# 冰盖状态与参数估计器（ICESEE）库（v1.0.0）：冰盖模型的集合卡尔曼滤波

Brian Kyanjo[1], Talea L. Mayo[2], 和 Alexander A. Robel[1]1佐治亚州亚特兰大佐治亚理工学院地球与大气科学学院，美国2埃默里大学数学系，亚特兰大，佐治亚州，美国**通讯作者：** Brian Kyanjo(bkyanjo3@gatech.edu)

arXiv:2603.26947v1 [cs.CC] 27 Mar 2026

**摘要**。ICESEE（冰盖状态与参数估计器）是一个基于 Python 的开源数据同化框架，旨在与冰盖模型及其他地球系统模型无缝集成。ICESEE 实现了支持完整消息传递接口（MPI）的并行集合卡尔曼滤波器（EnKF）架构，可在模型状态和参数空间中实现可扩展的同化。其核心算法采用 Evensen（2003）最初提出的非线性变换式更新方案，无需显式构建预报误差协方差矩阵。这种无矩阵形式避免了局地化的需求，同时在高维非线性系统中保持稳健性。除核心 EnKF 外，ICESEE 还提供四种备选集合卡尔曼滤波变体的串行实现，包括一种局地化形式，为方法测试和比较研究提供灵活性。除状态估计外，ICESEE 还支持通过混合同化—反演策略间接推断未观测或约束较弱的模型参数。在此方法中，基于集合的数据同化利用可用观测校正模型状态，随后应用基于物理的反演方法推断基底摩擦等参数。该框架具有模块化耦合接口、自适应状态索引和高效的并行输入/输出，因此易于扩展到多种数值建模环境。ICESEE 已成功耦合多个现有模型，包括基于 MATLAB/C++的冰盖与海平面系统模型（ISSM）、基于 Firedrake 的 Python 模型 Icepack、一维流线模型以及降阶的 Lorenz–96 系统。本研究重点探讨 ICESEE 在 ISSM 和 Icepack 中的应用，以展示其互操作性、数值性能、可扩展性，以及协同改进状态估计并推断不确定模型参数的能力。性能基准测试显示了其在高性能计算平台上的强扩展和弱扩展能力，凸显了 ICESEE 在支持大陆尺度冰盖演化的长期观测约束再分析方面的潜力。

## 1 引言

冰盖在地球气候系统中发挥着核心作用，调节全球海平面，并影响更广泛的大气和海洋环流模式（Nowicki 等人，2016；Shepherd 等人，2018）。因此，准确模拟冰盖动力学对于预报海平面上升和评估气候变化的长期影响至关重要。然而，传统冰盖模型通常依赖静态或约束不足的参数化方案，这些方案通常未针对稀疏的

观测数据集进行瞬态校准（Goelzer 等人，2018；Aschwanden 等人，2019）。这些局限性导致预报结果存在相当大的不确定性，降低了基于模型的预报的可靠性和实用性。

为应对这些挑战，数据同化（DA）方法通过系统地调整模型状态和参数，使其更接近观测结果，从而被用于提升模型精度和预测能力（Evensen，2009；Carrassi 等人，2018）。在现有的数据同化技术中，集合卡尔曼滤波器（EnKF）得到广泛应用，因为它无需修改模型本身，即可与高维非线性模型集成（Evensen，1994；Gillet-Chaulet，2020；Choi 等（2025））。EnKF 提供了一套严谨的统计框架，用于融合模型预报与观测，从而能够联合估计状态和参数变量，提升模型输出的可靠性和稳健性。

多项研究展示了 EnKF 推进冰盖建模的潜力。Gillet-Chaulet（2020）证明，EnKF 可用于同化理想化海洋终止冰川的瞬态流线（即中等复杂度）模型中的表面高程和速度数据，从而有效约束底部条件并提升接地线迁移预测。Bonan 等人（2013）将 EnKF 集合方法应用于流线浅冰近似冰盖模型的初始化，纳入了基岩和表面地形、表面速度以及高程趋势。他们的研究结果表明，将模型初始条件与当前观测相匹配，对于可靠地预测未来至关重要。Bonan 等人（2014）进一步提出了集合卡尔曼滤波器（ETKF），这是 EnKF 的一种变体，用于联合估计浅冰近似流线模型的基岩地形、冰厚度和底部滑动参数。同样，Hossain 等人（2023）将 ETKF 用于同化冰川表面高程和冰体横向范围观测，方法是更新流线模型中描述海洋终止冰川冰界面的水平集函数。他们的研究展示了数据同化在监测季节性及多年冰川进退周期方面的有效性。通过纳入遥感表面高程剖面，他们更准确地追踪了冰川末端的迁移以及冰川表面特征的变化。近期，Choi 等（2025）将基于 EnKF 的方法与冰盖系统模型（ISSM）（Larour 等人，2012）耦合，构建了一个框架，旨在改进二维冰盖模型中半理想化区域的状态和参数估计；该模型被广泛用于预测（即高复杂度）。他们的研究表明，基于集合的数据同化方法只需经过少数几个同化循环，就能稳健地恢复底部条件和模型状态。此外，他们发现，同化更多观测能够系统地提高这些估计的精度，从而带来更可靠的模型预测。

尽管基于集合卡尔曼滤波的方法具有诸多优势，但在冰盖建模中实际应用集合卡尔曼滤波仍面临若干长期存在的挑战。其中一个最突出的问题是集合欠采样：有限的集合规模相对于模型状态向量的高维特性而言不足。此外，在预报步骤中，为生成集合实现而运行计算密集型模型模拟，会带来高昂的计算成本。并行可扩展性仍是一个难题，尤其是在将数据同化软件包与热力学、基岩调整等缓慢演变的过程耦合时。此外，观测数据往往有限且稀疏，加剧了估计的不确定性（Gillet-Chaulet，2020年；Goelzer 等人，2017；Choi 等人，2023年）。这些因素共同限制了基于集合卡尔曼滤波的方法的广泛适用性及其影响。文献还指出，另一个主要瓶颈是缺少一种

标准化、可扩展的集合卡尔曼滤波软件库，能够与多种冰盖模型及同化工作流集成。现有实现往往与特定模型架构紧密耦合，或缺乏模块化，这给互操作性、可复现性和协作开发带来了挑战。

为解决这些局限，我们推出了冰盖状态与参数估计器（ICESEE），这是一个专为冰盖模型设计的开源、并行、基于 Python 且可扩展的数据同化库。ICESEE 提供自适应、模块化的接口，并配备标准化的应用程序接口（API），能够与多种建模框架无缝耦合。其架构与模型无关，结合了稳健的基于消息传递接口（MPI）的并行性，以及针对状态和参数估计的自适应机制。ICESEE 的核心是采用 Evensen（2003）提出的非线性公式实现无矩阵更新策略，无需显式计算协方差矩阵，并支持高效的并行执行。除了这一主要算法外，ICESEE 还包含四种串行集合卡尔曼滤波变体，可用于基准测试和实验，灵活适用于各种场景。ICESEE 通过多项创新应对文献中指出的关键挑战：支持有界随机型更新，以实现可靠的参数估计；采用自适应膨胀和局地化，以减轻滤波发散；并提供可扩展的并行化策略，适用于计算开销大的高分辨率模型。此外，它还支持观测数据的动态同化以及集合先验的灵活配置，从而有效平衡计算与建模需求。

我们通过一系列冰盖模型验证 ICESEE 框架，包括 ISSM（Larour 等人，2012年）、基于 Firedrake 的 Icepack 模型（Shapero 等人，2021年）、一维 Flowline 模型（Robel，2021年）以及降阶 Lorenz-96 系统（Lorenz，1995年）。本文重点将 ISSM 和 Icepack 模型作为主要案例研究。这两个应用的结果表明，ICESEE 能够在不同建模尺度和计算平台上提供准确的状态和参数估计。

ICESEE 的一个显著特点是注重标准化和互操作性。通过提供一个用于数据同化的通用平台，ICESEE 支持可复现研究，实现跨方法比较，并促进协作开发。其与模型无关的设计以及在 GitHub 上的开源发布（Kyanjo 和 Robel，2025a）使 ICESEE 成为地球系统建模领域理论进步和实际应用的宝贵资源。

## 2 集合卡尔曼滤波器

集合卡尔曼滤波（EnKFs）是一类相关的数据同化算法，旨在高效估计复杂系统不断演变的状态。它们通过模拟一组模型实现，并利用可用的观测数据更新这些实现。ICESEE 纳入了多种集合卡尔曼滤波变体，以应对不同的计算限制和建模需求。这里我们将简要介绍集合卡尔曼滤波的某些数学和计算要素；有关集合卡尔曼滤波及其与其他数据同化方法之间关系的详细讨论，请参阅原始的集合卡尔曼滤波研究（Evensen，1994年、2003年）及后续综述（Carrassi 等人，2018年；Ahmed 等人，2020年）。

集合卡尔曼滤波器通常可分为两大类：随机型和确定型。ICESEE 中实现了这两种变体，它们都遵循包含两个步骤的同化循环：预报和分析（见图 1）。两种类型的预报步骤一致。在此步骤中，模型方程用于推进 $N_e$ 集合成员的模型状态。该过程会得到集合平均预报 $\bar{X}^f$，其定义如下：

$$\bar{X}^f = \frac{1}{N_e}\sum_{i=1}^{N_e} X_i^f, \tag{1}$$

其中， $X_i^f \in \mathbb{R}^{n\times N_e}$ 是每个集合成员 $i$的模型状态向量，其大小为 $n$ 。相应的预报误差协方差矩阵 $P^f \in \mathbb{R}^{n\times n}$ 计算如下：

$$P^f = \frac{1}{N_e-1}\sum_{i=1}^{N_e}(X_i^f - \bar{X}^f)(X_i^f - \bar{X}^f)^T. \tag{2}$$

待同化的数据记为 $y \in \mathbb{R}^m$，这是一个维度为 $m$的观测向量，xxs 在给定的分析时刻 $t^a$可用。这些观测可能包括表面速度、冰厚度或表面高程，具体取决于应用场景。

### 2.1 随机集合卡尔曼滤波（EnKF）

随机型集合滤波器（如集合卡尔曼滤波）在处理大型集合中的非线性问题时，通常比确定型滤波器（第 2.2 节）更有效。这一优势源于在观测向量中加入高斯扰动，可减轻非线性系统动力学生成的集合中高阶非高斯矩的增长；这些矩会降低集合状态估计的准确性（Evensen（2009））。

在 ICESEE 实现的随机集合卡尔曼滤波表述中，会对每个集合成员的观测向量分别施加扰动，以抵消分析步骤低估集合离散度的倾向。ICESEE 采用受 Evensen（1994）、Burgers 等（1998）和 Evensen（2009）启发的串行实现。在 $t^a$，更新后的分析集合为：

$$X_i^a = X_i^f + K(d_i - H(X_i^f)) \quad \forall\, i \in [1, N_e], \tag{3}$$

其中， $H$： $\mathbb{R}^n \to \mathbb{R}^m$ 是观测算子， $d_i = y^a + \Upsilon_i$ 是时间 $t^a$ 的集合虚拟观测，其中$\Upsilon_i$ 表示从均值为零、观测误差协方差为 $R$的多元正态分布中抽取的伪随机观测误差。 $K$ 是卡尔曼增益矩阵，定义如下：

$$K = P^f H^T (H P^f H^T + R_e)^{-1}, \tag{4}$$

其中， $R_e = DD^T$ 是误差协方差矩阵， $D \in \mathbb{R}^{m\times N_e}$ 是集合观测扰动矩阵，其列为 $d_i\ \forall\, i \in [1, N_e]$。

对于大型系统，显式计算 $P^f$ 并对公式（4）中的矩阵求逆会带来高昂的计算开销。ICESEE 采用 Evensen（2003）提出的、更为实用的公式，避免直接计算 $P^f$。在这种

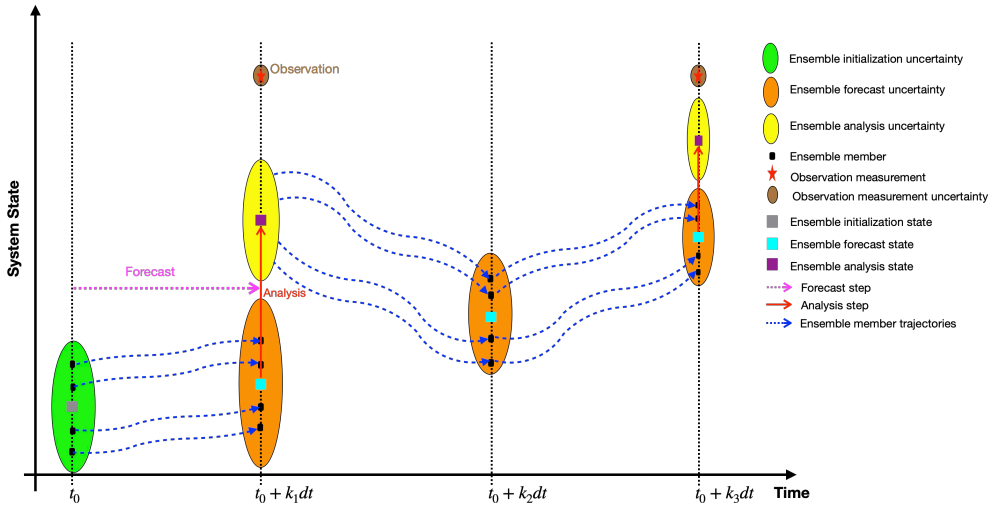


图 1：集合卡尔曼滤波器示意图（改编自 Gillet-Chaulet（2020年）），展示了四个同化循环。该过程从在时间 $t_0$ 初始化的四个集合成员（绿色椭圆内的黑点）开始，随后系统状态（绿色椭圆）经过预报阶段（以洋红色虚线箭头表示）向前推进，生成预测集合状态（橙色椭圆内的青色方框）。收到观测（红色星形）后，通过分析步骤（红色箭头）将状态更新为新状态（黄色椭圆内的紫色方框）。后续的预报阶段和分析阶段重复这一循环，蓝色点状箭头表示各个集合实现的轨迹。这里， $k$ 和 $dt$ 分别表示时间步索引和时间增量。

公式中， $P^f$ 表示为：

$$P^f = \frac{1}{N_e - 1} X'(X')^T, \tag{5}$$

其中 $X' = X^f - \bar{X}^f$。同理， $R_e$ 为：

$$R_e = \frac{1}{N_e - 1} \Upsilon \Upsilon^T. \tag{6}$$

改写公式（4）后，卡尔曼增益为：

$$K = X'(X')^T H^T (HX'(X')^T H^T + \Upsilon \Upsilon^T)^{-1}. \tag{7}$$

假设观测扰动互不相关，项 $HX'(X')^T H^T + \Upsilon \Upsilon^T$可简化为 $(HX' + \Upsilon)(HX' + \Upsilon)^T$。这样一来，矩阵求逆的规模便从 $m^2$ 缩小到 $mN_e$，对于通常满足 $m >> N_e$的情况，计算处理将容易得多。

对 $(HX' + \Upsilon)(HX' + \Upsilon)^T = U\Sigma\Sigma^T U^T$进行奇异值分解（SVD），其中 $U$是正交矩阵，对角矩阵 $\Sigma\Sigma^T$对应于对角矩阵 $\Lambda$ 的上方 $N_e \times N_e$ 块：

$$\text{SVD}(HX'(X')^T H^T + \Upsilon\Upsilon^T) = Z\Lambda Z^T. \tag{8}$$

现在可以将卡尔曼增益更简洁地写为：

$$K^* = X'(HX')^T(U\Lambda^{-1}U^T) \tag{9}$$

其中， $\Lambda^{-1}$ 只有对角线上包含非零元素。将公式 (9) 代入公式 (3) 中的分析步骤，可得以下更新表达式：

$$X_i^a = X_i^f \mathcal{X}^* \tag{10}$$

此处， $\mathcal{X}^* = I + K_i^*(d_i - H(X_i^f))$ 表示一个 $N_e \times N_e$ 矩阵，计算成本低且存储时占用的内存少。这种矩阵形式支持可扩展且便于并行的更新机制，并已完全集成到 ICESEE 框架中。有关此方法的详细推导和进一步说明，请参见 Evensen（2003）。本文后续所述结果仅采用这种实用的滤波器形式。

### 2.2 确定性集合卡尔曼滤波器

确定性集合卡尔曼滤波变体有多种，其区别在于分析步骤的结构，尤其是卡尔曼增益的应用方式。与随机方法不同，确定性集合卡尔曼滤波器不会针对每个集合成员扰动观测。这样便消除了观测扰动引入的额外抽样噪声，从而能更稳定地估计后验协方差并降低计算成本；在大型集合中，这一优势尤为明显（Wang 和 Bishop，2003年；Whitaker 和 Hamill，2002）。ICESEE 实现了本节所述多个变体的串行版本。

#### 2.2.1 集合平方根卡尔曼滤波器（EnSRF）

ICESEE 实现的 EnSRF 变体无需扰动观测，在特定条件下兼具精度和计算效率方面的优势。EnSRF 不依赖随机型更新，而是通过涉及预报误差协方差矩阵平方根的确定型变换进行更新。这种方法可确定如何变换集合，使其与卡尔曼滤波器的分析误差协方差结构保持一致。

在 EnSRF 中，集合均值 (11) 使用传统卡尔曼增益进行更新，而相对于均值的偏差则使用约化卡尔曼增益进行调整。这样可以避免对每个集合成员单独应用完整的卡尔曼增益，从而降低计算成本。通过确定性地执行这些更新，EnSRF 最大限度地减少了其他集合方法中的随机性，使同化过程更加可控且一致。有关详细的理论背景，请参阅 Tippett 等人 (2003) 以及 Whitaker 和 Hamill (2002)。分析集合均值的计算公式为：

$$\bar{X}^a = \bar{X}^f + K_e(y^a - H(\bar{X}^f)), \tag{11}$$

其中，约化卡尔曼增益 $K_e = X'(HX')^T S^{-1}$，且 $S = HX'(HX')^T + R$。随后，使用降秩变换更新每个集合成员：

$$X_i^a = \bar{X}^a + X_i' \left[ I - (HX_i')^T S_i^{-1} HX_i' \right]^{\frac{1}{2}}. \tag{12}$$

### 2.2.2 集合变换卡尔曼滤波器（EnTKF）

集合变换卡尔曼滤波器（EnTKF）是一种平方根滤波器变体，其采用的平方根矩阵比集合平方根滤波器（EnSRF）使用的平方根矩阵具有更高的计算效率。我们实现了 Bishop 等人（2001）最初提出、随后由 Wang 等人（2004）改进的一个版本。与 EnSRF 类似，EnTKF 在集合子空间而非整个状态空间或观测空间中进行分析；随机集合卡尔曼滤波则是在整个状态空间或观测空间中进行分析。与随机集合卡尔曼滤波不同，后者会逐个更新集合成员，而 EnTKF 会同时对整个集合执行一次分析更新，从而降低顺序更新所带来的计算成本。分析方程（3）在集合空间中重新表述，因此可以在这个降维空间中计算卡尔曼增益矩阵（（13）），而不是在维度更高的观测空间或状态空间中计算。方程（11）中的卡尔曼增益矩阵变为

$$K_e = X^f \Omega (H(X'))^T R^{-1}, \tag{13}$$

其中，$\Omega = (I + (H(X'))^T R^{-1} H(X'))^{-1}$ 为集合变换矩阵。因此，该滤波器的分析更新公式为：

$$X^a = \bar{X}^a + \sqrt{(N_e - 1)\Omega}, \tag{14}$$

¯其中， $X^a$ 由公式 (11) 给出，但将 $K_e$ 替换为公式 (13)。

### 2.2.3 确定性集合卡尔曼滤波器 (DEnKF)

我们按照 Sakov 和 Oke (2008) 描述的方法实现确定性集合卡尔曼滤波器 (DEnKF)。DEnKF 融合了集合平方根滤波器 (EnSRF) 的简洁性与随机集合卡尔曼滤波的特性。与 EnSRF 一样，DEnKF 是一种确定型滤波器：尽管它源自随机集合卡尔曼滤波框架，但其分析误差协方差不依赖于观测的具体实现。从概念上讲，DEnKF 可视为 EnSRF 的线性近似，在分析增量较小时尤其有效。它无需引入观测扰动，即可使误差协方差渐近地匹配理论值。在 DEnKF 中，对每个集合成员异常应用卡尔曼分析更新时使用一半的卡尔曼增益 $\frac{K_e}{2}$，因此无需扰动观测，如公式 (15) 所示。每个集合成员的分析更新计算如下：

—

$$X_i^a = \bar{X}^a + X_i' - \frac{1}{2} K_e H(X_i'), \tag{15}$$

¯其中 $X^a$ 和 $K_e$ 在公式 (11) 中定义。

### 2.3 集合初始化与噪声建模

集合初始化是生成多样化初始条件集合的关键步骤，这些初始条件用于表示模型状态的不确定性。在 ICESEE 中，这通过使用随机噪声扰动初始状态向量来实现，噪声按用户定义的集合离散度进行缩放。噪声通过两种方式生成：采用受 Evensen（2003年）启发的伪随机场，或使用 gstools 库（Müller 等，2022年）生成的随机场并结合 Cholesky 分解。由此得到的扰动值构成初始集合矩阵，其中每一列对应一个集合成员，每一行对应一个特定的状态变量或参数。这种结构针对预报和分析阶段的并行处理进行了优化。

按照 Evensen（2003）的公式，状态向量通过增添模型噪声向量进行扩展，使分析步骤能够联合更新观测模型变量与噪声分量之间的协方差。这种方法既能校正噪声均值，也能降低其方差。该策略对于大气和海洋模型等高度动态且混沌的系统十分有效——在这类系统中，状态演变迅速，系统不太可能从先前的集合实现中获益；但对于冰盖系统，这种方法的优势可能较小，因为其主要动力变化发生在更长的时间尺度上（例如，百年尺度）。在这类缓慢演变的系统中，滤波器更有可能从先验实现中提取有用信息；若初始集合离散度设置合理，模型噪声也不太可能发生显著变化。此外，将模型噪声向量增广到状态向量中，会因二者维度相同而产生大量内存和计算开销。在 ICESEE 中，我们采用了一种更具针对性的方法：将每个预报步骤生成的模型噪声写入文件，并在后续预报步骤的噪声生成中重复使用。这种方法避免了不必要的计算开销，并将集合欠采样的风险降至最低；集合欠采样可能由变量幅值差异引起，尤其是模型噪声的幅值通常低于状态变量或参数。此外，我们还修改了伪随机场生成算法，以便在初始化和噪声演变期间更精细地控制集合离散度。具体而言，我们引入了一个具有物理依据的空间去相关尺度，通过参数去相关_长度定义为

$$decorrelation_length = \frac{\sqrt{L_x \times L_y}}{(n_x \times n_y + 1)}, \tag{16}$$

其中，$L_x$ 和 $L_y$ 是区域的空间范围，$n_x$、$n_y$ 是相应的网格分辨率。该公式确保集合变异性在不同空间尺度上得到适当维持，并确保在整个数据同化过程中保留观测敏感性。更多实现细节请参见 Kyanjo 和 Robel（2025a）。

在随机集合卡尔曼滤波中生成创新项 $d_i$ （公式（10）），对于防止集合塌缩和滤波发散至关重要。这些问题源于观测覆盖有限、模型长期趋势以及季节效应，而这些情况在冰盖建模中十分常见。ICESEE 采用两种策略来缓解这些问题，用户可任选其一：

1. 假设集合已经捕捉到由先验实现提供信息的潜在状态空间不确定性。集合扰动（$X_i^f - \bar{X}^f$）按零均值计算，随后投影到观测空间中。

然后使用以下公式计算创新项：

$$d_i = y^a + H(X_i^f - \bar{X}^f). \tag{17}$$

这种方法降低了对观测扰动的依赖，维持集合离散度，并减少观测稀疏区域中的虚假相关。

2. 或者，使用伪随机场生成扰动集合。这些场按观测标准差进行缩放，调整至均值为零，然后投影到观测空间。该方法有助于消除因观测稀疏而可能产生的虚假长程空间相关。

这两种方法都能确保 ICESEE 维持稳健的集合离散度并避免人为相关，从而在较长的模拟期间保持稳定、准确的同化性能。

### 2.4 膨胀与局地化

鉴于当前计算资源的限制，若集合规模远小于模型状态空间维数，则无法用该集合完整表示高维非线性系统的误差协方差矩阵：$N_e \ll n$。预报协方差矩阵的秩亏会引入抽样误差和虚假长程相关，往往导致滤波发散。为缓解这些问题，通常采用膨胀与局地化技术来减少此类误差并改善滤波性能。在包括冰盖建模在内的许多应用中，这些方法通常依赖人工调参，以确定最佳膨胀与局地化参数。这一过程既费时费力，又对模型配置十分敏感（Gillet-Chaulet, 2020年；Choi 等人，2025年）。

在 ICESEE 中，我们为随机集合卡尔曼滤波的串行实现采用了手动局地化和自适应局地化策略。自适应方法基于空间依赖的 Gaspari-Cohn函数，通过分析集合异常，利用集合相关计算随空间变化的局地化长度尺度（Bergemann和Reich，2010年；Evensen等人，2022年）。此外，还结合了加性膨胀或乘性膨胀。尽管采取了这些措施，手动局地化仍然繁琐，而自适应局地化则带来了显著的计算负担。因此，在本研究中，我们主要采用第2.1节介绍的随机集合卡尔曼滤波方法，以及第2.3节所述的噪声建模策略。这些方法无需显式局地化，同时保持了计算效率和稳健性，因为它们隐式利用预报误差协方差矩阵，而不直接计算该矩阵；更多详情请参见 Evensen（2003年）。

### 2.5 参数估计

ICESEE 可以通过扩展模型状态向量，同时纳入物理状态变量（即由模型更新的变量）和模型参数（即不由模型更新的参数），从而实现“联合”估计。该联合向量会在分析循环期间同步更新，使状态和参数都能够得到估计。为实现这一点，需要对参数进行人为扰动，以引入初始不确定性，确保足够的参数离散度（定义为集合参数空间的标准差）。与状态变量不同，这种离散度在预报步骤期间保持不变，仅在分析阶段更新。然而，随着时间推移，联合估计可能导致滤波发散。具体而言，参数离散度逐渐减小，会降低滤波器对观测的敏感性；Ruiz 和 Pulido（2015年）以及 Sueki 等（2022年）也指出了这一问题。

此外，随着时间推移，联合估计可能导致滤波发散。具体而言，参数离散度逐渐减小，会降低滤波器对观测的敏感性；Ruiz 和 Pulido（2015年）以及 Sueki 等（2022年）也指出了这一问题。存在多种缓解策略，例如局地化与膨胀（第 2.4 节）。然而，当状态和参数合并为一个向量时，同时对状态变量和参数应用局地化并非易事。正如 Aksoy 等（2006年）所强调的，仅在状态空间中直接应用局地化可能导致局地化矩阵不是半正定的，从而可能产生不符合物理规律的结果。不过，由于存在长程相关性，参数空间中仍然需要进行局地化。

为解决这一问题，ICESEE采用了Ruckstuhl和Janji´c（2018年）提出的参数动力学模型。这种方法尤其适用于冰盖建模，因为它有助于在冰盖特有的漫长响应时间内维持参数变异性。与传统的基于高斯分布或基于膨胀的策略不同，该方法使用贝塔分布描述有界的参数演化。这种方法的一项主要挑战是需要手动指定贝塔分布的上下界。ICESEE通过利用可用的观测数据自动估计这些界限来克服这一难题。对于未观测到的参数，ICESEE会根据当前和先前的集合矩阵自适应地获取上下界。这种策略能够动态维持参数离散度，确保更新符合物理规律，同时不损害滤波器稳定性。ICESEE还采用了一种互补方法，有意使参数子空间的膨胀程度大于状态子空间。该方法与第2.1节所述的随机集合卡尔曼滤波表述以及第2.3节所述的噪声建模策略相结合。噪声建模方法会在初始化时为参数集合设置足够大的离散度，并使其在整个同化循环中得以维持。这有助于保留参数对观测的敏感性，防止集合塌缩，从而进一步增强滤波器鲁棒性。本文的实验均采用这一互补策略。

除了在集合卡尔曼滤波框架内联合估计状态和参数外，ICESEE 还可与外部反演建模工具（例如在 ISSM 或相关优化框架中实现的基于伴随的方法）对接，以推断观测较弱或未直接纳入同化状态向量的参数。在这类情况下，ICESEE 将同化后的状态变量作为基于物理的反演程序的输入，从而实现同化—反演混合工作流：先后应用基于集合的数据同化和确定性反演。在本研究中，ICESEE 与 ISSM 耦合，以展示这一混合方法：利用同化后的厚度和速度场，通过 ISSM 的反演框架推断基底摩擦（Morlighem 等，2013年）。这种组合体现了 ICESEE 将基于集合的数据同化与成熟的反演建模工具相结合的灵活性，有助于提高冰盖模拟中的参数估计效果。

## 3 ICESEE 库的实现

ICESEE库采用模块化且可扩展的架构开发，旨在支持冰盖模型高效且可复现的数据同化，并有望在未来用于其他应用。其设计强调互操作性、灵活性和可维护性，使研究人员能够在不同的建模环境中调整和扩展该框架。该代码库

完全使用Python实现，选择Python是因为它语法易用、科学计算生态系统丰富，且能与其他编程语言无缝互操作。Python能够与使用C++、Fortran或MATLAB编写的软件交互，因此ICESEE可以集成传统和现代高性能冰盖建模系统。

该软件包采用清晰的目录结构，每个主要组件都配有专门的*README*文件，介绍其功能、依赖项和使用示例。这些文档有助于新用户快速上手，并支持可复现工作流。详细的安装说明、使用指南，以及展示如何与各种模型耦合的示例，均见 ICESEE GitHub Wiki（Kyanjo 和 Robel，2025b）；该 Wiki 是面向用户和开发者资源的中心。

### 3.1 软件包布局与结构

ICESEE 采用系统化的目录结构，旨在支持模块化开发和便捷集成。图 2 展示了 ICESEE 代码库实际布局的一个示例。该框架划分为若干独立组件，包括模型接口、数据同化函数、并行化工具、通用函数和测试基础设施。这种模块化设计便于集成新模型，支持串行和并行的集合卡尔曼滤波变体，并使用户无需修改核心同化逻辑即可实现复杂工作流。图 2 中的双向箭头表示组件之间的信息双向流动。值得注意的是，ICESEE 核心例程与应用程序目录完全分离，因此可以为新应用添加耦合功能，而不影响核心功能。

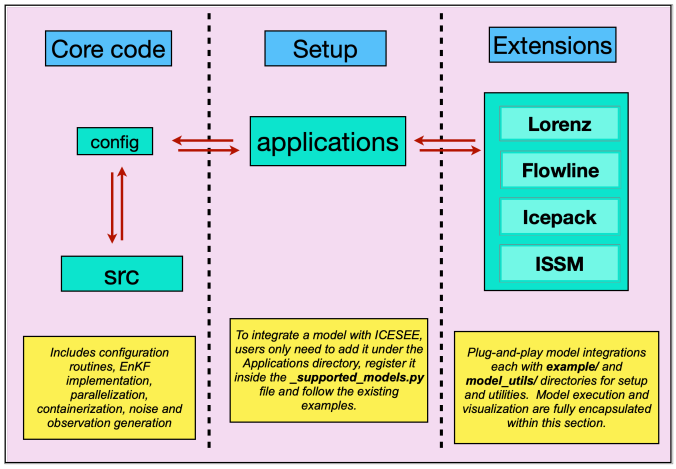


图 2：ICESEE 代码库的高级结构，展示了用于数据同化、模型接口、实用工具和并行化的模块化组件。

ICESEE 也遵循一种结构化工作流，如图 3 所示。工作流中的每个步骤都按顺序执行，但明确标注为并行的流程除外，这些流程会并发运行。在将新应用集成到 ICESEE 时，

用户主要与工作流图（图 3）中以青色和灰色突出显示的组件交互。以绿色、橙色和洋红色显示的 ICESEE 核心例程与具体模型无关，可在不同应用中重复使用。这种模块化设计使用户能够轻松地根据自身建模需求调整 ICESEE，而无需修改底层的数据同化逻辑。

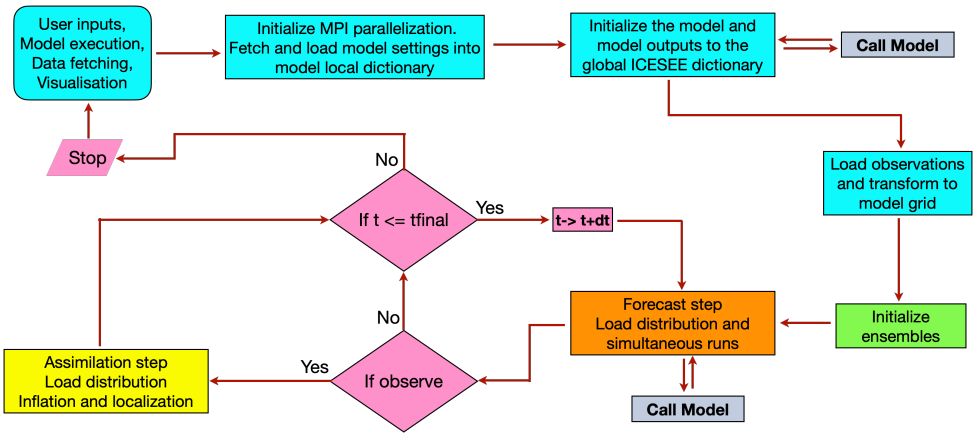


图3：ICESEE 工作流概览，展示了从模型初始化、观测加载与转换至模型网格、集合初始化、模型预报到同化的操作顺序。绿色、橙色、洋红色和黄色部分代表 ICESEE 核心例程，青色和灰色部分则对应用户可配置的特定模型扩展。

### 3.2 主要组件与例程

本节概述构成 ICESEE 的关键组件和例程，并将其划分为几个主要功能类别：数据同化、并行执行、实用工具、配置、应用程序包装器和执行管线。对于每个类别，我们将概述其核心例程、所在目录，以及它们与 ICESEE 其他部分和外部模型之间的交互。

#### 3.2.1 数据同化核心

数据同化核心涵盖了 ICESEE 中实现的集合滤波技术相关的所有例程和模块。这些例程支持串行和并行集合卡尔曼滤波（EnKF）操作，以及高效数据同化工作流所需的辅助功能。主要组件位于 *src/* 目录中。*EnKF* 子目录包含以 Python 编写的串行 EnKF 实现，以及基于 Cython 的随机集合卡尔曼滤波。这些模块负责基本的同化操作，包括预报、分析、集合扰动、卡尔曼增益计算、为串行 EnKF 版本应用观测算子，以及实现自适应局地化和膨胀技术。

并行化 子目录包含实现并行 EnKF 框架的模块，包括并行输入/输出操作、集合初始化、预报和分析步骤，以及生成合成观测的例程。公式（10）中所述的并行随机集合卡尔曼滤波，在 *_mpi_分析_functions.py* 模块和 *EnKF_并行_io.py*类中均有实现，分别对应第 3.4 节所述的两种并行化策略。运行_模型_数据同化 目录包含主驱动脚本 运行_模型_*da.py*，该脚本通过函数 *icesee*_模型_数据_同化协调整个数据同化循环

此函数充当中央控制器，协调工作流中的所有关键阶段：加载分布、集合初始化，以及执行预报和分析步骤。集成到 ICESEE 中的每个模型都会调用此函数，以执行其完整的数据同化循环。

每个受支持的模型（如 ISSM、Icepack、Lorenz 或流线模型）都在*applications/*文件夹中拥有专属子目录。这些子目录包含用于初始化、配置、预报、生成真实状态和扰动状态（用于孪生实验）以及后处理的特定模型脚本。直接与 ICESEE 核心交互的脚本以 ICESEE 全局字典（*kwargs*）作为输入，并遵循标准化的命名约定，以确保所有模型之间的兼容性和一致性。对于每个集成到 ICESEE 中的模型，名为 *_modelname_enkf.py* 的通用数据同化脚本位于该模型的实用工具子目录中（例如 *modelname_utils/*）。此通用脚本定义模型的 EnKF 程序，并支持与该模型相关的所有示例配置。不过，ICESEE 也允许在特定模型的*examples/* 目录中提供示例专用版本的 *_modelname_enkf.py* 。如果存在这些专用脚本，它们会自动取代通用版本，从而在保持与 ICESEE 框架兼容的同时，支持定制化的同化设置。这种设计既支持新实验的快速原型开发，也便于集成新模型。

The_*modelname_enkf.py* 脚本通常定义以下核心函数：

– *initialize_ensemble(**kwargs)*：根据模型的真实状态和指定的扰动参数构造集合矩阵，为同化引入初始不确定性。– *generate_true_state(**kwargs)*：生成模型的真实状态，用作在完美模型实验中评估同化性能的参考。此函数在同化循环开始之前调用。– *generate_nurged_state(**kwargs)*：为完美模型实验创建真实状态的扰动（或“错误”）版本，代表集合卡尔曼滤波试图通过同化进行修正的初始猜测。– *forecast_step_single(**kwargs)*：逐个集合成员向前推进模型状态，生成预报集合，并为后续分析步骤做好准备。

### 3.2.2 并行执行

本小节介绍 ICESEE 中支持并行执行的例程；软件包整体并行化策略的详细说明将在第 3.4 节中给出。如图 3 所示，MPI 并行化在同化循环开始时初始化，在高效数据处理以及并发管理

多个处理器上的集合成员方面发挥着核心作用。这些例程负责分配计算工作负载、处理处理器间通信，并确保集合成员在预报和分析阶段均能并发执行。ICESEE 的大部分并行工作流由 *ParallelManager()* 类协调，该类在 *icesee_mpi_parallel_manager*模块中实现，位于 *src/parallelization/parallel_mpi/* 目录下。此类负责配置 MPI 环境、将集合成员分配给可用进程，并管理整个同化过程中的通信。它提供广播数据、收集结果以及在所有进程间同步集合状态的方法。用户通常只需调用 *icesee_mpi_init()* 函数，即可初始化 MPI 环境，并启用 ICESEE 通信器，以用于基于 MPI 的工作流（参见第 3.4 节）。此函数会建立通信基础设施，并为分布式计算做好集合准备。其他 MPI 例程会在*icesee_model_data_assimilation*中自动调用，从而向 ICESEE 核心例程提供子通信器、进程、大小及其他 MPI 属性。

*ICESEE_MPI_并行_管理器*类还包含用于管理集合矩阵的实用工具，例如将集合状态广播到所有进程、收集分析结果，以及并行计算集合均值。该类在设计时充分考虑了灵活性和可扩展性，允许用户根据需要调整或扩展并行执行策略，并集成外部的 MPI 兼容库。

在数据管理方面，ICESEE 通过*h5py*接口（h5py 开发者，2022年）使用 HDF5 文件格式，所有数据都存储在 *_modelrun_datasets/* 目录中。HDF5 标准尤其适用于高性能计算，因为它支持并行输入/输出，允许多个进程同时读写，因此非常适合大规模集合模拟。ICESEE 的并行输入/输出例程可高效管理 MPI 和非 MPI 环境中的集合状态、观测及其他关键数据结构的读写。这些例程主要在 *_并行_输_出.py*模块中实现，部分则实现在 集合卡尔曼滤波_并行_输入输出.*py*类中，该类位于 *parallelization/*目录下。两者共同负责 HDF5 数据结构的分布式访问和更新，使集合成员能够跨处理器并发读写其输出，同时确保数据一致性。输入/输出例程经过优化，可最大限度地减少通信开销和内存使用量，并提高数据吞吐量，以满足大规模、高计算强度冰盖建模应用对可扩展性和效率的要求。

### 3.2.3 实用工具

ICESEE 在 *src/* 文件夹中设有专门的*utils/* 目录，其中提供了支持框架多个组件的全面通用函数集合。该目录包含两个主要模块：*tools.py*和*utils.py*。*tools.py* 模块包含用于自动配置 ICESEE Python 环境、并行读写用于可视化和后处理的 HDF5 文件，以及提供计算性能跟踪、进度条显示和运行时统计信息汇总等实用工具的函数。它还包含自适应索引例程*icesee_icesee_get_index(**kwargs)*，用于为 MPI 和非 MPI 配置下存储在状态向量中的状态和参数生成索引映射和数组。*utils.py* 模块定义了*UtilsFunctions* 类，提供了广泛的可复用方法，包括为完美模型实验在预定义观测时刻生成合成观测、计算观测算子（$H$）及其雅可比矩阵，以及定义观测函数。这些实用工具

函数的设计注重模块化和可复用性，便于用户在 ICESEE 框架内的不同模型和应用中轻松集成。集成到 ICESEE 中的每个模型还可以在其模型目录下的专用子目录（*modelname_utils/*）中定义自己的专用实用工具。这些特定于模型的实用工具可以扩展或覆盖核心*utils/* 目录中提供的通用函数，从而实现满足各模型独特需求的定制功能，同时保持与 ICESEE 核心例程的兼容性。

### 3.2.4 配置

ICESEE 主仓库中的 *config/*目录包含两个主要配置模块：*config_loader.py* 和*_utility_imports.py*。*config_loader.py* 模块负责读取并解析各扩展模型目录中的 *params.yaml* 文件。此 YAML 文件定义模型专属参数、与集合卡尔曼滤波相关的参数以及运行时参数，提供了一个集中且可编辑的配置接口，使用户无需修改源代码即可调整设置。文件按不同部分组织，例如模型配置、同化选项和运行时控制，便于灵活尝试不同的同化策略。

*_utility_imports.py* 模块整合了 ICESEE 中常用的组件。它通过*config_loader.py*加载已解析的参数，导入常用的函数、字典和类，应用通用配置设置，并获取命令行参数。这种结构确保整个框架能够一致地访问共享参数和实用工具。所有特定模型脚本，包括*_modelname_enkf.py*、*_modelname_model.py*、*run_da_modelname.py*和*_modelname_utils.py*，都会导入此模块，以便在数据同化过程中访问所需的共享字典、实用函数和配置参数。

为了使 ICESEE 能够与受支持的模型交互，框架使用了一个集中式注册脚本 *受支持_models.py*，该脚本位于*applications/*目录中。该脚本维护所有已集成模型的列表、目录路径及相关接口函数。通过强制采用统一的结构和命名约定，ICESEE 确保在同化工作流期间一致且动态地加载模型专用例程。作为主要配置中心，该文件允许用户指定在某次运行中包含哪些模型。按照既定的注册格式即可无缝纳入新模型，从而增强可扩展性并简化集成。这种集中式方法简化了模型管理，增强了互操作性，并确保与 ICESEE 核心例程兼容。

### 3.2.5 应用程序包装器

由于传统冰盖模型采用多种编程语言、框架和架构开发，ICESEE 使用专门的应用程序包装器，以便与这些模型有效交互。这些包装器对于自动执行模型、管理数据流，以及在异构建模环境中实现同化过程的无缝集成至关重要。一个典型示例是将 ISSM 与 ICESEE 耦合，这需要采用混合工作流，在 Python 和 MATLAB 之间进行通信，因为 ISSM 的前处理与后处理程序完全使用 MATLAB 实现。相比之下，Icepack 等模型完全使用 Python 编写，其计算核心采用 Fortran，并提供 Python 接口，因此不需要如此复杂的工作流。

为便于与 ISSM 交互，ICESEE 采用 MATLAB 服务器（参见第 4.2.1 节）方案：运行一个持久的 MATLAB 进程，用于侦听来自 Python 的命令，并管理多个并发的 ISSM 运行任务。MATLAB 服务器的实现位于*issm_model/issm_utils/* 子目录中，该子目录位于*applications/* 目录下。此模块包含用于启动和终止服务器、发送命令代码、处理错误（例如，在失败时终止 MATLAB 进程）以及支持 Python–MATLAB 通信工作流的其他实用工具的脚本。该服务器是一种通用解决方案，可适配其他需要类似接口策略的模型。

由于 ICESEE 使用 Python 编写，为外部模型实现封装器通常十分简单。现有集成示例表明，ICESEE 可以轻松扩展，以支持多种模型工作流。每个模型都在 *applications/* 目录下设有专用目录（例如 *modelname_utils/*），其中包含将该模型与 ICESEE 对接所需的脚本。此外，ICESEE 还提供了若干通用封装工具，位于 *src/utils/* 目录下的 *tools.py* 和 *utils.py* 模块中。这些工具包括将模型状态映射到 ICESEE 主矩阵、获取模型数据，以及协调模型与同化框架之间数据交换的函数。这些工具旨在供多个模型重复使用，可提升一致性，并显著减少将新应用集成到 ICESEE 所需的工作量。

### 3.2.6 执行管线

本小节概述 ICESEE 的执行管线，该管线负责协调完整的数据同化工作流，包括模型设置、集合初始化、预报和分析，如图 3 所示。该图还展示了特定文件和目录如何用于实现这一工作流。图 3 中以青色阴影标示的组件，对应于获取用户输入和配置设置、初始化模型、加载 ICESEE 全局字典，以及加载观测并将其转换到模型网格等任务；这些组件由每个模型示例目录中的脚本*run_da_modelname.py*实现（例如 *issm_model/examples/model-example*）。该脚本是执行给定模型同化过程的主要入口，负责处理从设置到分析的所有关键阶段。它导入 ICESEE 核心例程（例如 *icesee_model_data_assimilation*），以及特定于模型的函数，使用户只需进行极少的手动配置即可启动工作流。执行时，*run_da_modelname.py*会从 *params.yaml*加载模型参数和运行时设置，配置模型并初始化同化环境

，

并调用*ICESEE_模型_数据_同化函数*。该函数负责协调整个同化过程，包括集合初始化、模型传播和分析更新。这些操作通过全局 ICESEE 字典（*kwargs*）进行交互；该字典作为所有参数和状态变量的集中容器，确保整个工作流中的访问一致性。生成的集合矩阵存储在 ICESEE 的主要 HDF5 数据结构中，这些数据结构作为所有同化相关信息的中央存储库。每个预报步骤都会使用传播后的集合更新这些数据结构，随后在分析阶段利用可用观测进行同化并计算集合校正。执行管线定义了操作的逻辑顺序，而这些计算如何在处理器之间进行分配和并行化，则将在第 3.4 节中讨论。

### 3.3 与 ICESEE 耦合

ICESEE 被设计为一个灵活且可扩展的数据同化框架，能够与广泛的冰盖模型和外部应用程序进行交互。它同时支持内存通信和并行文件式输入/输出，用户可以根据目标模型的架构与计算特性以及可用计算资源，选择最合适的方法。集成通常通过基于 Python 的接口与模型专用 API 相结合来实现，从而支持 ICESEE 与底层模拟代码之间高效的数据交换和同步。

#### 3.3.1 内存通信

对于完全使用 Python 实现的模型（如 Icepack），内存通信尤为有效。这种方法无需承担文件输入/输出的开销，即可直接访问模型状态和参数，从而加快数据交换速度，缩短同化循环。通常可通过共享内存结构或直接函数调用来实现，从而实时更新集合和观测数据。内存通信尤其适用于规模较小至中等的集合和模拟（即元素数量），以及内存需求适中的模型；在同化过程中，这种方法能够最大限度地降低延迟并提高计算效率。

#### 3.3.2 基于并行文件的输入/输出

对于使用 Python 以外语言编写的模型，或内存占用较大且计算需求较高的目标模拟，基于文件的通信策略可提供更具可扩展性的解决方案。在这种方法中，模型状态、观测及相关数据以 HDF5 等结构化格式写入磁盘并从中读取，而 HDF5 支持并行输入/输出操作。基于文件的通信利用分布式文件系统和多个处理器间的并发访问，高效管理大型集合和复杂模型。在高性能计算（HPC）环境中，这种方法尤具优势，因为内存限制可能使纯内存解决方案无法实施。一个典型示例是冰盖系统模型（ISSM），它使用 C++ 编写，并采用基于 MATLAB 的前处理与后处理。在这种情况下，ICESEE 在 Python 和 MATLAB 之间采用基于文件的数据交换，确保异构编程环境之间可靠集成。

#### 3.3.3 状态索引映射

无论采用何种通信策略，ICESEE 都会使用动态例程 *icesee_get_index(**kwargs)*，将模型状态索引（无论是存储在内存中还是从 HDF5 文件中读取）映射到 ICESEE 全局状态矩阵中。此函数确保在整个同化循环中一致地访问和处理模型数据。它会构建索引映射和数组，以便高效检索和更新状态变量，从而实现 ICESEE 核心例程与外部模型接口之间的无缝交互。每次数据交换阶段都会调用 *icesee_get_index* 例程，以保持模型专属状态变量与其在 ICESEE 全局向量中对应条目之间的映射一致。此

映射对于基于内存和基于文件的工作流都至关重要，可确保在每个同化步骤中准确应用集合预报、更新和分析。

### 3.4 软件包并行化

ICESEE 为集合初始化、预报和分析阶段实现了多种并行化策略，包括基于 MPI 的并行性、Python 的 *multiprocessing* 模块以及 Dask。其中，MPI 是大规模随机集合卡尔曼滤波模拟（10）的主要方法，具有出色的可扩展性和计算效率。Python*multiprocessing* 和 Dask 与基于 MPI 的工作流结构相似，但针对规模较小的问题或计算环境受限的情况进行了优化。本节重点介绍 MPI 实现，它最为稳健、可扩展，并使用了高性能计算系统中广泛可用的 MPI 例程。

第2.1节所述的集合卡尔曼滤波算法尤其适用于冰盖模型中常见的大规模集合和高维状态空间。然而，管理此类集合需要大量计算资源，特别是在处理预报和观测更新时。集合卡尔曼滤波方法的一大优势是，预报步骤在大型问题中消耗了大部分计算资源（如第5节所示），但各集合成员的预报步骤彼此独立。这使得完全并行 执行成为可能，并具有出色的可扩展性。相比之下，分析步骤在计算卡尔曼增益和更新状态集合时会引入全局依赖关系，这可能导致大型系统中出现通信开销。

为应对这些挑战，ICESEE 提供了两种由 MPI 驱动的并行化模式：部分并行化 和完全并行化。在这两种模式下，初始化和预报阶段都会使用 MPI 子通信器，根据可用进程总数（*MPI* 进程）和集合规模（$N_e$），将集合成员分组为多个处理轮次：

– **情况 1：** $N_e \geq$MPI 进程 – 集合被划分为多个轮次，每轮并行执行*MPI* 进程 个集合成员。

– **情况 2：** $N_e <$MPI 进程 – 全局通信器被拆分为 $N_e$ 个子通信器。每个子通信器使用MPI 进程 $N_e$ 个进程执行一个集合成员。

虽然这两种方法都依靠 MPI 子通信器来实现可扩展的预报和初始化执行，但它们在数据交换、存储和进程间分发方式上有所不同。部分并行化 模式将数据管理集中在根进程上，简化了协调工作，但会增加内存需求；而完全并行化 模式则将计算和输入/输出均匀分布到所有进程，从而消除根进程级别的瓶颈。实际上，用户会根据模型规模、问题大小和可用资源，在两种模式之间进行选择。部分并行化模式最适合较小的集合和问题规模，或用于调试的运行；在这些情况下，集中式数据管理可以简化协调并降低输入/输出的复杂性。相比之下，对于大规模实验，完全并行化模式更为合适，因为将计算和输入/输出分布到所有进程能够最大限度地提高可扩展性、尽量减少内存瓶颈，并有效利用高性能并行文件系统。接下来的小节将详细介绍每种策略。

### 3.4.1 部分并行化

在部分并行化策略中，MPI 子通信器负责管理初始化和预报轮次，而全局根进程则协调文件输入/输出和分析阶段。当有 $N_e \geq$ 个 MPI 进程时，初始化和预报阶段的集合成员会在并行组中并行执行。每个子通信器的根进程都会收集其各进程的本地结果，将其汇总到列表中，然后转发给全局根进程。全局根进程随后合并所有子通信器的输出，以重建完整的集合数组，计算集合均值，并将两个输出写入磁盘。

在分析步骤中，由于集合输出已在根进程上可用，全局根进程会计算 $\boldsymbol{\mathcal{X}}^*$，并将其广播给所有进程。根进程还会将集合数据块分发给各个进程，以便并行计算局部分析更新。随后，将更新后的集合片段收集起来、合并，并写回磁盘，供后续预报步骤使用。

当有 $N_e <$ 个 MPI 进程时，全局通信器会划分为 $N_e$ 个子组，每个子组负责一个集合成员。在每个子通信器中，根进程会收集局部状态更新；全局根进程再将这些更新汇总、打包并写入磁盘，同时在初始化或预报阶段后计算集合均值。

在这两种配置中，分析阶段遵循相同的工作流：每次预报后，所有进程都会同步；根进程构建 $\boldsymbol{\mathcal{X}}^*$；每个进程计算各自的局部分析更新。随后，根进程执行集合膨胀，按照 Evensen（2003）重新计算均值，并将集合场和均值场写回磁盘，以供下一次迭代使用。

此方法确保集合均值、扰动和更新均以并行方式计算和存储。预报和分析数据通过 MPI-IO 高效写入 HDF5 文件。ICESEE 维护一个维度为（$n_d$，$N_e$，$N_t$）的全局数据集，其中 $n_d$ 是状态向量维度，$N_t$ 是时间步数。尽管在大型集合数据同化实验中，此工作流可使预报阶段和分析阶段实现强扩展，但当完整集合数组必须驻留在根进程上时，可能会受到内存限制，尤其是在模拟维度极高的情况下。

### 3.4.2 完全并行化

ICESEE 中的完全并行化模式通过专用 Python 类实现，该类封装了大规模、支持 MPI 的集合卡尔曼滤波操作所需的所有实用工具，包括直接并行读写 HDF5 数据集超切片、并行计算集合均值、生成合成观测、构建观测算子，以及并行执行分析更新步骤的函数。

在初始化阶段，观测算子 **H** 和合成观测由根进程计算一次，并存储为所有进程均可访问的 Zarr 文件（Deakin，2025）。维度为（$n_d$，$N_e$）的 HDF5 集合数据集以批次方式集体创建，并使用用户定义的分块方案来优化输入输出性能。随后，每个 MPI 进程并发打开这些文件，并保留对分配给自己的数据集句柄的引用，以供后续预报和分析操作使用。

在预报阶段，每个进程直接从 HDF5 超矩形中读取对应的状态切片，执行预报和噪声扰动，并行地将结果写回同一数据集区域。仅在观测时刻计算集合均值，并使用集体 MPI 归约操作来尽量减少同步开销。

在分析阶段，每个进程加载观测算子**H** 的本地块以及对应的合成观测子集，计算本地乘积（$HX$ 和 $Hy$），并执行 MPI *Allreduce* 操作，以汇集所有进程上的全局向量。随后，利用得到的大小为（$m, N_e$）的全局矩阵，在每个进程上并发构造$\boldsymbol{\mathcal{X}}^*$。每个进程从文件中读取集合的指定部分，通过与$\boldsymbol{\mathcal{X}}^*$进行矩阵-向量乘法来应用分析更新，执行集合膨胀，并将更新后的状态写回磁盘。

这种方法可确保充分利用可用的 MPI 资源，同时最大限度地减少通信开销、冗余输入/输出和内存消耗。通过将所有主要操作并行化（从输入/输出到集合更新），ICESEE 在集合规模较大、模型状态维度较高时实现了近线性可扩展性（如第 5 节进一步讨论）。因此，ICESEE 特别适用于冰盖建模和气候建模应用中的高级数据同化实验。

### 3.5 软件包容器化

冰盖建模中反复面临的一项挑战，是如何在高性能计算（HPC）平台上部署模型。此类平台的计算资源十分充足，但软件环境往往差异很大。容器化通过将所有必需的依赖项、库和配置封装到可移植且可复现的环境中，提供了一种可靠的解决方案。这一功能在冰盖建模中尤为重要，因为不同模型依赖特定的编译器版本、科学计算库和辅助工具，而这些资源未必能在各个系统中统一获得。

ICESEE 支持对适合此类封装的扩展模型进行容器化，其中最典型的是 Icepack 和 ISSM。这些模型十分复杂，依赖紧密耦合的软件栈，因此难以在各种计算环境中保持一致的复现。通过容器化，ICESEE 确保这些模型能够可靠且可复现地运行，不受底层硬件或操作系统的影响。针对 Icepack 和 ISSM，ICESEE 提供了可直接使用的容器脚本，使用户能够在受控环境中运行模型。每个容器都打包了所有必需的依赖项，从而简化安装、缓解兼容性问题并提高可复现性。这种方法还简化了模型在各种平台上的部署，涵盖个人工作站到多节点高性能计算集群。

目前，Icepack 容器支持多节点执行，可在分布式系统中进行可扩展的并行模拟。相比之下，ISSM 容器目前配置为单节点运行，但可根据需要扩展，以支持更复杂的并行配置。两种容器的配置文件均位于各模型源代码树中的 *modelname_utils/containers/*目录下。我们提供了全面的构建和执行说明，帮助用户根据具体的建模需求和系统要求构建、运行和自定义容器。

## 4 支持的应用示例结果

ICESEE 目前支持四种模型应用：Flowline、Lorenz96、Icepack 和 ISSM。本文重点展示 ICESEE 在 Icepack 和 ISSM 中的应用能力，因为它们体现了两种不同的耦合策略，并凸显了该框架与多种建模架构集成的灵活性。
Flowline 和 Lorenz96 应用分别是

一个简单的一维流线海洋终止冰川模型（Robel，2021年），以及对经典 Lorenz 模型的实现，该模型描述了一个简单的混沌系统（Lorenz，1995年）。这些应用主要用于测试。

#### 4.1 Icepack

Icepack 是一款基于 Python 的冰盖模型，提供了灵活且可扩展的框架，用于模拟冰动力学。它基于 Firedrake（Ham 等，2023）构建。Firedrake 是一个高性能有限元库，可用于求解偏微分方程（PDE）。借助 Firedrake 的能力，Icepack 能够求解控制冰流的斯托克斯方程的近似解，因此非常适合大规模冰盖模拟（Shapero 等，2021）。

ICESEE 将集合卡尔曼滤波器（EnKF）与 Icepack（v1.0.1）集成，用于估算关键冰盖状态变量，包括冰流速、冰厚度和表面质量平衡（SMB）。本演示采用 Icepack 教程中提供的合成冰流示例，模拟同时包含接地部分和漂浮部分的冰流演化。为开展本研究，我们对示例的几何配置、网格分辨率和总运行时间进行了调整，创建了一个定制测试案例，以展现 ICESEE 和 Icepack 的核心功能。该测试案例旨在演示如何将数据同化技术与 Icepack 的模拟引擎相结合。它展示了通过将合成观测同化到符合物理规律的模型中，ICESEE 如何提升预测性能。此设置展示了完整的数据同化循环，以及 ICESEE 如何在现代冰动力学建模环境中实现实时状态估计。

##### 4.1.1 Icepack：问题设置与模型初始化

我们采用一个狭长的峡湾状区域，长 $L_x = 5$ 千米、宽 $L_y = 1.2$ 千米，从流入边界延伸至冰前缘。该区域在 $x$ 方向上离散为 $n_x = 12$ 个单元，在 $y$ 方向上离散为 $n_y = 8$ 个单元。总模拟时间为 100 年。此设置旨在表示一个“较小”的问题：每个变量仅包含 96 个元素和 425 个自由度，以此展示 ICESEE 的一些基本功能。所有其他冰盖配置参数均与原始 Icepack 教程设置保持一致。

对于数据同化，我们采用公式（10）中所述的并行随机集合卡尔曼滤波器（EnKF），并结合第 3.4.1 节概述的部分并行化策略。状态和参数向量定义为 $\mathbf{X} = [h, u, v, \text{表面质量平衡}]$，其中 $h$ 表示冰厚度，$u$ 和 $v$ 分别表示 $x$ 和 $y$ 方向上的水平速度分量，表面质量平衡 则对应表面质量平衡。通过向真实状态添加高斯扰动生成合成观测，其标准差分别为 $\sigma_h = 5$ 米、$\sigma_u = 0.5$ 米/秒、$\sigma_v = 0.1$ 米/秒和 $\sigma_{smb} = 0.1$ 米/年。在模拟期间选定的时间步长对这些观测进行同化：共采用十个观测时刻，从第 4 年开始，此后每隔 8 年一次，直至第 76 年。本次实验的集合规模固定为 $N = 40$。

初始集合状态使用伪随机场进行扰动，这些伪随机场按照第2.3节所述的方法生成，标准差为：$\sigma_h = 0.8$ m、$\sigma_u = 0.4$ m/s、$\sigma_v = 0.01$ m/s 和 $\sigma_{smb} = 0.08$ m/yr。除表面质量平衡外，所有状态变量均使用 96 m 的去相关长度尺度；表面质量平衡使用较短的 75 m 长度尺度，以反映更大的空间变异性。

通过修改表面质量平衡生成错误的（初始猜测）状态：将流入端的积累速率从 1.7 m/yr 降至 0.17 m/yr，并将下游积累速率的空间变化从 $-2.7$ 降至 $-0.27$ m/yr。

此外，通过减去一个振幅为 350 m 的平滑线性凸起来扰动冰厚度，该凸起作用于流入边界附近区域前 25% 的范围内。此凸起使用线性间隔数组定义，以确保状态相对于真实状态逐渐偏离。这些扰动旨在模拟一种常见情形，即卫星观测前冰川状态和气候强迫都不太明确。为补偿集合中的欠离散，仅对表面质量平衡更新应用 1.12 的膨胀因子。

Icepack 模型使用 _初始化_模型 函数进行初始化，该函数位于 *_icepack_model.py* 脚本中，脚本所在目录为*icepack_model/examples/*。在此函数中，ICESEE 通过 Firedrake 网格对象将 MPI 通信器传递给 Icepack，从而建立 MPI 环境。这使 Icepack 能够在预报步骤中并行运行，从而高效计算集合成员。初始化例程还会从*params.yaml*文件读取配置参数，该文件指定模拟所需的模型设置、同化选项和运行时控制。

### 4.1.2 Icepack：实验模拟

目前，只有当集合成员数超过处理器数时，ICESEE 才支持 Icepack 并行执行。当 $N_{\text{ens}} <$ 个 MPI 进程时，区域会在子通信器中的独立 MPI 进程之间不均匀分配。这可能导致通信效率低下，或需要在模拟过程中进行额外的数据汇集，而 ICESEE 目前尚不支持这些操作。这一限制源于 Icepack 依赖 Firedrake，而 Firedrake 基于 PETSc（Balay 等，2024）构建，以支持并行计算。Icepack 通常以串行模式运行；但在 ICESEE 框架中，通过为每个进程分配一个集合成员来实现并行性，使每个进程都能独立存储并模拟完整的计算域。这样便无需在模型执行过程中进行进程间通信，并能实现可扩展的集合预报。因此，即使在单节点上运行，ICESEE 也能有效地在中大型问题上使用 Icepack 进行数据同化，而不会产生显著的通信开销。

冰厚度的空间演变和速度差异（分别见图 4 中的面板（$b_1$–$e_1$）和（$b_2$–$e_2$））表明，仅通过四个同化循环（见图 5），ICESEE 就能够在 25 年内恢复真实模拟。厚度差异的幅度显著减小，从约 350 m 降至接近零。同样，速度差也从约 85 myr$^{-1}$ 降至接近零。这些结果凸显了 ICESEE-Icepack 耦合在校正初始偏差状态方面的稳健性。

图 5 展示了 ICESEE 的联合估计能力。在参考点，冰厚度的集合均值在约 20 年后趋近真实状态，而速度场在最初 10 年内便与真实值一致。这表明，一旦获得足够的观测信息，滤波器就能迅速校正动力变量。整个模拟过程中，均值周围的集合离散度几乎保持不变，表明不确定性表示稳定，且集合塌缩程度有限。

表面质量平衡的时间演变（图 5 (c)）表明，其对同化观测的敏感性更高，且这种变化一直持续到最终同化阶段。约 60 年后，滤波器成功将 $smb$ 校正至真实值附近，表明 ICESEE 能够恢复那些未被直接观测、缓慢演变或约束较弱的参数。

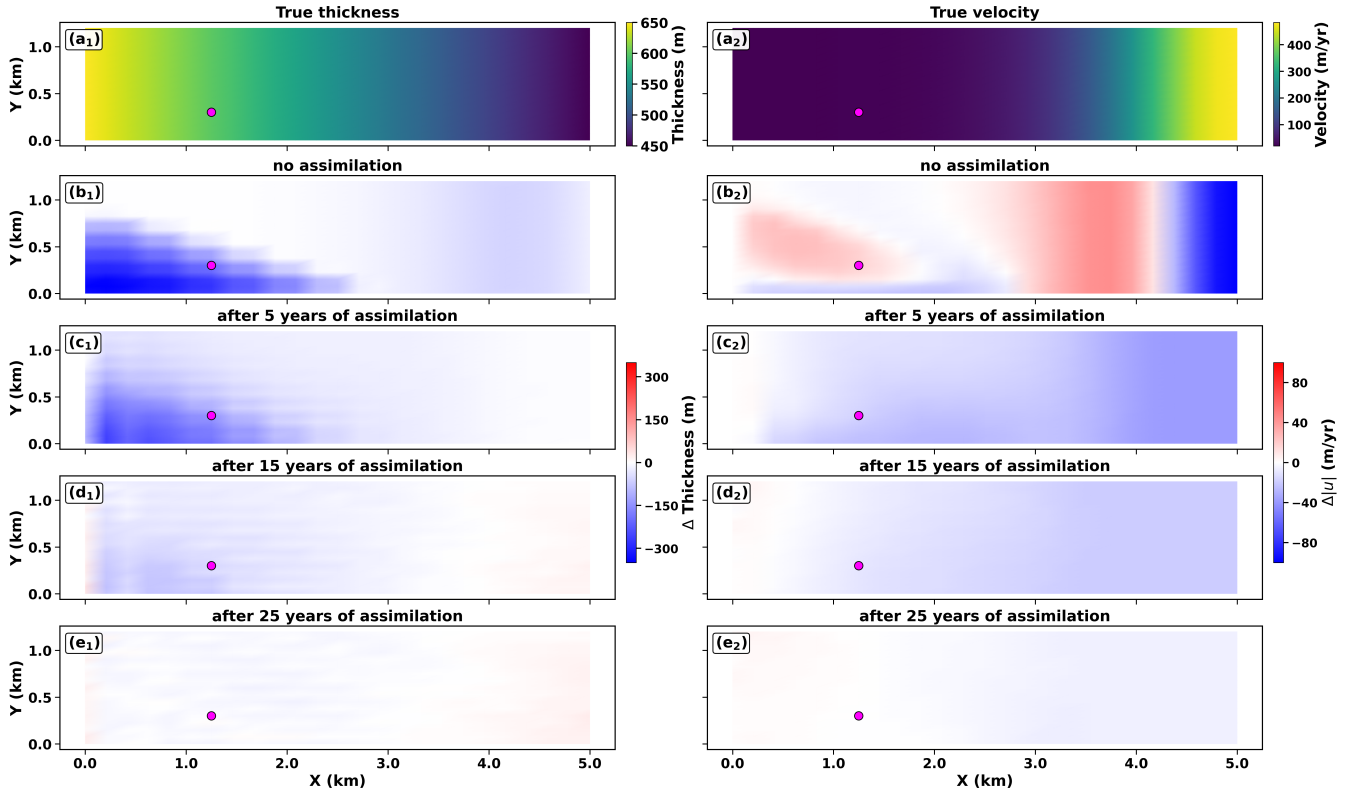


图 4：**左侧：**($a_1$) 显示零时刻的真实冰厚度。($b_1$) 显示未进行同化的模拟结果与零时刻真实冰厚度之间的差值。面板 ($c_1$–$e_1$) 分别显示同化后 5 、 15 和 25 年的厚度场与真实厚度场之间的差值。**右侧：**($a_2$) 显示零时刻的真实速度大小。($b_2$) 显示未进行同化得到的速度与零时刻真实速度大小之间的差值。面板 ($c_2$–$e_2$) 分别显示同化后 5、15 和 25 年的速度大小与真实速度大小之间的差值。洋红色标记表示位于计算域受扰动（隆起）区域内的参考点 (1.25 km 0.3 km)。 $\Delta|u|$ 表示速度大小的有符号差值。

总体而言，这些结果证实，ICESEE 通过联合状态–参数估计，有效约束了耦合冰盖系统的快、慢分量，实现了稳健且稳定的收敛。

### 4.2 冰盖与海平面系统模型（ISSM）

ISSM 是一个全面的冰盖建模框架，能够捕捉支配冰动力学的多种物理过程，包括冰流、热力学，以及冰盖与海洋和大气之间的相互作用。ISSM 旨在模拟冰盖在长时间尺度上的演变，深入揭示其对海平面上升的影响及其敏感性

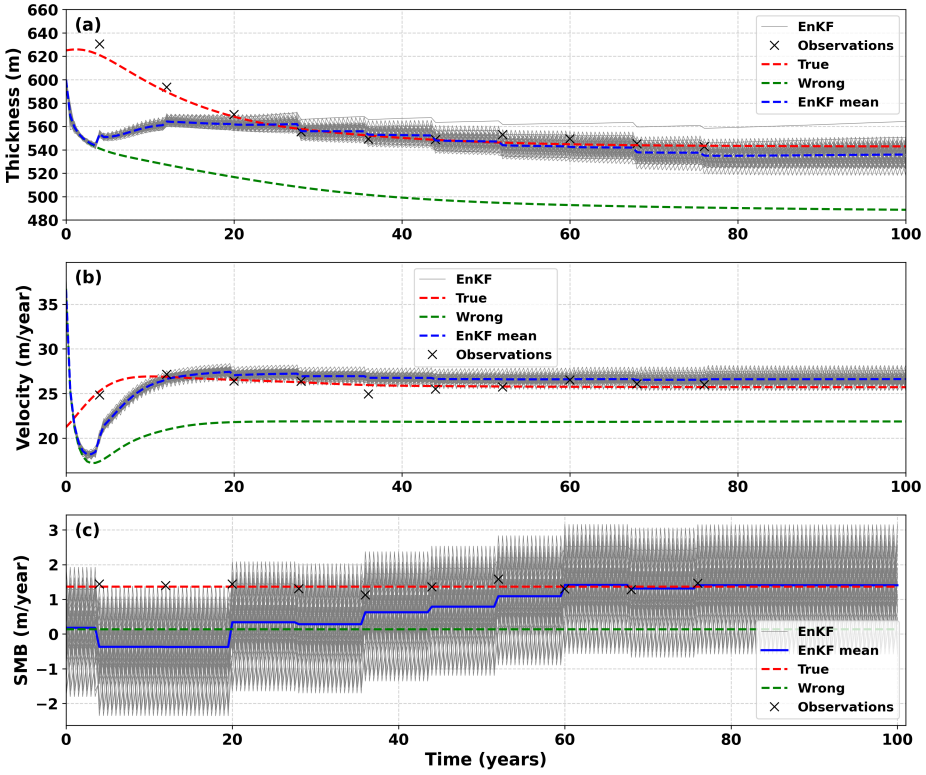


图 5：图 4 中洋红色标记所示参考点处冰厚度 (a)、速度 (b) 和表面质量平衡 (c) 的时间演变。

取决于气候变化。该模型采用 C++实现，并提供基于 MATLAB 的接口，用于前处理与后处理，从而支持灵活的配置和分析工作流（Larour 等，2012年）。

在本研究中，我们通过将 ICESEE 与 ISSM 耦合，并使用集合卡尔曼滤波（EnKF）进行数据同化，展示了 ICESEE 的功能。该集成展示了 ICESEE 如何通过纳入合成观测来约束并预报关键变量，从而提升 ISSM 的预测性能。这些变量包括接地线位置、厚度、速度和基底地形。

ISSM 应用凸显了 ICESEE 在对接复杂建模系统方面的适应性，这类系统需要专门的集成技术。具体而言，我们采用基于 MATLAB 服务器的通信方法，实现 ICESEE（Python）与 ISSM（MATLAB/C++）之间的无缝交互，展示该框架如何跨越语言和平台的边界，以支持高级数据同化任务。

### 4.2.1 MATLAB 服务器

MATLAB 服务器类（*MatlabServer*）是 ICESEE–ISSM 集成的关键组件，可实现 Python 与 MATLAB 之间高效、自动化的通信。用户无需手动运行 MATLAB 脚本，即可通过它让 ICESEE 执行 ISSM 命令并获取模型输出。服务器会持续监听由 Python 环境发出的命令，在 MATLAB 中执行这些命令，然后将结果返回给 ICESEE。为确保通信不会阻塞，MATLAB 输出通过多线程处理，从而能够在 Python 端实时捕获并显示 MATLAB 控制台输出，同时不中断 ICESEE 的主要进程。根据集合成员的数量，服务器可以通过启动与集合成员数量或请求的进程数相同的独立 MATLAB 实例，管理多个并发的 ISSM 运行。

我们采用两级并行策略：每个 MATLAB 实例都初始化为指定数量的内部秩，而 ICESEE 则负责在这些实例间分配集合成员通过这一设计，ISSM 和 ICESEE 均可实现完全并行化执行。服务器由位于 *mat2py_utils.py* 脚本中的 *issm_model/issm_utils/matlab* 实现

目录。该模块包含管理服务器进程所需的所有支持例程，包括启动和终止 MATLAB 实例、发送命令代码、处理异常（例如终止卡住的 MATLAB 进程）以及协调通信。

MATLAB 与 Python 之间的通信采用基于文件的并行输入/输出实现，如第 3.3.2 节所述。服务器经过设计，具有稳健性和灵活性，并可通过配置调整行为以支持不同的使用场景。尽管该服务器专为 ISSM 量身打造，但它具有通用性，也可适配其他需要类似 Python–MATLAB 耦合策略的模型。

### 4.2.2 ISSM：SyntheticTwin 配置

我们考虑一个经过修改的 ISMIP 基准问题，该问题最初由 Gillet-Chaulet（2020年）描述，后由 Choi 等（2025年）改编，用于模拟海洋终止冰川的三维长期演化。模型域是一个尺寸为 $L_x = 640$ 千米 × $L_y = 80$ 千米的矩形区域，代表理想化的海洋终止冰盖几何形状。所有模拟均使用 ISSM 4.24 版进行，并与 ICESEE 耦合以开展基于集合的数据同化。我们参考 Choi 等（2025年）的一般配置构建参考（真实）模拟，并针对基底和基底地形以及网格离散化进行调整，以确保当前案例研究能够得到稳态解。具体而言，我们调整冰床（$b$）几何系数，以控制特征侧壁宽度（$f_c$）、槽谷半宽（$w_c$）以及槽谷相对于侧壁的深度（$d_c$）。这些参数分别从 $f_c = 4000$ 调整为 $3000\,m$、从 $w_c = 24000$ 调整为 $23000\,m$，以及从 $d_c = 500$ 调整为 $400\,m$。我们还修改了表面和基底参数化方案：表面高程从 $\max(b_x(x) + b_y(0) + 100, 10)$ 改为 $\max(b_x(x) + b_y(0) + 300, 20)$，而基底则从 $\max(b, -90)$ 调整为 $\max(b, -280)$。计算域采用网格离散化，在 $x$ 方向上划分为 $n_x = 222$ 个网格单元，在 $y$ 方向上划分为 $n_y = 56$ 个网格单元（对应的平均水平间距约为： $x$ 方向 2.9 千米， $y$ 方向 1.4 千米；网格为非均匀网格，局部单元尺寸从浮冰区域精细网格中的约 500 米到接地冰粗网格中的 10 千米不等

区域）。完成参数化后，我们向基底地形添加指数噪声，并从区域内每个节点的数值中减去 284 米的常数偏移量。完成这些修改后，我们进行了 25,000 年的自旋模拟，以获得稳态参考配置。表面质量平衡（$smb$）保持与 Choi 等（2025年）一致，而基底融化率则从每年 200 米降至每年 150 米。随后，模型向前运行额外 100 年，生成用于同化实验的最终“真实”参考模拟。

ICESEE 状态向量定义为

$$\mathbf{X} = [h,\, s,\, u,\, v,\, b,\, c],$$

其中，$h$、$s$、$u$ 和 $v$ 分别表示状态变量（冰厚度、表面高程和水平速度），而 $b$和$c$表示在同化框架中联合估计的状态参数（基底地形和底部摩擦系数）。预报步骤中用于伪随机场生成的相应去相关长度尺度分别为 $[88, 85, 90, 85, 88, 100]$ 米和 $[h,\, s,\, u,\, v,\, b,\, c]$。

遵循 Choi 等（2025）的方法，我们使用 *GSTools* 生成空间相关随机场：为底部摩擦系数生成高斯随机场，并生成以基底观测为条件的冰床地形指数克里金场。我们调整摩擦变异函数参数，使其更好地匹配调整后的基底几何形态：基台值从 90,000 降至 10,000，变程从 5,000 降至 $500\ m$。集合初始化时，利用生成的摩擦噪声对平均摩擦场（$2500\ Pa\ m^{-1/3} yr^{1/3}$）进行微调，同时针对每个集合成员，使用基底噪声扰动冰床、底部和表面场。此外，在整个区域内，从冰床和底部高程中减去统一偏移量 50 米，并从表面高程中减去 25 米。这些扰动场作为 ISSM 瞬态求解器的“错误”初始条件，从而得到集合初始状态 $\mathbf{X}$；选取这些状态是为了使其具有代表性，反映冰盖模拟初始条件中所用误差的量级。

通过向真实状态添加加性高斯噪声，生成表面高程和速度 $x$分量及 $y$分量的合成观测。规定的标准差为：表面高程 $\sigma_s = 2.5$ m，水平速度分量 $\sigma_u$ =3.0 m $\mathrm{yr}^{-1}$，基底地形 $\sigma_b = 2.0$ m。假设在整个区域内均可获得表面高程和速度观测（类似于可获取卫星测高和合成孔径雷达数据的区域）；而基底地形观测仅限于二维采样掩膜，该掩膜由沿 $x$方向排列、相邻间隔为 4000 m 的机载雷达航线组成，以模拟真实的机载测量。基底地形在第 2 年观测一次，而表面高程和速度观测每两年同化一次，直至模拟的第 24 年。冰厚 $h$ 不直接观测；而是在同化过程中通过模型状态变量与参数之间的交叉协方差进行推断。

底部摩擦无法直接观测，而是在每个同化循环中，结合速度集合、合成速度观测、有效压力（根据同化后的冰厚度计算）以及接地线演变（根据同化后的厚度和基底地形推导）进行推断。这一推断通过在 ISSM 中采用传统冰盖速度反演方法（MacAyeal，1989年；Morlighem 等，2013），并将其与 ICESEE数据同化框架耦合来实现，同时应用蒂霍诺夫正则化以稳定反演过程。底部摩擦系数

被约束在 2000 至 4000 Pa $\mathrm{m}^{-1/3}\mathrm{yr}^{1/3}$ 之间，并在整个区域内统一应用。速度失配的绝对权重和相对权重均设为 1，而正则化权重固定为 $10^{-13}$。每当有速度数据可用时，便会进行底部摩擦系数的速度反演，并替代先前的底部摩擦估计。

在 ICESEE 数据同化运行中，我们采用 0.2 年的时间步长，并向前积分共 50 年，足以捕捉本研究关注的效应。本实验的集合规模同样固定为 $N = 40$。由于冰床仅观测一次，我们在快照时刻采用自适应欠松弛策略，

$$b^{\text{new}} = b^{\text{prior}} + \alpha(t)\left(b^{\text{assimilated}} - b^{\text{prior}}\right). \tag{18}$$

松弛参数 $\alpha(t)$由确定型和随机型两部分构成，

$$\alpha(t) = (\eta + \beta_t)\Delta t + \sqrt{\Delta t}\,\sigma_b\,\rho, \tag{19}$$

其中， $\eta$ 是基准调整系数， $\beta_t$ 是集合推导的偏差项， $\rho$ 是随机参数（其计算方法参见 Evensen（2003））。这种结构使更新幅度能够根据集合统计量进行调整，同时引入受控的变异性。松弛参数上限设为 $\alpha(t) \leq 0.5$，以确保更新具有压缩性，将施加的修正限制在推断所得冰床增量的至多 50%。这种受控更新可缓解滤波器过度调整、抑制虚假的高频噪声，并在后续没有冰床观测的情况下稳定集合。

### 4.2.3 ISSM：SyntheticTwin 结果

这里，我们利用上述 ICESEE 与 ISSM 的耦合，评估 ICESEE 数据同化框架纠正错误模型状态$\mathbf{X}$ 的能力。我们在图6、图7和图8中总结了结果，这些结果表明，耦合按预期发挥作用，且 ICESEE 中实现的数据同化方法成功地将模型状态校正至“真实”状态。在图6左侧，冰床地形仅在第2年被直接观测了一次。此后的校正是通过集合卡尔曼滤波学习模型变量与参数之间的交叉协方差，从间接信息中实现的。总体而言，在最初几个同化循环之后，接地线周围100千米范围内以及冰川中央槽谷中的相对误差均低于10%；不过，在上游局部区域和缓慢流动的边缘地带，相对误差仍然较高。其他利用数据同化估计冰床地形的研究也发现了类似的规律：快速流动区域的冰床性质估计更准确（Gillet-Chaulet，2020年；Brinkerhoff 等人，2025年；Choi 等人，2025年）。然而，在选定时间步长上冰床高程相对误差总体降低，证明即使观测有限，这种间接校正仍然有效。

图6右侧（$\mathrm{b}_2$–$\mathrm{h}_2$）展示了采用和未采用数据同化的模拟中，冰厚度和接地线位置时间演变的差异。通过 ICESEE 实施同化后，冰厚度误差显著降低；尽管这些模拟并未显式同化接地线位置，观测到的接地线后退仍与“真实”状态高度吻合。图8进一步佐证了冰厚度和接地线位置的改善。图8（a）和（d）分别使用冰厚度的均方根误差（RMSE）以及接地线位置的绝对差值，对这些修正进行量化；比较对象包括具有“真实”状态的模拟、同化

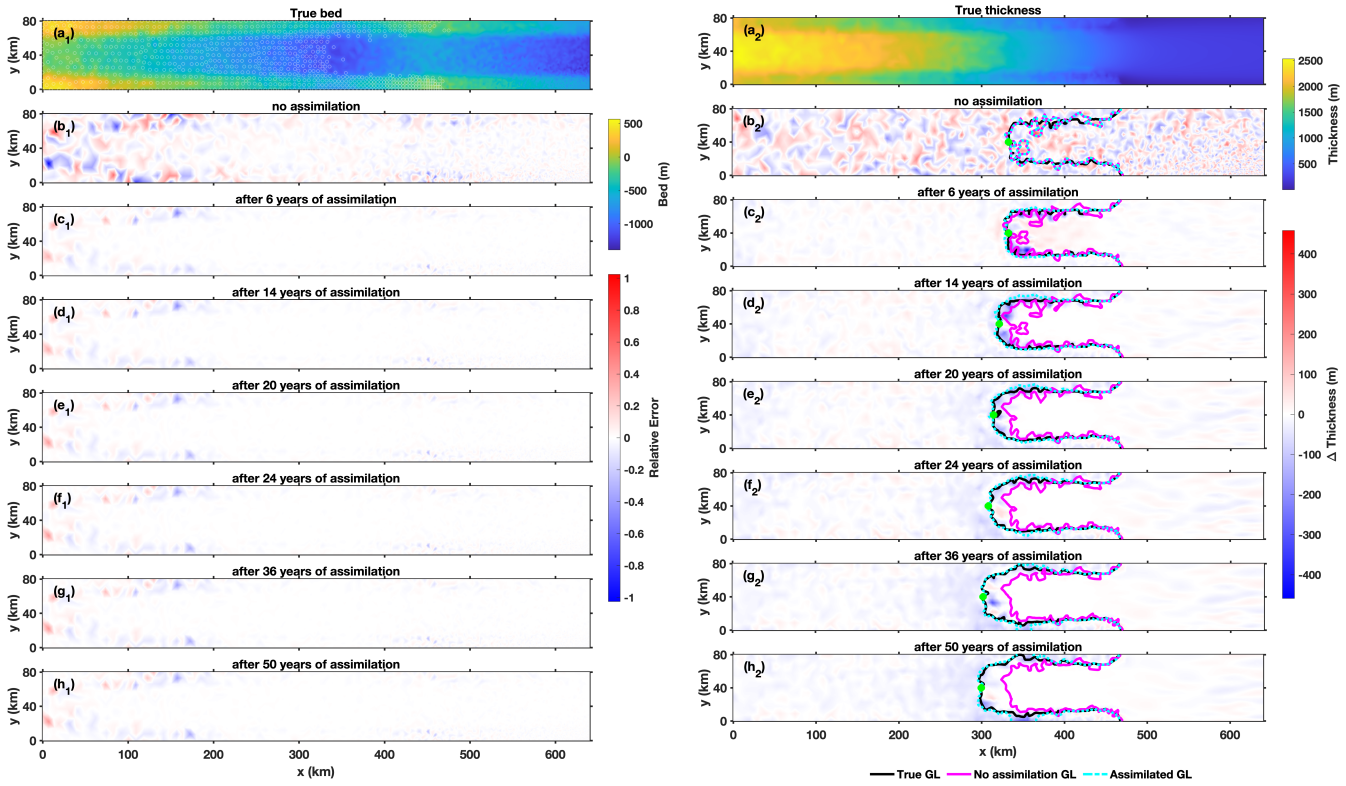


图6：**左列：**参考海床地形（$a_1$）、未同化模拟相对于真实状态的海床高程误差（$b_1$），以及选定时刻同化模拟相对于真实状态的海床高程误差（$c_1$–$h_1$）。叠加在面板 $a_1$ 上的小白圆标示了第 2 年同化的基底观测位置。**右列：**初始时刻的参考冰厚度（$a_2$）、未同化模拟与真实模拟之间的冰厚差（$b_2$），以及同化状态与真实状态之间冰厚差的时间演变（$c_2$–$h_2$）。黑色实线、洋红色实线和青色虚线分别表示真实状态、未同化状态和同化状态的接地线（GL）位置。真实接地线上的绿色实心点表示中心线与接地线的交点。

（虚线）和未进行同化的模拟（实线）。测量沿着水平中心线进行，该中心线穿过图6左侧所示的绿色实心圆点。未进行同化的情况下（实线），冰厚均方根误差（图8a）逐渐增大；而进行同化的情况下（虚线），第一年内均方根误差迅速下降，从约80 m降至30 m以下，此后在模拟剩余时间内基本保持稳定，因为主要的初始偏差已得到修正。类似地，图8（d）显示，采用同化的模拟中接地线位置（实线）逐渐偏离真实状态，几十年内累积的误差超过30 km。采用同化的情况下（虚线），接地线位置逐渐趋近真实接地线，最终误差小于网格间距。这一表现与图6所示的接地线演变一致，并证实 ICESEE 数据同化能够有效修正冰床误差、改善接地线演变，而无需显式同化

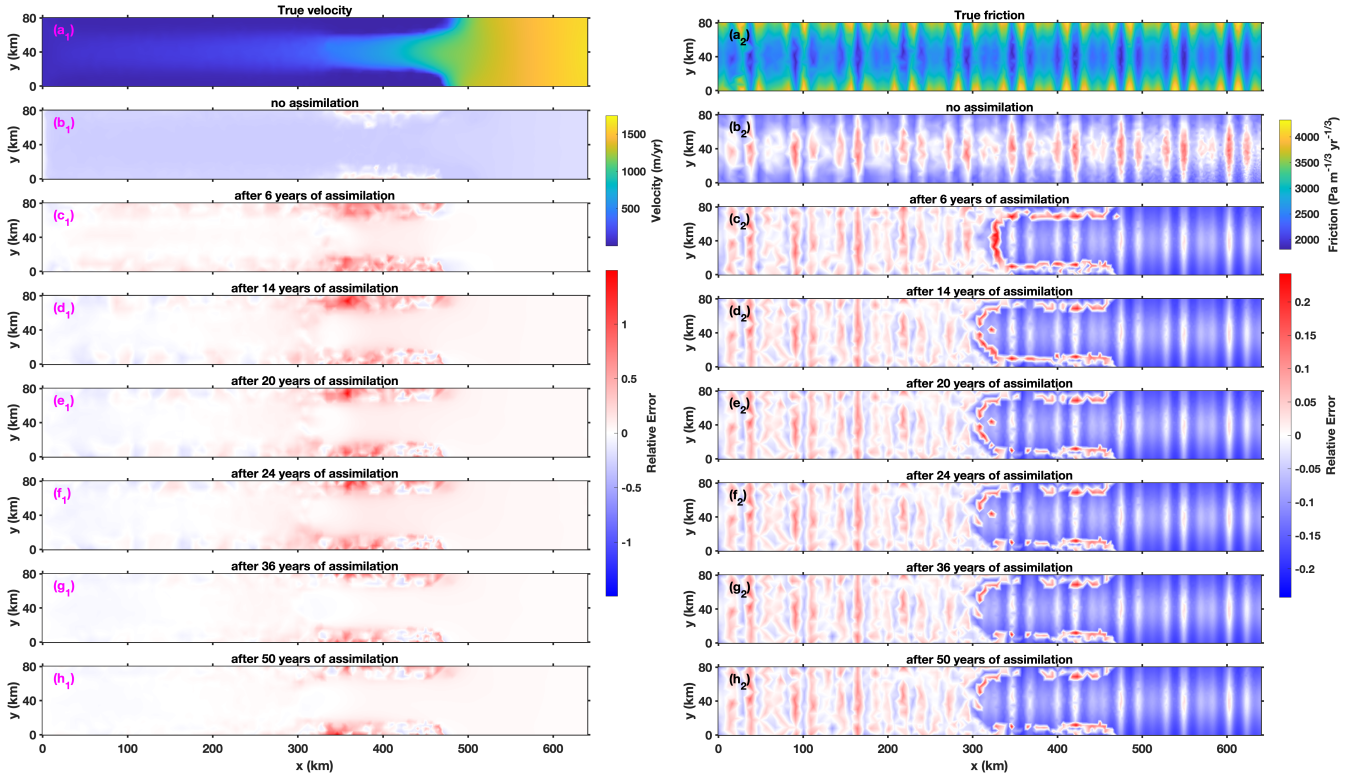


图7：**左侧：**时间 0 ($a_1$, $b_1$) 时的参考速度场和相对于真实状态的未同化速度误差，随后是所选时间点 ($c_1$–$h_1$) 相对于真实速度的同化速度误差。**右侧：**参考底部摩擦系数 ($a_2$)、未同化状态与真实状态之间的底部摩擦系数差值 ($b_2$)，以及连续同化循环中同化后的底部摩擦系数与真实底部摩擦系数之间的差值 ($c_2$–$h_2$)。

观测到的接地线位置。床面修正仍然相对较小，这符合预期，因为初始观测集之后可用的基底观测有限。

ICESEE 能够通过更传统的反演方法（例如 ISSM 中实现的确定型伴随反演框架）更新未观测参数的估计值，例如底部摩擦系数。由于黏性冰盖流动是非惯性的，所有现代冰盖模型中的速度求解器（其中包含底部摩擦系数）完全是诊断型的。因此，对于 ICESEE 中实现的序贯数据同化方法等方法，通过纯反演方法更新底部摩擦可能具有一定价值。ICESEE 支持用户选择是否通过集合卡尔曼滤波或传统反演来更新底部摩擦场。选择反演选项后，在预报步骤中，完整的模型状态向量 **X** 会随时间向前演化。然而，在每个同化步骤中，**X** 会缩减并排除摩擦系数；随后，利用厚度、表面高程、速度和基底地形之间的交叉协方差，以及整个区域内均可获得的速度和地表观测，并结合仅在第 2 年获取的受限基底观测，对其他变量和参数进行同化。在每次

同化步骤之后，ICESEE 会利用同化后的厚度和速度场、速度观测以及先验（初始）摩擦场，通过 ISSM 的反演框架（Morlighem 等，2013）及其相关代价函数推断底部摩擦系数。随后，推断出的摩擦场会被重新添加到模型状态 **X** 中，以供下一次预报使用。

图7右侧（$c_2$–$f_2$）展示了所选同化时刻真实场与同化场之间推断出的摩擦差异。第24年之后，未观察到摩擦系数的进一步更新——见（$g_2$–$h_2$），因为此时已达到最后一个观测时刻。图7展示了时刻0的“真实”速度（$a_1$），以及无同化情形（$b_1$）和有同化情形（$c_1$–$h_1$）下相对于真实状态的速度误差。结果表明，同化情形有了显著改善：第2年首次反演后，接地区域的基底摩擦误差降至5%以下，接地线上游和下游的速度误差降至约10%；随着反演纳入其他状态变量，结果继续逐步改善，这些变量也在后续同化循环中逐渐改善。浮冰下方的摩擦设为零，以免较大的速度对反演造成偏差。因此，同化能够更有效地约束接地区域的摩擦，使每个同化循环中的速度与真实状态更加吻合。图8（b）和（c）所示的速度和摩擦均方根误差诊断进一步支持了这些发现。速度均方根误差在同化时刻出现明显峰值，这反映了速度求解器的诊断特性以及速度观测的稀疏性。

与接地线区域内或下游相比，ICESEE 在接地线以上游的接地冰区域中表现更佳，冰厚度（图6，$b_2$–$h_2$, 面板和图8a）、速度（图7，$b_1$–$h_1$, 面板和图8b）以及基底摩擦（图7，$b_2$–$h_2$, 面板和图8c）的结果均证实了这一点。例如，图8显示，对于不包括接地线的接地冰区域，基底摩擦的同化均方根误差（蓝色虚线）始终低于其未同化对应值，也低于针对包括接地线在内的接地冰区域计算的同化均方根误差。速度和厚度也呈现类似情况；对于速度和厚度，整个区域的同化均方根误差也低于未同化情况。这一现象反映出，滤波器最能从观测和模型物理过程均能良好约束的变量和参数中学习，而这些条件主要在接地冰区域得到满足。相比之下，接地线区域具有强非线性和较高的模型不确定性，因而限制了滤波器的效能。尽管如此，这些结果表明，ICESEE 能够通过与 ISSM 的耦合，成功利用跨变量协方差传播冰床信息，并改善冰盖几何形态和动力学。

## 5 性能运行

我们使用第 3.4.2 节中介绍的完全并行化配置，通过强扩展和弱扩展实验评估 ICESEE-ISSM 耦合框架的计算性能。我们的目标是评估 ICESEE 与大型冰盖模型耦合时的可扩展性和效率；在这种情况下，内存需求和输入/输出操作占主导地位，而且在耦合期间，ICESEE 无法轻易覆盖宿主模型（ISSM）的 MPI 通信器。所有模拟均在佐治亚理工学院的高性能计算集群**Phoenix** （PACE，2017）上进行

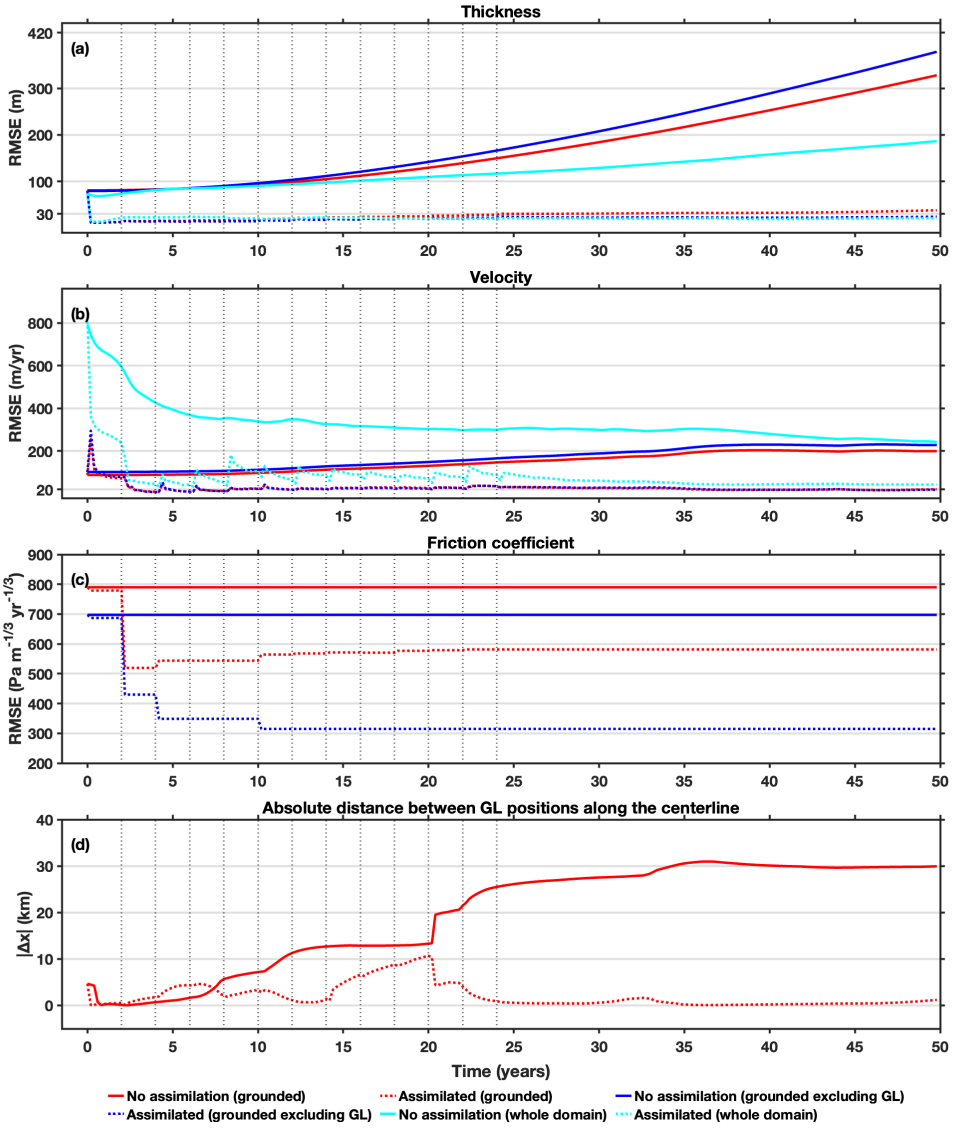


图8：冰厚度、速度和底部摩擦系数的均方根误差（RMSE）的时间演变（a–c），以及沿水平中心线测量的同化状态与真实状态、未同化状态与真实状态之间接地线位置的绝对差异。该中心线经过图6右侧真实接地线上的绿色实心点（d）。细灰色竖直点线表示同化时刻。红色、蓝色和青色点线分别表示在接地冰、接地冰（不包括接地线上游区域）和整个区域上评估的同化运行；实线表示相应的未同化情况。

技术。Phoenix 上的每个计算节点均配备双路英特尔至强 CPU（每个节点 24 个核心）和 192 GB 随机存取存储器，并通过 Mellanox HDR InfiniBand（100–200 Gb/s）互连。

为单独评估 ICESEE 本身的性能，在整个扩展性实验中，我们保持 ISSM 进程数（模型_进程数）固定，因为我们的重点不是评估 ISSM 的独立性能，而是考察 ICESEE 与计算密集型模型集成时的效率。我们采用与第 4.2 节所述相同的物理和数值设置，唯一的区别是将模拟时长缩短至 40 年；这一时长足以代表真实的工作负载，同时还能使扩展性测试的运行时间保持在可控范围内。

在两项扩展性测试中，问题规模和 ISSM 进程数保持不变，而 ICESEE 进程数从 5 增至 320。每个 ISSM-MATLAB 实例使用四个进程，另预留一个进程用于 ICESEE 控制，因此每个集合成员总计使用五个进程。因此，耦合系统中的 MPI 进程总数为

$$\text{ICESEE 进程数} = N_p \times （\text{模型_进程数} + 1），\tag{20}$$

其中，$N_p$ 表示分配给每个 ISSM 实例的进程数。

### 5.1 强扩展

在强扩展实验中，集合规模和问题规模均保持不变，同时改变 ICESEE 进程数。加速比根据阿姆达尔定律（Amdahl，1967）计算如下：

$$S_p = \frac{T(1)}{T(N_p)},\tag{21}$$

其中，$T(1)$ 表示使用单个 ICESEE 进程顺序执行多个 ISSM 实例时的实际运行时间，而 $T（N_p）$ 表示使用 $N_p$ 个 ICESEE 进程完成相同工作负载所需的时间。并行效率定义为：

$$E_p = S_p \times \frac{N_1}{N_p}.\tag{22}$$

随着资源数量增加，总实际运行时间近乎成比例地减少，直至约 80 个进程；超过这一数量后，加速比和效率开始趋于平缓（表 1）。这一下降主要是由于 MATLAB 实例数量增加，以及耦合过程中 MATLAB 与 Python 之间的通信开销不断增大。尽管 ICESEE 中的预报步骤已完全并行化，但分析阶段和并行输入/输出操作仍会耗费额外时间。

图 9 详细分解了各组成部分的总运行时间。在所有配置中，预报 步骤（橙色）占据了计算成本的主要部分，但随着 MPI 进程数增加，其耗时显著减少，证实了并行性有效。分析 （蓝色）和输入/输出 （红色）组件的扩展性也较好。不过，由于同步和通信开销，它们在总运行时间中的相对占比会随着进程数增加而上升。总体而言，结果表明，在测试范围内具有良好的可扩展性，并能有效利用计算资源。

表 1：ICESEE-ISSM 耦合的强扩展性能。

| 节点 | ICESEE 进程数 | ISSM 进程数 | $N_{\text{ens}}$ | 墙钟时间（分钟） | 效率（%） | 加速比 |
|---|---|---|---|---|---|---|
| 1 | 5 | 4 | 64 | 695.69 | 100.00 | 1.00 |
| 1 | 10 | 4 | 64 | 365.01 | 95.30 | 1.91 |
| 1 | 20 | 4 | 64 | 206.85 | 84.08 | 3.36 |
| 2 | 40 | 4 | 64 | 116.55 | 74.61 | 5.97 |
| 4 | 80 | 4 | 64 | 81.31 | 53.47 | 8.56 |
| 7 | 160 | 4 | 64 | 44.27 | 49.10 | 15.71 |
| 14 | 320 | 4 | 64 | 26.06 | 41.72 | 26.70 |

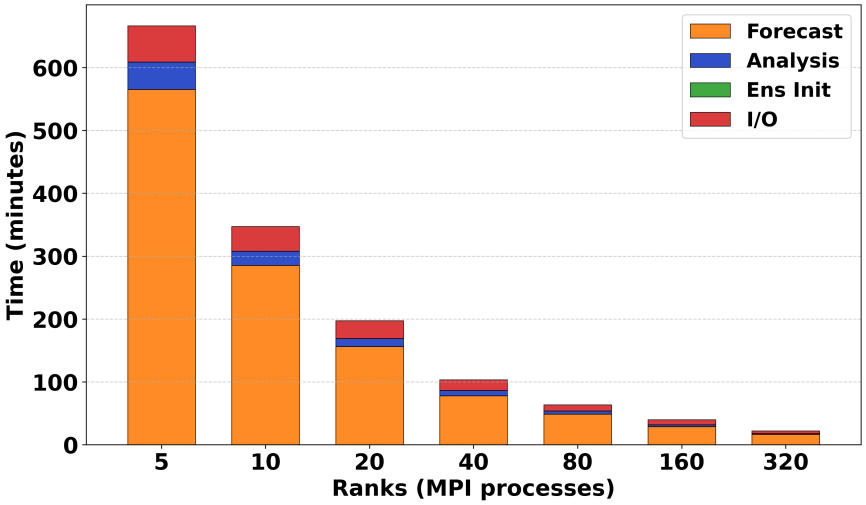


图 9：ICESEE-ISSM 耦合的强扩展运行时间分解，展示了预报、分析、集合初始化和输入/输出各自的耗时。预报步骤的扩展效率最高，而随着进程数增加，输入/输出和分析的相对开销变得更加明显。

### 5.2 弱扩展

在弱扩展实验中，我们按比例增加问题规模和集合成员数量，使每个 ICESEE 进程的工作负载保持不变。具体而言，我们从 4 个集合成员和 5 个 ICESEE 进程开始，同时保持 ISSM 进程数不变。随后，每当集合成员数量翻倍时，ICESEE 进程数也翻倍，这遵循 Gustafson（1988）提出的弱扩展模型。此配置使我们能够评估随着问题总规模增长，ICESEE-ISSM 耦合的表现，同时保持每个进程的计算负载固定

表 2：ICESEE–ISSM 耦合的弱扩展性能。

| 节点 | ICESEE 进程数 | ISSM 进程数 | $N_{\text{ens}}$ | 墙钟时间（分钟） | 效率（%） |
|---|---|---|---|---|---|
| 1 | 5 | 4 | 4 | 78.56 | 100.00 |
| 1 | 10 | 4 | 8 | 81.59 | 96.28 |
| 1 | 20 | 4 | 16 | 83.38 | 94.22 |
| 2 | 40 | 4 | 32 | 89.92 | 87.36 |
| 4 | 80 | 4 | 64 | 97.90 | 80.24 |
| 7 | 160 | 4 | 128 | 108.90 | 72.13 |
| 14 | 320 | 4 | 256 | 127.87 | 61.43 |

进程。弱扩展效率的计算公式为

$$E_{\text{weak}} = \frac{T(1)}{T(N_p)}, \tag{23}$$

其中， $T(1)$ 是 ICESEE-ISSM 基准配置（5 个进程和 4 个集合成员）所需的实际运行时间， $T(N_p)$ 是扩展到 $N_p$ ICESEE 进程数时测得的实际运行时间。

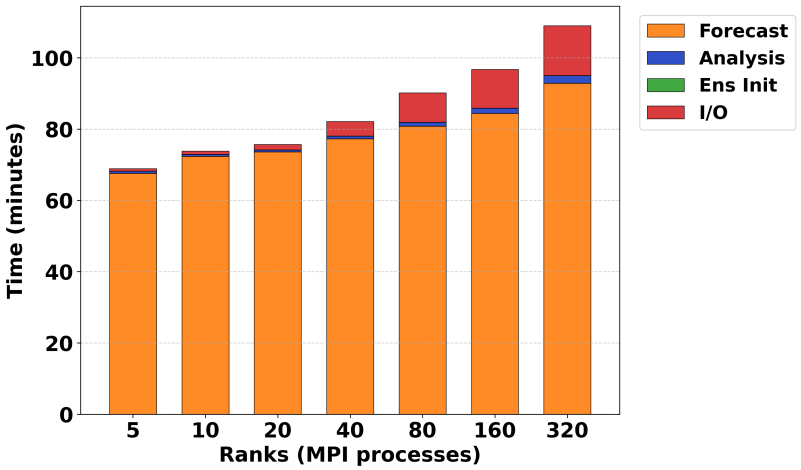


图 10：ICESEE–ISSM 耦合的弱扩展运行时间分解，展示了预报、分析、集合初始化和输入/输出所占的时间。预报阶段仍占主导地位，且扩展性良好；而在进程数较高时，由于集体数据同步，输入/输出开销略有增加。

如表 2 所示，随着进程数和集合成员数增加，总实际运行时间仅小幅上升。从 5 个进程和 4 个集合成员时的 78.56 分钟，增加到 320 个进程和 256 个集合成员时的 127.87 分钟。这种逐步增长表明，ICESEE 在大规模集合情况下仍保持良好的并行可扩展性

规模。随着系统规模扩大，整体弱扩展效率从 100% 降至约 61%，这主要是由于通信开销、分布式集合成员之间的 I/O 同步成本，以及并发 ISSM 实例数量增加所带来的额外预报同步开销。

图 10 展示了总运行时间按组件划分的情况。与强扩展结果类似，预报阶段（橙色）占据了总计算成本的主要部分，并且在所有配置下都具有良好的扩展性。分析（蓝色）和集合初始化（绿色）阶段对总体运行时间的贡献很小，而输入/输出（红色）的相对成本在进程数较高时略有增加。观察到的输入/输出时间增长可归因于多个集合成员和节点之间的集体数据写入与读取同步。尽管存在这些通信和输入/输出影响，ICESEE 在弱扩展下仍展现出高效的并行性能，证实其设计能够有效支持大规模集合数据同化问题。

## 6 结论

ICESEE 是一个灵活且可扩展的数据同化框架，旨在通过先进的集合卡尔曼滤波技术和紧密耦合的反演方法，增强冰盖模型的预测能力。通过支持与基于 Python 以及 MATLAB/C++ 的模型集成，ICESEE 提供了一个统一平台，用于冰盖动力学中的实时状态估计、参数推断和预报。该框架集成了多种并行化策略，可在高性能计算（HPC）系统上实现高效执行，并提供能够适应广泛建模架构的灵活耦合机制。

ICESEE 已成功应用于两个具有代表性的冰盖模型 Icepack 和 ISSM，展现了其适应性以及通过同化合成观测来提高模型精度的有效性。与 Icepack 的集成凸显了 ICESEE 与基于 Python 的建模工作流无缝对接的能力，而与 ISSM 的耦合则展示了它通过专用 MATLAB 服务器与基于 MATLAB 的系统协同运行的能力。在 ISSM 应用中，ICESEE 还能够通过将同化后的状态变量与 ISSM 的反演建模能力相结合，间接估计底部摩擦系数等未观测参数。这种同化–反演混合方法使 ICESEE 能够利用跨变量协方差进行状态校正，同时借助基于物理的代价函数最小化推断观测不足的参数，从而改善冰体几何形态、速度场和接地线演变。

性能分析证实，ICESEE 在高性能计算系统上实现了良好的强扩展和弱扩展效率，凸显了其适用于大规模数据同化冰盖模拟的优势。总体而言，ICESEE 是迈向统一高性能框架的重要一步，该框架将基于集合的数据同化与反演建模相结合，推动新一代冰川学建模中的状态估计、参数推断和不确定性量化。

**致谢**

我们感谢佐治亚理工学院位于佐治亚州亚特兰大的先进计算环境合作组织（PACE）提供计算资源，使本研究得以完成；相关计算额度由佐治亚大学系统通过一项启动经费提供。我们还要感谢研究科学家 Fang (Cherry) Liu 在解决与 PACE 和高性能计算相关的挑战时给予的帮助。

**经费支持**

作者和 ICESEE 项目获得美国国家科学基金会极地项目办公室授予的 2235920 号资助。

. ICESEE 当前版本可在 GitHub 上获取：https://github.com/ICESEE-project/ICESEE，并采用 BSD 两条款许可证。模型的确切版本（包括用于运行模型并生成本文所述结果的所有配置和绘图脚本）已存档于 Zenodo：https://doi.org/10.5281/zenodo.18716132（Kyanjo 和 Robel，2026年）。

. Brian Kyanjo：概念构思、软件开发、数值实验及初稿撰写。Talea L. Mayo：数据同化方法指导及稿件编辑。Alexander Robel：监督、将工作整合到冰盖建模中，以及稿件修订。Talea L. Mayo 和 Alexander Robel 均提供了重要反馈，并负责项目的整体指导。

. 作者声明不存在任何利益冲突

## 参考文献

Ahmed, S. E., Pawar, S., 和 San, O.：《PyDA：使用 Python 进行动力数据同化的实践入门》，《流体》，5，225，2020年。Aksoy, A., Zhang, F., 和 Nielsen-Gammon, J. W.：《基于集合的状态与参数同步估计：MM5》，《地球物理研究快报》，33，https://doi.org/https://doi.org/10.1029/2006GL026186，2006年。Amdahl, G. M.：《单处理器方法实现大规模计算能力的有效性》，载于：《1967年4月18日至20日春季联合计算机会议论文集》，AFIPS '67（春季），第483–485页，美国计算机协会，美国纽约州纽约市，https://doi.org/10.1145/1465482.1465560，1967年。Aschwanden, A., Fahnestock, M. A., Truffer, M., Brinkerhoff, D. J., Hock, R., Khroulev, C., Mottram, R., 和 Khan, S. A.：《格陵兰冰盖对未来一千年海平面变化的贡献》，《科学进展》，5，eaav9396，https://doi.org/10.1126/sciadv.aav9396，2019年。Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., 和 Zhang, J.：《PETSc 网页》，https://petsc.org/，https://petsc.org/，2024年。Bergemann, K. 和 Reich, S.：《集合卡尔曼滤波器的局地化技术》，https://arxiv.org/abs/0909.1678，2010年。Bishop, C. H., Etherton, B. J., 和 Majumdar, S. J.：《使用集合变换卡尔曼滤波器的自适应采样。第一部分：理论方面》，《月刊天气评论》，129，420–436，https://doi.org/10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2，2001年。Bonan, B., Nodet, M., 和 Ritz, C.：《基于集合卡尔曼滤波的数据同化，用于冰盖初始化》，载于：2013年欧洲地球科学联盟全体大会，奥地利维也纳，https://inria.hal.science/hal-00837845，2013年。Bonan, B., Nodet, M., Ritz, C., 和 Peyaud, V.：《一种用于冰盖建模中初始状态和参数估计的 ETKF 方法》，《地球物理学中的非线性过程》，21，569–582，https://doi.org/10.5194/npg-21-569-2014，2014年。Brinkerhoff, D. J., Tober, B. S., Daniel, M., Devaux-Chupin, V., Christoffersen, M. S., Holt, J. W., Larsen, C. F., Fahnestock, M., Loso, M. G., Timm, K. M. 等：《世界最大山麓冰川的消亡：概率预报》，《冰冻圈》，19，2321–2353，2025年。Burgers, G., van Leeuwen, P. J., 和 Evensen, G.：《集合卡尔曼滤波器中的分析方案》，《月刊天气评论》，126，1719–1724，https://doi.org/10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2，1998年。Carrassi, A., Bocquet, M., Bertino, L., 和 Evensen, G.：《地球科学中的数据同化——方法、问题与前景概述》，https://arxiv.org/abs/1709.02798，2018年。Choi, Y., Seroussi, H., Morlighem, M., Schlegel, N.-J., 和 Gardner, A.：《时间依赖的数据同化对冰流模型初始化和预测的影响：凯尔冰川案例研究》，《冰冻圈》，17，5499–5517，https://doi.org/10.5194/tc-17- 5499-2023，2023年。Choi, Y., Petty, A., Felikson, D., 和 Poterjoy, J.：《使用集合卡尔曼滤波器和观测系统模拟实验估计冰盖模型的状态和参数》，EGUsphere，2025年，1–25，https://doi.org/10.5194/egusphere-2025-301，2025年。Deakin, L.：《zarrs：用于 Zarr 数组存储格式的高性能 Rust 库》，https://doi.org/10.5281/zenodo.12719776，2025年。Evensen, G.：《使用非线性准地转模型进行序贯数据同化：利用蒙特卡洛方法预报误差统计》，《地球物理研究杂志：海洋》，99，10 143–10162，https://doi.org/10.1029/94JC00572，1994年。Evensen, G.：《集合卡尔曼滤波器：理论构建与实践应用》，《海洋动力学》，53，343–367，https://doi.org/10.1007/s10236-003-0036-9，2003年。

Evensen, G.：《数据同化：集合卡尔曼滤波》，Springer Science & Business Media，柏林，第2版，https://doi.org/10.1007/978-3-642-03711-5，2009年。Evensen, G., Vossepoel, F. C. 和 van Leeuwen, P. J.：《局地化与膨胀》，第111–122页，Springer International Publishing，尚佩，https://doi.org/10.1007/978-3-030-96709-3_10，2022年。Gillet-Chaulet, F.：在瞬态海洋冰盖模型中使用集合卡尔曼滤波器同化地表观测，《冰冻圈》，14，811–832，https://doi.org/10.5194/tc-14-811-2020，2020年。Goelzer, H., Robinson, A., Seroussi, H. 和 van de Wal, R. S. W.：格陵兰冰盖模拟的最新进展，《当前气候变化报告》，3，291–302，https://doi.org/10.1007/s40641-017-0073-y，2017。Goelzer, H., Nowicki, S., Edwards, T., Beckley, M., Abe-Ouchi, A., Aschwanden, A., Calov, R., Gagliardini, O., Gillet-Chaulet, F., Golledge, N. R., Gregory, J., Greve, R., Humbert, A., Huybrechts, P., Kennedy, J. H., Larour, E., Lipscomb, W. H., Le clec'h, S., Lee, V., Morlighem, M., Pattyn, F., Payne, A. J., Rodehacke, C., Rückamp, M., Saito, F., Schlegel, N., Seroussi, H., Shepherd, A., Sun, S., van de Wal, R. 和 Ziemen, F. A.：冰盖模型初始化实验initMIP-Greenland的设计与结果：一项ISMIP6比较研究，《冰冻圈》，12，1433–1460，https://doi.org/10.5194/tc-12-1433-2018，2018年。Gustafson, J. L.：重新评估阿姆达尔定律，《《ACM通讯》》，31，532–533，https://doi.org/10.1145/42411.42415，1988。h5py 开发者：h5py：适用于Python的HDF5，https://doi.org/10.5281/zenodo.6575970，2022年。Ham, D. A., Kelly, P. H. J., Mitchell, L., Cotter, C. J., Kirby, R. C., Sagiyama, K., Bouziani, N., Vorderwuelbecke, S., Gregory, T. J., Betteridge, J., Shapero, D. R., Nixon-Hill, R. W., Ward, C. J., Farrell, P. E., Brubeck, P. D., Marsden, I., Gibson, T. H., Homolya, M., Sun, T., McRae, A. T. T., Luporini, F., Gregory, A., Lange, M., Funke, S. W., Rathgeber, F., Bercea, G.-T. 和 Markall, G. R.：《Firedrake用户手册》，伦敦帝国理工学院、牛津大学、贝勒大学和华盛顿大学，第一版，https://doi.org/10.25561/104839，2023年。Hossain, M. A., Pimentel, S. 和 Stockie, J. M.：使用结合数据同化的水平集方法模拟海洋出口冰川的地表高度和冰川末端位置，《计算物理学杂志》，474，111766，https://doi.org/https://doi.org/10.1016/j.jcp.2022.111766，2023年。Kyanjo, B. 和 Robel, A.：ICESEE：用于耦合冰盖模型的先进数据同化软件包，https://github.com /ICESEE-project/ICESEE，2025a。Kyanjo, B. 和 Robel, A.：ICESEE：用于耦合冰盖模型的先进数据同化软件包，https://github.com / ICESEE-project/ICESEE/wiki，2025b。Kyanjo, B. 和 Robel, A.：ICESEE（v1.0.0）：冰盖状态与参数估计器，https://doi.org/10.5281/zenodo.18716132，2026年。Larour, E., Seroussi, H., Morlighem, M. 和 Rignot, E.：使用冰盖系统模型开展大陆尺度、高阶、高空间分辨率的冰盖建模，《地球物理研究杂志：地球表面》，117，F01022，https://doi.org/10.1029/2011JF002140，2012年。Lorenz, E.：可预测性：一个部分解决的问题，1995年。MacAyeal, D. R.：黏性基底沉积物上的大尺度冰流：理论及其在南极洲B冰流中的应用，《地球物理研究杂志：固体地球》，94，4071–4087，1989年。Morlighem, M., Seroussi, H., Larour, E. 和 Rignot, E.：利用高阶模型的精确与不完整伴随反演南极洲的基底摩擦，《地球物理研究杂志：地球表面》，118，1746–1753，https://doi.org/https://doi.org/10.1002/jgrf.20125，2013年。Müller, S., Schüler, L., Zech, A. 和 Heße, F.：GSTools v1.3：用于Python地统计建模的工具箱，《地球科学模型开发》，15，3161–3182，https://doi.org/10.5194/gmd-15-3161-2022，2022年。

Nowicki, S. M. J., Payne, A., Larour, E., Seroussi, H., Goelzer, H., Lipscomb, W., Gregory, J., Abe-Ouchi, A. 和 Shepherd, A.：冰盖模式比较计划（ISMIP6）对CMIP6的贡献，《地球科学模型开发》，9，4521–4545，https://doi.org/10.5194/gmd-9-4521-2016，2016年。PACE：高级计算环境合作项目（PACE），http://www.pace.gatech.edu，2017。Robel, A.：SSAsimpleM：发布供发表的SSAsimpleM，https://doi.org/10.5281/zenodo.5245271，2021年。Ruckstuhl, Y. M. 和 Janji´c, T.：用于对流尺度应用的集合卡尔曼滤波器算法参数与状态估计，《英国皇家气象学会季刊》，144，826–841，https://doi.org/https://doi.org/10.1002/qj.3257，2018年。Ruiz, J. 和 Pulido, M.：存在模型误差时使用基于集合的数据同化进行参数估计，《月刊天气评论》，143，1568–1582，https://doi.org/10.1175/MWR-D-14-00017.1，2015年。Sakov, P. 和 Oke, P. R.：集合卡尔曼滤波器的确定性形式：集合平方根滤波器的替代方案，《Tellus A：动力气象学与海洋学》，60，361–371，https://doi.org/10.1111/j.1600-0870.2007.00299.x，2008年。Shapero, D. R., Badgeley, J. A., Hoffman, A. O. 和 Joughin, I. R.：icepack：一个新的Python冰川流动建模软件包，1.0版，《地球科学模型开发》，14，4593–4616，https://doi.org/10.5194/gmd-14-4593-2021，2021年。Shepherd, A., Ivins, E., Rignot, E., Smith, B., van den Broeke, M., Velicogna, I., Whitehouse, P., Briggs, K., Joughin, I., Krinner, G., Nowicki, S., Payne, T., Scambos, T., Schlegel, N., A, G., Agosta, C., Ahlstrøm, A., Babonis, G., Barletta, V., Blazquez, A., Bonin, J., Csatho, B., Cullather, R., Felikson, D., Fettweis, X., Forsberg, R., Gallee, H., Gardner, A., Gilbert, L., Groh, A., Gunter, B., Hanna, E., Harig, C., Helm, V., Horvath, A., Horwath, M., Khan, S., Kjeldsen, K. K., Konrad, H., Langen, P., Lecavalier, B., Loomis, B., Luthcke, S., McMillan, M., Melini, D., Mernild, S., Mohajerani, Y., Moore, P., Mouginot, J., Moyano, G., Muir, A., Nagler, T., Nield, G., Nilsson, J., Noel, B., Otosaka, I., Pattle, M. E., Peltier, W. R., Pie, N., Rietbroek, R., Rott, H., Sandberg-Sørensen, L., Sasgen, I., Save, H., Scheuchl, B., Schrama, E., Schröder, L., Seo, K.-W., Simonsen, S., Slater, T., Spada, G., Sutterley, T., Talpe, M., Tarasov, L., van de Berg, W. J., van der Wal, W., van Wessem, M., Vishwakarma, B. D., Wiese, D., Wouters, B. 和 team, T. I.：1992年至2017年南极冰盖的质量平衡，《自然》，558，219–222，https://doi.org/10.1038/s41586-018-0179-y，2018年。Sueki, K., Nishizawa, S., Yamaura, T. 和 Tomita, H.：基于集合卡尔曼滤波器的参数估计的精度与收敛速度：为可靠且高效的估计设定参数不确定性，《地球与行星科学进展》，9，47，https://doi.org/10.1186/s40645-022-00504-4，2022年。Tippett, M. K., Anderson, J. L., Bishop, C. H., Hamill, T. M. 和 Whitaker, J. S.：集合平方根滤波器，《月刊天气评论》，131，1485–1490，https://doi.org/10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2，2003年。Wang, X. 和 Bishop, C. H.：繁殖法与集合变换卡尔曼滤波器集合预报方案的比较，《大气科学杂志》，60，1140–1158，https://doi.org/10.1175/1520-0469(2003)060<1140:ACOBAE>2.0.CO;2，2003年。Wang, X., Bishop, C. H. 和 Julier, S. J.：哪种方式更好：正负配对集合还是中心化球面单纯形集合？《月刊天气评论》，132，1590–1605，https://doi.org/10.1175/1520-0493(2004)132<1590:WIBAEO>2.0.CO;2，2004年。Whitaker, J. S. 和 Hamill, T.M.：不使用扰动观测的集合数据同化，《月刊天气评论》，130，1913–1924，https://doi.org/10.1175/1520-0493(2002)130<1913:EDAWPO>2.0.CO;2，2002年。