqrt{|C|}$$



贝叶斯反问题

➤ 抽象的数学形式

$$y = g(\theta) + \eta$$

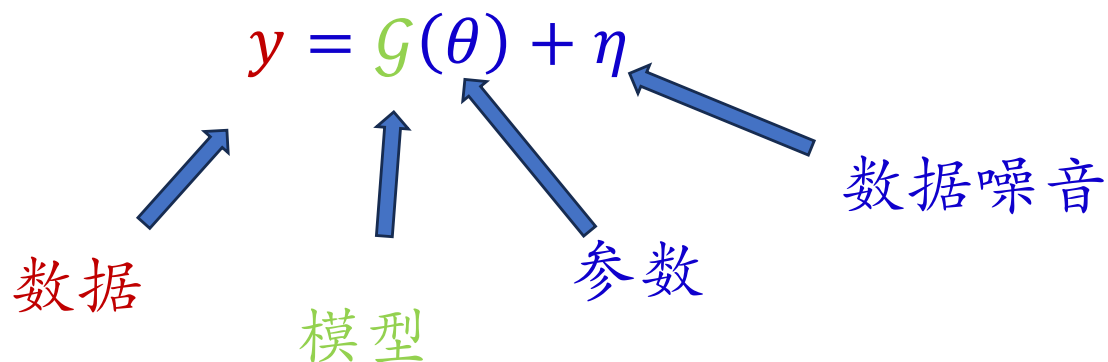
数据 模型 参数 数据噪声





贝叶斯反问题

➤ 抽象的数学形式



$$\theta = [\theta_1; \theta_2] \quad g(\theta) = \theta_1 + \theta_2$$

$$y = 1 \quad \eta \propto \mathcal{N}(0, 0.1^2)$$

$\theta|y$ 的后验分布是什么？



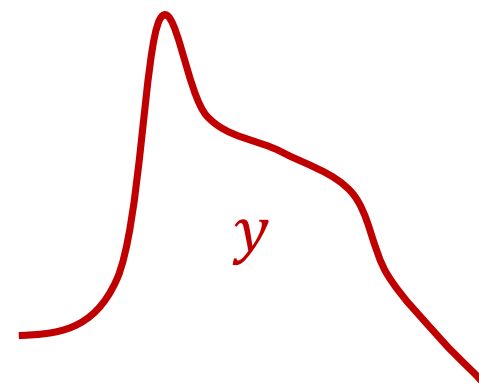
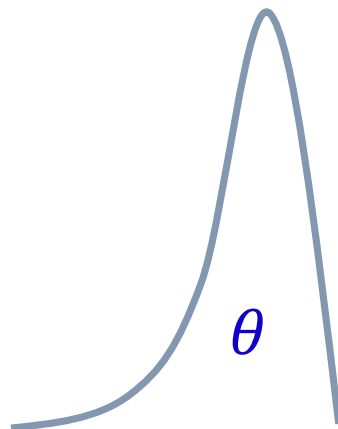
不确定性量化

➤ 抽象的数学形式

$$y = g(\theta) + \eta$$

数据 模型 参数 数据噪音

$g(\theta; \text{新的边界条件、初始条件等}) \longrightarrow \text{新的预测}$



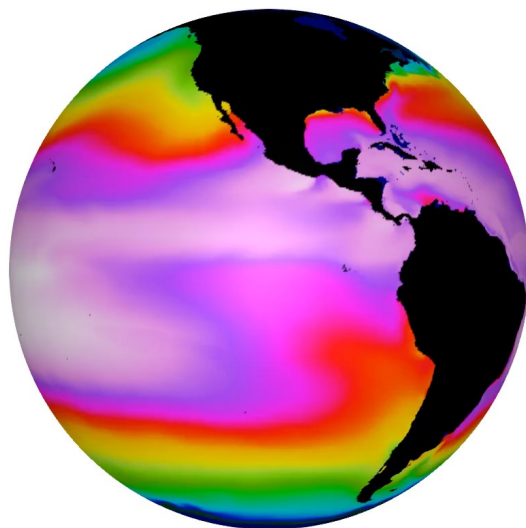
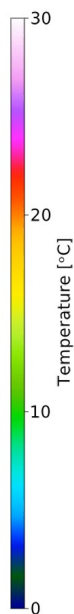


不确定性量化

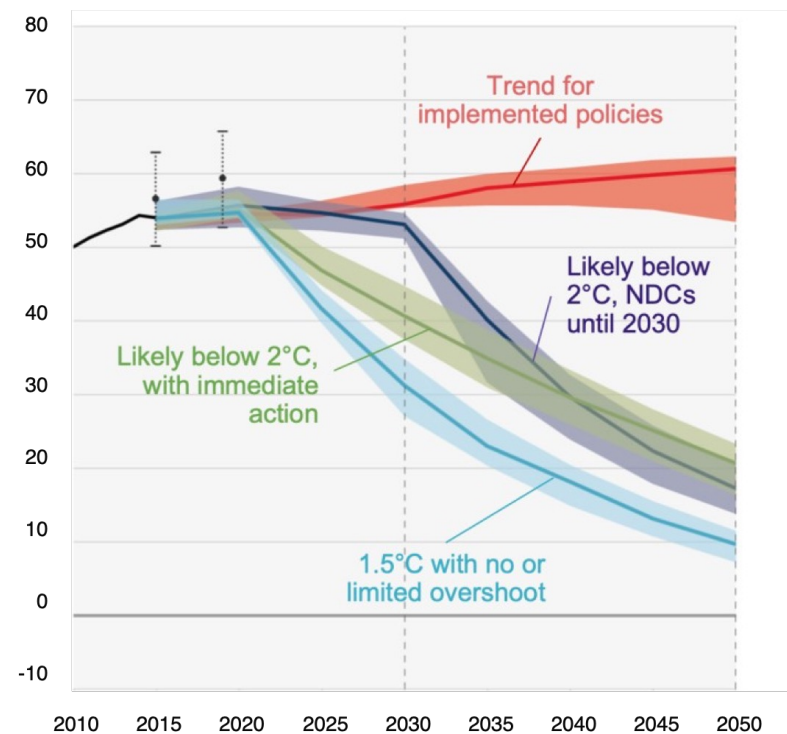
➤ 应用：气候模拟校准

湍流、冰川等模型参数 θ

模型 G



Global GHG Emissions (Gt CO₂-eq. yr⁻¹)



IPCC AR 6, Working Group III



贝叶斯反问题

➤ 基本理论

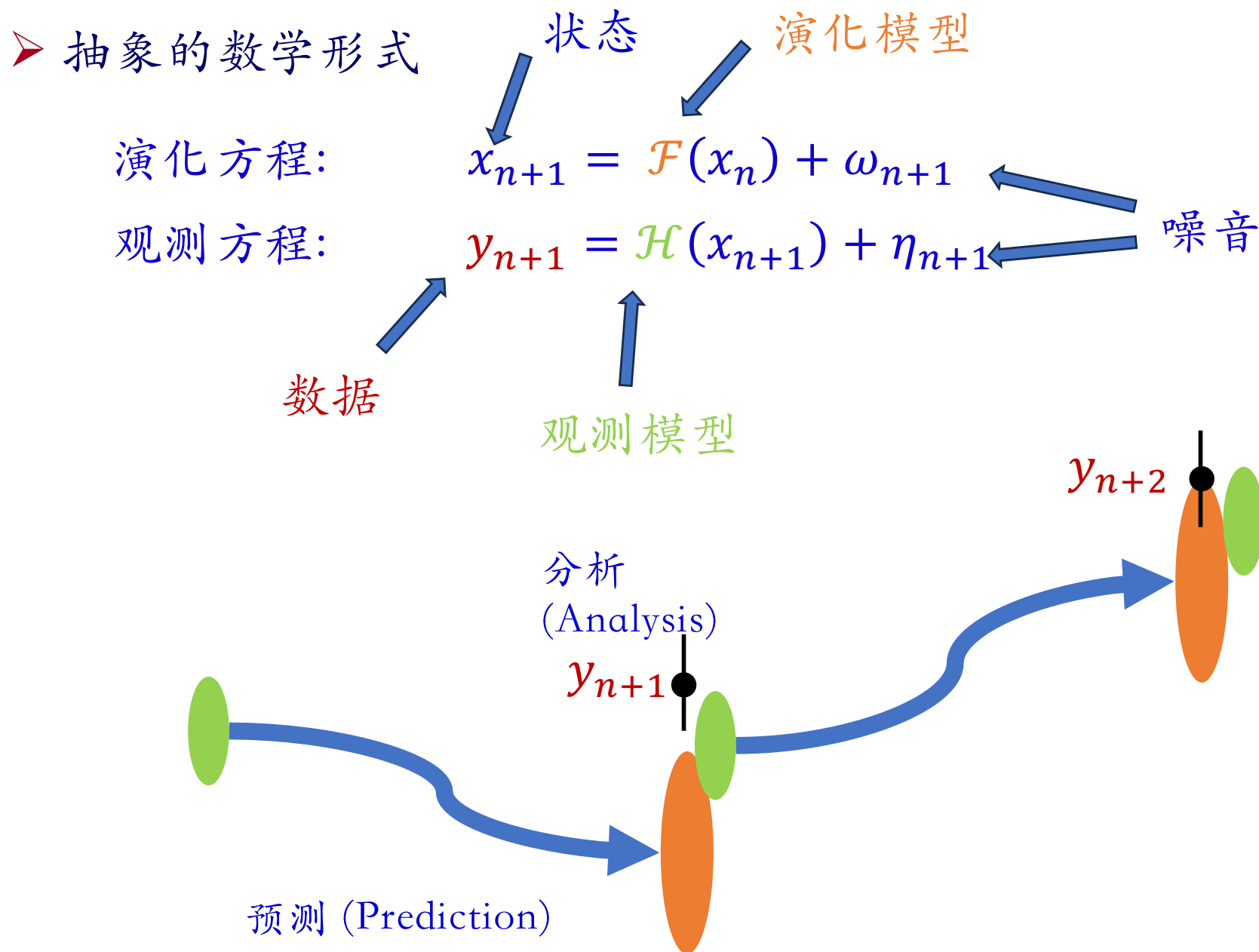
- 贝叶斯反问题的适定性
- 线性贝叶斯反问题 (显示解)

➤ 算法以及相关理论

- 基于输运的方法
- 马氏链蒙特卡洛方法
- 变分推理方法



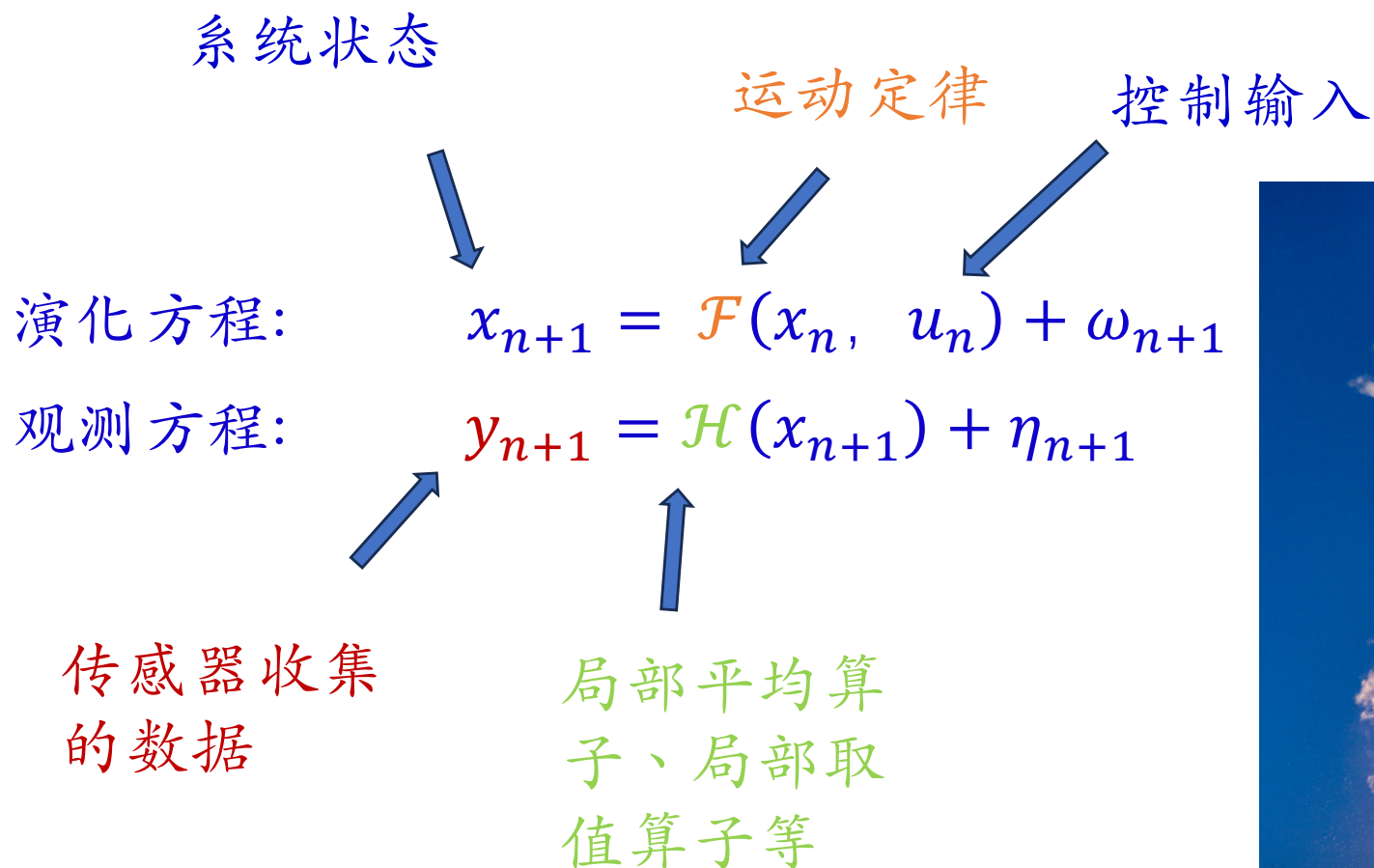
数据同化





数据同化

➤ 应用：系统控制（无人机、航天器、机器人等）





数据同化

➤ 应用：数值天气预报

全球的空气湿度、
风速、温度等状态

守恒律方程

演化方程:

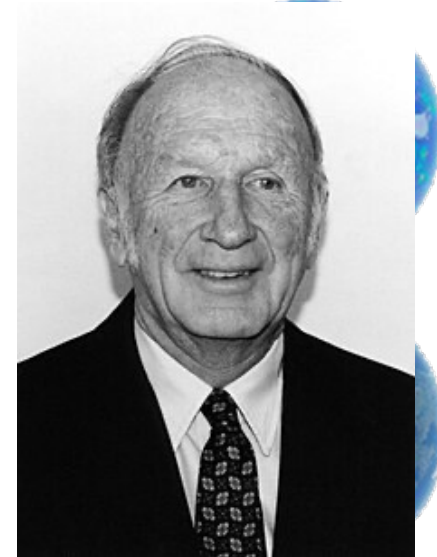
$$x_{n+1} = \mathcal{F}(x_n) + \omega_{n+1}$$

观测方程:

$$y_{n+1} = \mathcal{H}(x_{n+1}) + \eta_{n+1}$$

气象站收集
的数据

局部平均算
子、局部取
值算子等



Lorenz, 1963



数据同化

➤ 应用：股市预测

包括股价、模型参数
的潜在状态

金融模型

演化方程:

$$x_{n+1} = \mathcal{F}(x_n) + \omega_{n+1}$$

观测方程:

$$y_{n+1} = \mathcal{H}(x_{n+1}) + \eta_{n+1}$$

股票价格

计算股价





数据同化

➤ 基本理论

- 贝叶斯滤波 (filter)和平滑 (smooth)问题的适定性
- 线性数据同化问题 (显示解)

➤ 算法以及相关理论

- 卡尔曼滤波 (Kalman filter)
- 卡尔曼平滑 (Kalman smoother)
- 粒子滤波 (Particle filter)



Kálmán, 1960



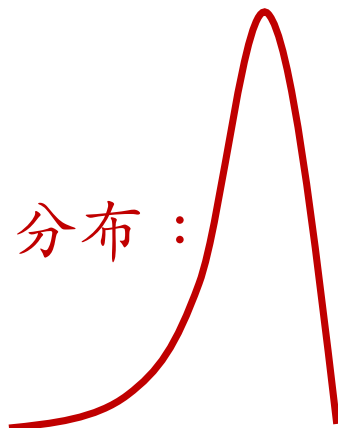
扩散模型

➤ 抽象的数学形式

数据： $\{\theta_i\}$



分布：



生成新的数据： $\{\theta'_i\}$

没有模型！



扩散模型

➤ 应用：视频、音乐生成





扩散模型

➤ 应用：条件生成（反问题）

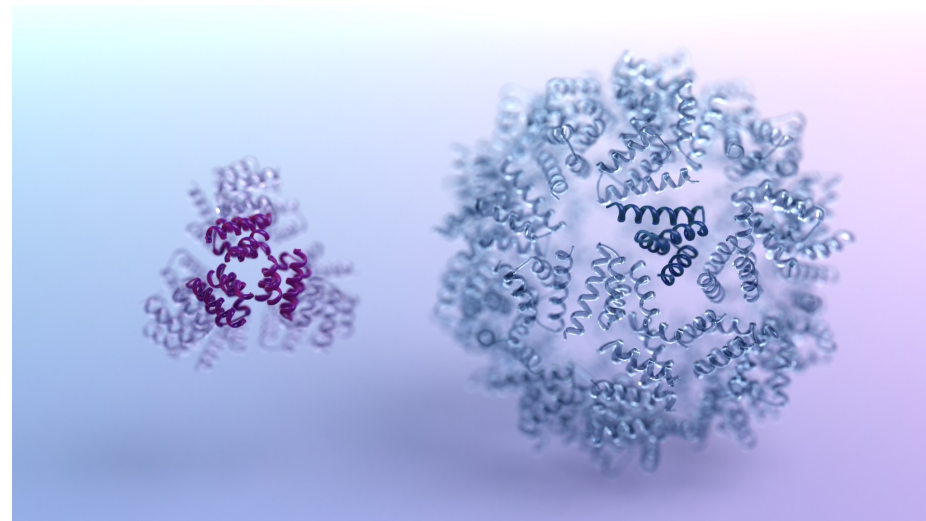


Before inpainting



After inpainting

图像修复



蛋白质设计



扩散模型

➤ 基本理论

- 扩散过程

➤ 算法以及相关理论

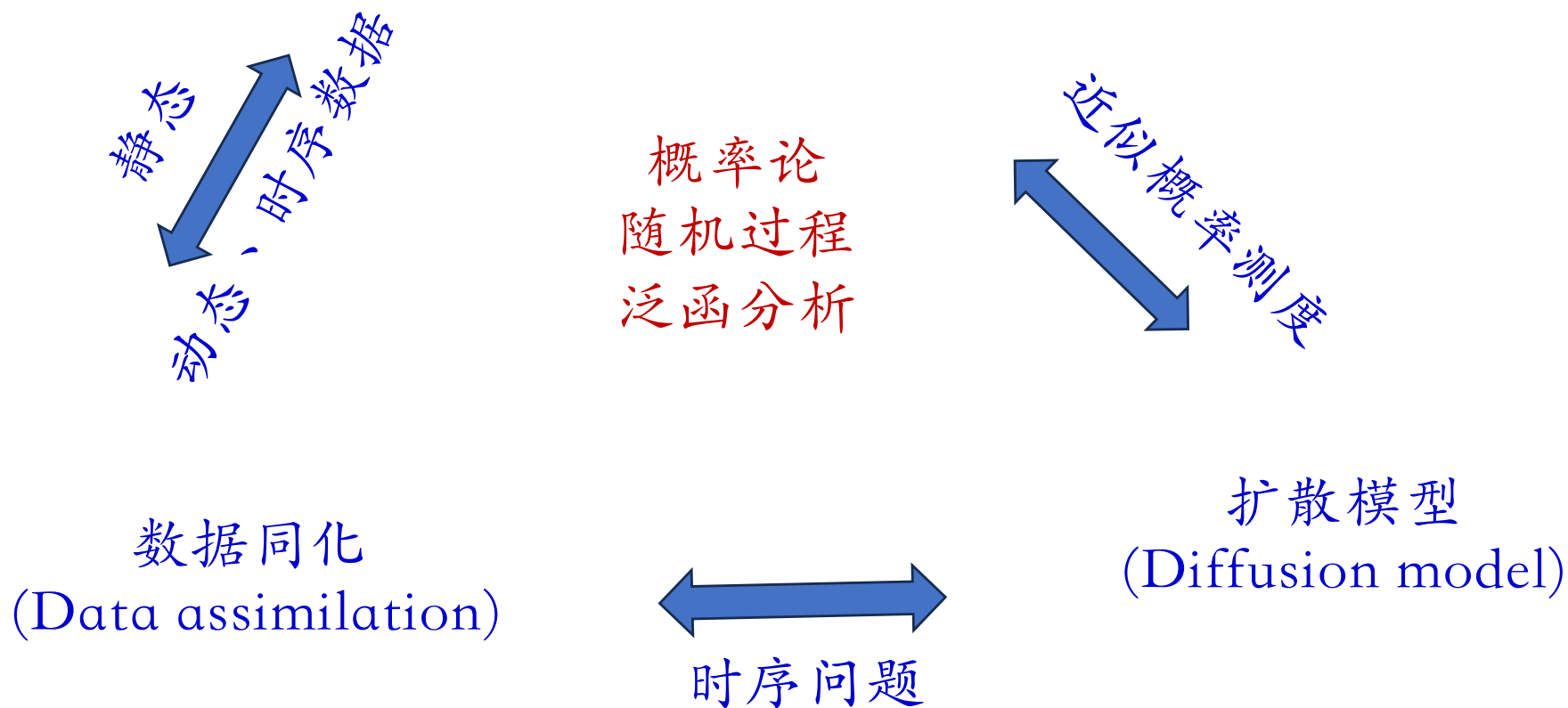
- 分数估计
- 神经网络 UNet
- 基于扩散过程的生成模型



课程简介

➤ 课程内容简介

贝叶斯反问题
(Bayesian inverse problems)





反问题与数据同化

➤ 练习

考虑如下统计模型(比如神经网络)

$$y = f(x, \theta)$$

我们有数据 $(x_i, y_i) \ i = 1, 2, \dots, N$ ，但是 y_i 的误差是 $\mathcal{N}(0, \sigma^2)$ 。

如何用反问题或者数据同化的方法，来更好预测新的 x 对应的 y ?



反问题与数据同化

➤ 练习

考虑如下常微分方程描述的系统

$$\frac{dx}{dt} = f(x, u) \quad x \in R^{N_x}$$

我们知道函数 $f(\cdot, \cdot)$ ，以及在时刻 $t_i = i$ ($i = 1, 2, \dots, N$) 的观测 $x[i \bmod N_x]$ ，即 x 向量的某个分量的信息，观测误差是 $\mathcal{N}(0, \sigma^2)$ 。

如何用反问题或者数据同化的方法，来更好预测时间 t_N 后的系统状态 x ？



课程要求

➤ 作业 (50%)

- 两次作业各占 25%
- 每次作业有理论推导和上机练习

➤ 期末报告 (50%)

- 选一篇相关文章阅读、重复数值实验、报告
- 或做相关科研并报告
- 1-3人一组



课程要求

➤ 课堂

- 会有（上机）练习
- 积极讨论

➤ 关于上机

- Python3（使用Anaconda管理）

<https://www.anaconda.com/download#Downloads>

- Julia

<https://julialang.org/downloads/>

```
julia> a = [1. 2; 3 5]
2×2 Matrix{Float64}:
 1.0  2.0
 3.0  5.0

julia> b = [1.; 2]
2-element Vector{Float64}:
 1.0
 2.0

julia> a\b
2-element Vector{Float64}:
-0.9999999999999999
 0.9999999999999999
```

```
>>> a = np.array([[1, 2], [3, 5]])
>>> b = np.array([1, 2])
>>> x = np.linalg.solve(a, b)
>>> x
array([-1.,  1.])
```



课程要求

➤ 课件

<http://faculty.bicmr.pku.edu.cn/~huangdz/teaching.html>