arXiv:2602.07771v1 [physics.comp-ph] 8 Feb 2026

用于蒙特卡罗模拟内存缩减的压缩感知

Ethan Lame、Camille J. Palmer、Todd Palmer、Ilham Variansyah
《核科学与工程》—俄勒冈州立大学，科瓦利斯，俄勒冈州

文章历史
编制于2026年2月10日

摘要
中子学系统的蒙特卡罗模拟计算量大，高保真建模需要大量内存资源。压缩感知能够以远少于传统方法的样本，准确重建信号。本文研究的压缩感知具体实现方式是使用重叠单元收集计数。增加样本数量可提高重建精度，但随着样本增多，边际收益会逐渐降低。基追踪去噪中使用的稀疏度参数会显著影响重建质量。在所考察的三个测试案例中，二维重建的内存减少最多达81.25%，三维重建最多达96.25%；在部分场景中，重建误差控制在相应高保真参考结果的1个标准差以内。

关键词
蒙特卡罗中子输运；压缩感知

1. 引言

蒙特卡罗中子输运模拟要达到较高精度，计算量和耗时可能非常大。减少所需内存量和/或计算时间的方法值得关注。

压缩感知是一种方法，旨在使用远少于传统所需数量的测量值，准确重建信号（Cand`es、Romberg 和 Tao，2005），其基础是两项关键原则：待重建的信号必须在某个基底下是稀疏的，并且测量值与新基底之间必须具有不相干性（Donoho，2006）。离散余弦变换能够很好地将自然图像表示为稀疏形式，而对信号进行随机采样可确保满足不相干性约束（Cand`es 和 Romberg，2007）。

压缩感知此前已成功应用于蒙特卡罗中子输运（Vaquer 等，2016；Madsen，2017）。其中一种方法采用随机不相交计数，将计数区域划分为非连续的空间单元，并随机分布于整个问题域中，从而降低覆盖整个问题域所需的内存。相比之下，本文提出了一种基于笛卡尔空间中重叠计数的压缩感知方法。与规则网格计数和随机不相交计数均不同，这些重叠

通讯作者 - Ethan Lame。电子邮箱：lamee@oregonstate.edu

区域既能进行局部测量，又能为重建全局通量测量值提供充分覆盖。本文的实现只需利用这些重叠区域构建一个基底，即可覆盖问题空间的绝大部分。粗单元重叠后能够分辨小尺度细节，因为重叠区域内的任何内容都会同时计入两个粗单元，其影响便可定位到两个重叠单元共同包含的小区域内。当考虑多个变量（空间、能量、角度、时间）时，蒙特卡罗模拟的计数数量会迅速增加。通过压缩感知，我们旨在从模拟中获得中子径迹长度通量计数，而无需承担大量计数所需的内存，同时尽可能保持相近的统计不确定度水平。

压缩感知算法在蒙特卡罗动态代码（MC/DC）（Morgan 等，2024）中实现，该代码由百亿亿次级蒙特卡罗中子输运中心（CEMeNT）开发。MC/DC 是一款高性能、可扩展的软件包，用于开发和测试中子输运应用的新型算法。MC/DC 使用 Python 编写，也采用 Numba 编译器来提升性能，并能够在 CPU 和 GPU 上运行，同时通过使用 mpi4py 支持并行计算（Dalcín 等，2005）。

1.1. 压缩感知

给定一个信号 $\mathbf{f}$，可以对$\mathbf{f}$ 进行变换，得到稀疏信号$\mathbf{x}$ ——即$\mathbf{f}$ 在稀疏基中的表示。我们还可以对$\mathbf{f}$ 进行随机采样，得到信号的测量值$\mathbf{b}$。假设变换操作是通过某个矩阵$\mathbf{T}$实现的，而采样是通过某个矩阵$\mathbf{S}$ 完成的，则可以写为：

$$\boldsymbol{S}\boldsymbol{f} = \boldsymbol{b} \tag{1}$$
$$\boldsymbol{S}\boldsymbol{T}^{-1}\boldsymbol{T}\boldsymbol{f} = \boldsymbol{b} \tag{2}$$
$$\boldsymbol{T}\boldsymbol{f} = \boldsymbol{x} \tag{3}$$
$$\boldsymbol{A} = \boldsymbol{S}\boldsymbol{T}^{-1} \tag{4}$$
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{5}$$

感知矩阵 $\mathbf{A}$ 对采样和逆变换进行编码，而 $\mathbf{x}$ 是感兴趣信号的稀疏表示。我们使用这个最终方程来寻找一个稀疏信号 $\mathbf{x}$，使得当 $\mathbf{x}$ 经过逆变换并采样（即应用$\mathbf{A}$ 时），结果与测量值 $\mathbf{b}$相符。在搜索之前，关于该问题唯一已知的信息是搜索所用的基、采样方法 $\mathbf{S}$以及信号的测量值$\mathbf{b}$。

1.1.1. 优化

在本研究中，我们选择离散余弦变换（DCT）作为变换$\mathbf{T}$ ，并令$\mathbf{T}^{-1}$ 为逆离散余弦变换（IDCT）。自然图像在 DCT 基中通常是稀疏的（Cand`es 和 Romberg，2007），这对压缩感知方法的有效运行至关重要。

要使用公式 5 在离散余弦变换域中进行搜索，我们需要对问题施加一些优化约束。可以采用多种优化方法，在该域中搜索与信号测量值相匹配的向量。

压缩感知可以采用基追踪（Donoho 2006），即选择一个稀疏向量，使其与测量值之间的差异最小化。基追踪优化的一种改进方法也常被采用，称为基追踪去噪（Chen、Donoho 和 Saunders 2001）。当测量值可能含有噪声时，这种方法尤其有用，因为它能在匹配含噪测量值与衡量信号稀疏程度之间取得平衡。我们选择采用基追踪去噪，其形式如下：

$$\min_x \left(\frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \right). \tag{6}$$

优化约束中的第一项是准确性项。如果 $\mathbf{Ax} = \mathbf{b}$，那么向量 $\mathbf{x}$ 就与**f**的稀疏表示完全匹配，因此优化过程可能会选择该**x**值，具体取决于第二项的权重。第二项控制**x**稀疏性的重要程度。在蒙特卡罗模拟中，我们通常不希望与含噪测量值完全匹配。相反，我们注意到真实信号在该基底下是稀疏的，因此也会强调**x**的稀疏性。λ 的值越高，越优先考虑稀疏性而非与测量值的匹配； λ的值越低，越优先考虑与测量值的匹配而非稀疏性。

2. 方法

2.1. 构造矩阵

2.1.1. 采样矩阵 S

采样矩阵**S** 生成感兴趣信号的随机样本。蒙特卡罗代码中最常用的计数结构是高分辨率结构化笛卡尔网格（Lepp¨anen 等，2025；Kulesza 等，2024；Romano 等，2015）。我们改用重叠单元来存储计数。这样可以用比高分辨率笛卡尔网格更少的网格单元覆盖整个问题区域，从而减少与计数相关的内存开销，同时仍可通过包含在多个单元中的小区域捕捉问题的小尺度特征。

通过简单的 Halton 序列，将粗单元的中心点布置在问题空间中。Halton 序列是一种低差异伪随机方法，可确保问题空间的覆盖比其他“真正的”随机采样技术更加均匀（Owen 2017）。网格单元的大小在执行问题之前确定。建议设定的大小应使所选数量的网格单元能够覆盖整个问题空间，同时确保网格单元足够小，以便解析精细细节。要求覆盖整个问题空间或许也是一项不错的约束，但本研究并未采用这一做法。

图 1 展示了这些重叠单元的一个示例。采样矩阵 **S** 按以下方式构建：将粗分箱叠加在一定尺寸的笛卡尔网格上。对于每个粗分箱，记录每个标准笛卡尔单元位于该粗分箱内的面积比例，并将所有单元值记录为一个行向量，其元素数量与标准单元的数量相同。对每个粗分箱重复此过程，再将各行上下堆叠，形成采样矩阵；该矩阵的行数等于粗分箱的数量，列数等于标准单元的数量。在对压缩感知数据进行后处理时，将该矩阵与某个信号的展平版本相乘，

即可将其数据有效地重新分配到粗分箱中。对于压缩感知的原位 实现，此过程仅用于获得采样矩阵，并将其用于构建感知矩阵，相关内容将在后文介绍。

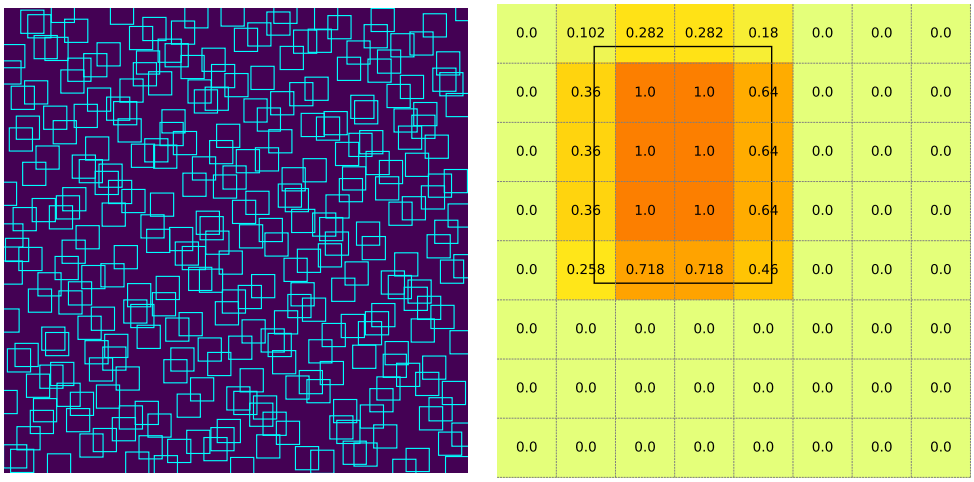


图 1. （左）二维中重叠单元的示例，覆盖了问题空间的绝大部分。（右）叠加在笛卡尔网格上的单个粗分箱，并显示了每个网格单元的重新分配值。将重新分配值矩阵展平后，即为采样矩阵 **S** 的一行。

2.1.2. 测量向量**b**

系统的测量值以每个粗分箱的径迹长度计数形式收集。粒子流经空间时，会在一个或多个粗分箱内描绘出其路径。分箱的计数得分会根据粒子路径位于该分箱内的长度成比例增加。如果粒子的路径穿过分箱间的重叠区域，则单个粒子可以计入多个分箱。

2.1.3. 感知矩阵**A**

感知矩阵 **A** 对离散余弦变换域中的采样及变换进行了编码。我们在公式 4 中确立了**A**可由两个矩阵的乘积构造，其中$\mathbf{T^{-1}}$ 为逆离散余弦变换（IDCT）。对于高分辨率重建，方阵$\mathbf{T^{-1}}$ 的构造会变得极其庞大。因此，我们不显式构造**S** 并将其与$\mathbf{T^{-1}}$相乘，而是使用 SciPy 的 spfft.dctn 函数直接执行相应的离散余弦变换；该函数无需构造矩阵即可高效完成相同的运算。

$$\mathbf{A} = \mathbf{S}\mathbf{T^{-1}} \Rightarrow \mathbf{A}^T = (\mathbf{S}\mathbf{T^{-1}})^T = (\mathbf{T^{-1}})^T\mathbf{S}^T = \mathbf{T}\mathbf{S}^T \tag{7}$$

$$\mathbf{A} = (\mathbf{T}\mathbf{S}^T)^T, \tag{8}$$

其中，上标 T 表示转置。IDCT 和 DCT 矩阵是正交的，因此 $(\mathbf{T^{-1}})^T = \mathbf{T}$。在方程 8 中，则通过对矩阵 **T** 应用 spfft.dctn 来完成与变换矩阵的乘法。

2.2. 重建

测量向量 $\mathbf{b}$ 由模拟得到，此时粒子被累计到粗分箱中。随后，选择重建所需的分辨率，并构造感知矩阵。使用 Python 的 CVXPY 软件包执行基追踪去噪优化（ Diamond 和 Boyd，2016），得到稀疏向量$\mathbf{x}$ ，再对其进行逆变换，得到信号$\mathbf{f}$。

重建误差量化为重建结果（$\mathbf{r}$）与参考解（$\mathbf{f}$）之间的相对 ℓ^2 误差：

$$\text{error} = \frac{\|\mathbf{f} - \mathbf{r}\|_2}{\|\mathbf{f}\|_2}. \tag{9}$$

该重建误差与参考解的全局标准差进行比较，
ref以此量化重建的准确性。

3. 结果

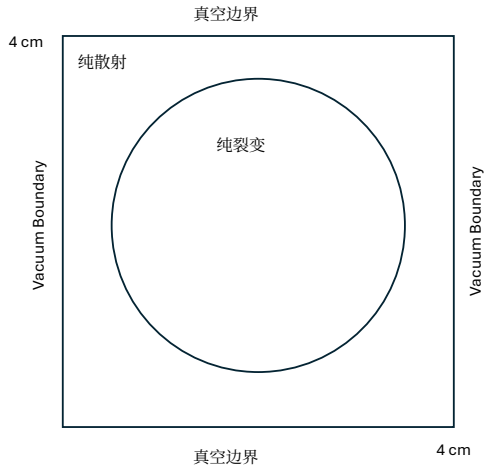
我们使用 MC/DC 模拟了三个测试问题，以评估基追踪去噪重建的性能。第一个问题是一个非常简单的纯裂变球体，位于纯散射立方体内部。这种简单几何结构便于对代码进行测试和故障排查。几何结构和材料性质见图 2 和表 1。该问题采用默认网格分辨率，在二维中包含40×40 个网格单元，在三维中包含 $20\times20\times20$ 个网格单元。网格单元均匀分布，二维覆盖范围为 4×4 厘米，三维覆盖范围为 $4\times4\times4$ 厘米。我们通过比较所用粗分箱数量与计算参考解所用网格单元数量，来量化内存缩减。

第二个问题基于小林狗腿基准问题（Kobayashi 等人，2001）。几何结构和材料性质如图 3 和表 2 所示。该问题默认采用的网格分辨率为二维 30×50 个网格单元、三维 $15\times25\times15$个网格单元。网格单元均匀分布，覆盖二维中 60×100 厘米、三维中 $60\times100\times60$厘米的空间。

第三个问题是 Cooper 和 Larsen（2001）所提出问题的组合。它包含一个强散射屏障区域和一个真空管道区域，因此我们可以观察压缩感知能否重建粒子通量很少甚至为零的区域。其几何结构和材料性质如图 4 和表 3 所示。该问题默认使用 40×40 个网格单元的网格分辨率，且仅为二维问题。网格单元均匀分布，覆盖 4×4 厘米的空间。三个问题的每次模拟均使用 10^4个粒子历史数。

区域	Σ_t	Σ_s	Σ_f	ν
球体	1.0	0	1.0	1.1
球体外部	1.0	1.0	0	N/A

表 1. 纯裂变球体问题的材料性质。截面单位为厘米$^{-1}$。粒子在整个问题中均匀生成。



Fi图 2. 纯裂变球体问题的几何结构。球体半径为 1.5 厘米，其中心位于 at 边长为 4 厘米的立方体的中心。

区域	源（n 厘米$^{-3}$ s^{-1}）	Σ_t （厘米$^{-1}$）	$c = \Sigma_s/\Sigma_t$
1	1	0.1	0.5
2	0	10^{-4}	0.5
3	0	0.1	0.5

表 2. Kobayashi 问题的材料性质。截面单位为厘米$^{-1}$。粒子在整个问题区域内均匀产生。

区域	Σ_t	$c = \Sigma_s/\Sigma_t$
屏障	5	0.8
Duct	0.01	0.8
其他位置	1	0.8

表 3。 修正 Cooper-Larsen 问题的材料性质。截面的单位为厘米$^{-1}$。粒子在整个问题区域内均匀生成。

3.1. 使用不同值进行球体重建 λ

3.1.1. 二维结果

二维球体问题的重建结果见图5，统计相对误差见图6。当 λ取较大值时，可以看到重建结果沿 x 轴和 y 轴均呈对称；但当 λ取其他值时，重建结果似乎呈径向对称，与参考解相符。不同 λ 的主要区别似乎在于球体边缘的锐利程度和背景通量的大小。较小的 λ 会产生非零背景通量，而较大的 λ 则会使球体与背景之间的边界变得模糊——中间某个值最适合基追踪去噪。选择 λ时覆盖了较大的取值范围，而基追踪去噪通常使用 0.5。使用 $\lambda = 0.5$ 进行的几项初步测试效果不佳，因此这些重建采用了较小的取值。

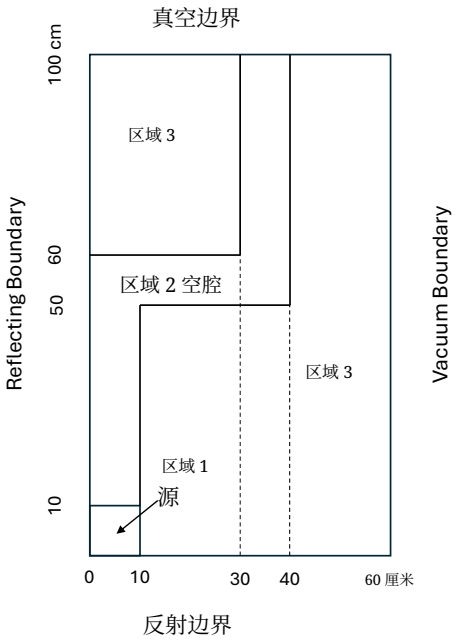


图 3. Kobayashi狗腿基准问题在 x-y 平面上的几何结构。

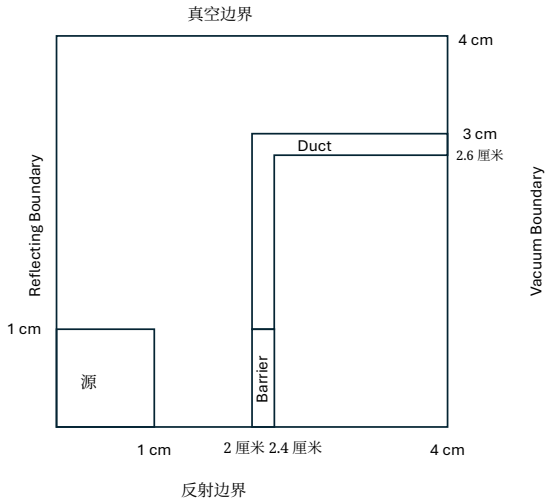


图 4. 修正 Cooper-Larsen 问题的几何结构。屏障高 1 厘米，与源的边缘齐平。

3.1.2. 三维结果

三维球体问题的重建结果如图7、图8和图9所示，图10展示了统计相对误差。

3.2. *Kobayashi* 问题在不同 λ 值下的重建

3.2.1. 二维重建中的统计相对误差

Kobayashi 问题重建的统计误差如图 11 所示。

3.2.2. 三维重建中的统计相对误差

Kobayashi 问题重建的统计误差如图 12 所示。

3.3. 修正 *Cooper-Larsen* 问题在不同 λ 值下的重建

3.3.1. 二维重建中的统计相对误差

修正 Cooper-Larsen 问题重建的统计误差如图 13 所示。

3.4. 运行时间

随着问题中使用的粗分箱数量增加，每个粒子都需要执行更多计算，从而延长总模拟时间。MC/DC 可以使用 Numba（Python 的即时编译器），在我们的案例中实现数量级的加速。图 14 展示了使用和不使用 Numba 时的模拟运行时间差异。

用于计数的粗分箱数量也会改变基追踪去噪算法在稀疏基中寻找解所需的时间。粗分箱越多，需要匹配的测量值就越多，因此算法需要搜索更大的空间。重建运行时间随粗分箱数量变化的关系如图 15 所示。

4. 讨论

这种压缩感知方法——使用重叠粗分箱收集计数，并使用基追踪去噪寻找重建——能够重建给定的测试案例，并且在简单几何情形下表现最佳。这很可能是因为简单几何在 DCT 基中被表示为比其他几何结构更稀疏的向量。

结合图5和图6可以看出，在一个简单的二维情形中，即使内存占用减少超过80%，我们仍能将误差控制在 MC/DC 解的标准差以内。此外，在同一情形中增加分箱数量，还可以进一步降低误差。其他情形的表现不那么理想，但仍呈现出分箱数量增加、误差减小的趋势。

三维球体重建（图10）显示，内存缩减最高可达96.25%，相应误差在参考解均值的 2σ 以内。然而，要将误差控制在参考解均值的 1σ 以内，就必须限制内存缩减幅度。没有任何三维重建能将误差控制在参考解均值的 1σ 以内。增加分箱数量或许最终能将误差降至 1σ 阈值以下，但这可能需要数量大到难以承受的测量值。例如，球体情形中内存缩减80%（1600个分箱）确实比内存缩减87.5%（1000个分箱）的情形误差更小，如图10所示。

随着所用粗分箱数量的增加，重建的精度随之提高。然而，模拟中执行的计算次数也会增加。图 14 表明，在 Numba模式下，模拟运行时间并不随粗单元数量而变化。粗分箱数量较少时确实需要更多计算时间，因为 Numba 在运行时编译代码需要一定开销。在包含高散射截面区域的修正 Cooper-Larsen 问题中，当分箱数量很大时，Numba 可提供一个数量级的加速。图 14 中的趋势还表明，随着分箱数量越来越大，来自 Numba 的编译开销变得不那么重要，而其加速能力则变得更加有用。

粗分箱数量的增加也会使基追踪去噪优化更加困难，因为需要考虑更多测量值。优化可探索的自由度更多，因此随着测量值数量增加，在 DCT 基中的搜索需要更多计算时间。如图 15 所示，随着粗分箱数量增加，用基追踪去噪寻找解的计算时间似乎呈二次增长。

5. 结论

在本研究中，我们提出了一种将压缩感知应用于蒙特卡罗中子输运的方法，通过使用大型、相互重叠的计数分箱覆盖问题空间，以少于常规蒙特卡罗计数所需数量的计数来完成覆盖。该方法利用了这样一个事实：中子通量计数的全局解——本研究所关注的量——可以在离散余弦变换基中表示为稀疏形式。通过寻找与粗分箱中的计数相匹配的稀疏向量，我们可以获得全局中子通量的近似值，同时实现

大幅内存减少，在某些情况下最高可达 96.5%。在内存减少 37.5% 的情况下，可以达到与参考解相当的误差。

我们研究了重建的统计相对误差如何随稀疏度参数 λ变化。我们发现，存在一个 λ 值可获得最佳重建（即相对误差最小的重建），但对于不同的问题，以及同一问题的不同分辨率，该值各不相同。

在此处展示的14个案例中，有2个案例成功重建了中子标量通量，其精度达到了参考解标准差的 1σ 以内，同时显著减少了内存（37.5%和81.25%）。然而，只有在最简单的几何结构中才观察到这种精度。对于几何结构更复杂的问题，重建解的误差可控制在参考解标准差的一个数量级以内。这些案例也实现了大幅内存缩减（$> 80\%$）。

在蒙特卡罗中子输运应用中，使用粗粒度计数箱来减少内存颇具前景。将粗粒度分箱重建方法扩展到其他维度（能量、角度、时间），可能会进一步降低内存需求，不过重建时间可能会限制其适用性。基追踪去噪并非因效率高而被选用，而且我们没有进行并行化，因此，高分辨率重建所需时间的增加可能成为限制因素。

人们也关注用于搜索解决方案稀疏基的其他算法，因为寻找重建所需的计算时间似乎会随着所使用粗分箱数量的平方增长。还有必要对所使用的分箱数量以及重建准确性与效率之间的权衡进行更定量的研究，以评估准确性方面的提升是否足以抵消额外的计算成本。

与高保真度（统计误差极低且计数网格分辨率很高）的结果进行比较将大有裨益；随着我们的软件实现能够扩展到更大的高性能计算系统，这类比较可能会在未来的出版物中纳入。

利益冲突声明

作者声明不存在需要披露的竞争性利益。

资助

本研究得到超大规模蒙特卡罗中子输运中心（CEMeNT）的支持。该中心是由美国能源部通过资助编号 DE-NA003967 资助的 PSAAP-III 项目。

作者贡献

C. Palmer、T. Palmer 和 I. Variansyah 提出了本研究的构想，协助开发方法，并对本文提出了反馈意见。E. Lame 实现了这些方法，完成了所有模拟和分析，并撰写了本文。

参考文献

Cand`es, Emmanuel, Justin Romberg, and Terence Tao. 2005. “从不完整且不准确的测量值中稳定恢复信号。” *arXiv* 预印本math/0503066. https://arxiv.org/abs/math/0503066. Donoho, David. 2006. “压缩感知。”*IEEE*信息论汇刊52 (5): 1289–1306. https://doi.org/10.1109/TIT.2006.871582. Cand`es, Emmanuel, and Justin Romberg. 2007. “压缩采样中的稀疏性与不相干性。”《逆问题》23 (3): 969–985. https://doi.org/10.1088/0266-5611/23/3/008. Vaquer, Pablo A., et al. 2016. “使用随机不相交计数的蒙特卡罗输运模拟压缩感知框架。”《计算与理论输运杂志》45 (3): 219–229. https://doi.org/10.1080/23324309.2016.1156550. Madsen, Jonathan Robert. 2017. “不相交计数方法：一种利用压缩感知降低统计噪声和内存的蒙特卡罗计分方法。”博士论文，德克萨斯农工大学。Chen, Scott Shaobing, David L. Donoho, and Michael A. Saunders. 2001. “通过基追踪进行原子分解。”《*SIAM*评论》43 (1): 129–159. https://doi.org/10.1137/S003614450037906X. Morgan, Joanna, et al. 2024. “蒙特卡罗/动态代码 (MC/DC)：用于全瞬态中子输运和快速方法开发的加速Python软件包。”《开放源代码软件杂志》9 (94): 6415. https://doi.org/10.21105/joss.06415. Dalc´ın, Lisandro, et al. 2005. “Python的MPI。”《并行与分布式计算杂志》65 (9): 1108–1115. https://doi.org/10.1016/j.jpdc.2005.03.010. Kobayashi, Keisuke, et al. 2001. “简单几何中含真空区域的三维辐射输运基准问题及结果。”《核能进展》39 (2): 119–144. https://doi.org/10.1016/S0149-1970(01)00007-5. Cooper, Marc A., and Edward W. Larsen. 2001. “用于全局蒙特卡罗粒子输运计算的自动权窗。”《核科学与工程》137 (1): 1–13. https://doi.org/10.13182/NSE00-34. Owen, Art B. 2017. “R中的随机化Halton算法。” *arXiv* 预印本arXiv:1706.02808. https://arxiv.org/abs/1706.02808. Diamond, Steven, and Stephen Boyd. 2016. “CVXPY：一种嵌入Python的凸优化建模语言。”《机器学习研究杂志》17 (83): 1–5. Lepp¨anen, Jaakko, Ville Valtavirta, Antti Rintala, and Riku Tuominen. 2025. “2024年Serpent蒙特卡罗代码的状况。” *EPJ*核科学与技术11: 3. https://doi.org/10.1051/epjn/2024031. Romano, Paul K., Nicholas E. Horelik, Bryan R. Herman, Adam G. Nelson, Benoit Forget, and Kord Smith. 2015. “OpenMC：用于研究与开发的先进蒙特卡罗代码。”《核能年鉴》82: 90–97. https://doi.org/10.1016/j.anucene.2014.07.048. Kulesza, Joel A., et al. 2024. $MCNP^{®}$ 代码版本 *6.3.1* 理论与用户手册。LA-UR-24-24602，修订版1。洛斯阿拉莫斯国家实验室。https://doi.org/10.2172/2372634.

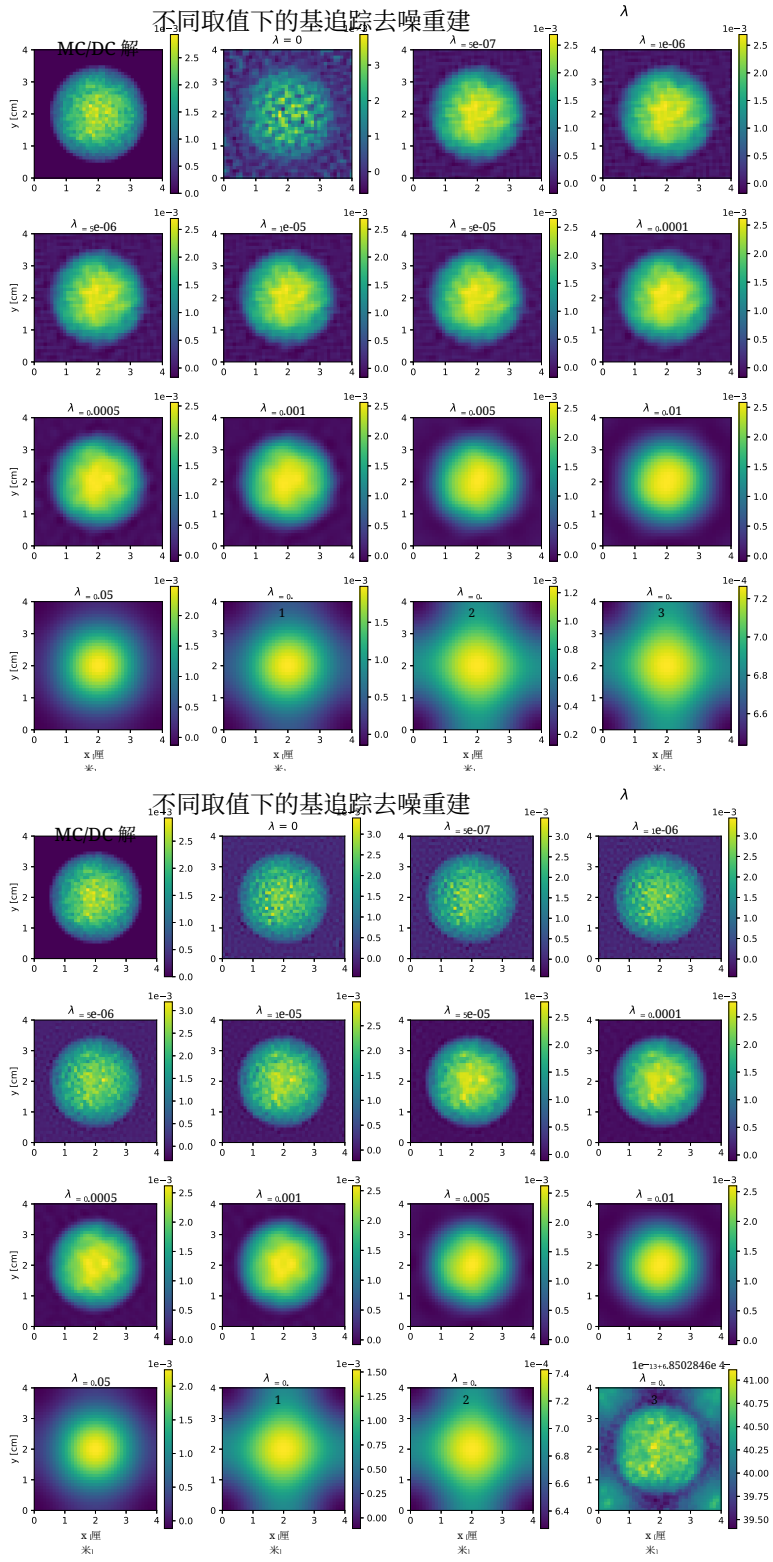
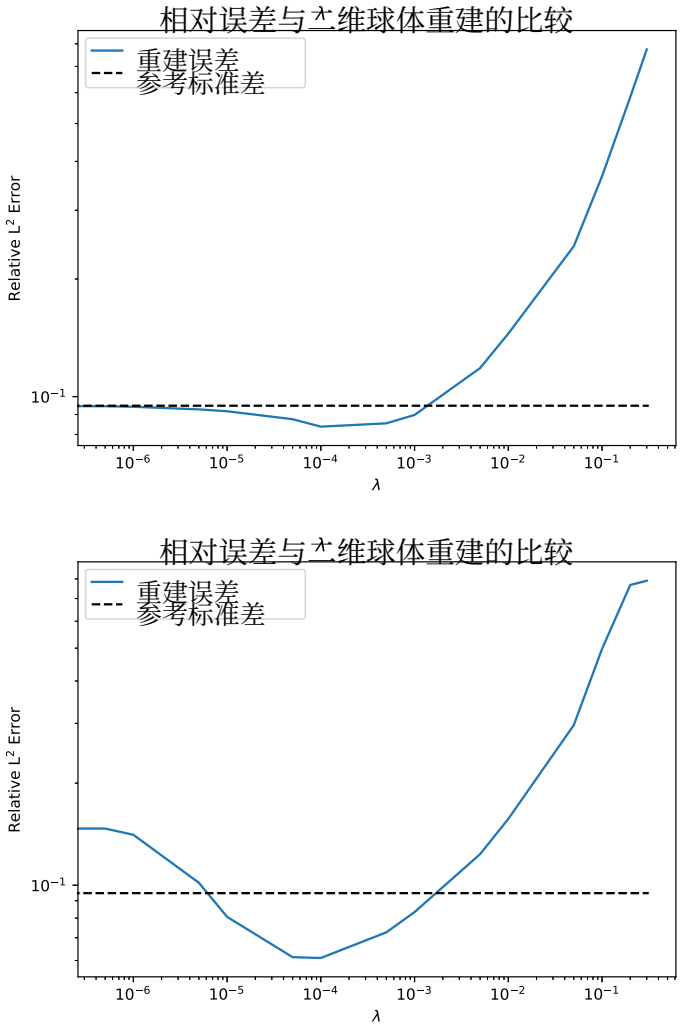


图5. 使用基追踪去噪对二维球体问题进行不同取值下的重建。左侧为300个大小为 3×3 像素的分箱；右侧为1000个大小为 2×2 像素的分箱。每组左上角显示参考解，使用 10^4 粒子运行。每种重建对应的误差绘于图6。



Fi图6。图5中二维球体各次重建对应的误差。上： 300 3×3分箱，
81.内存缩减25%。下： 1000 2×2 分箱，内存缩减37.5%。

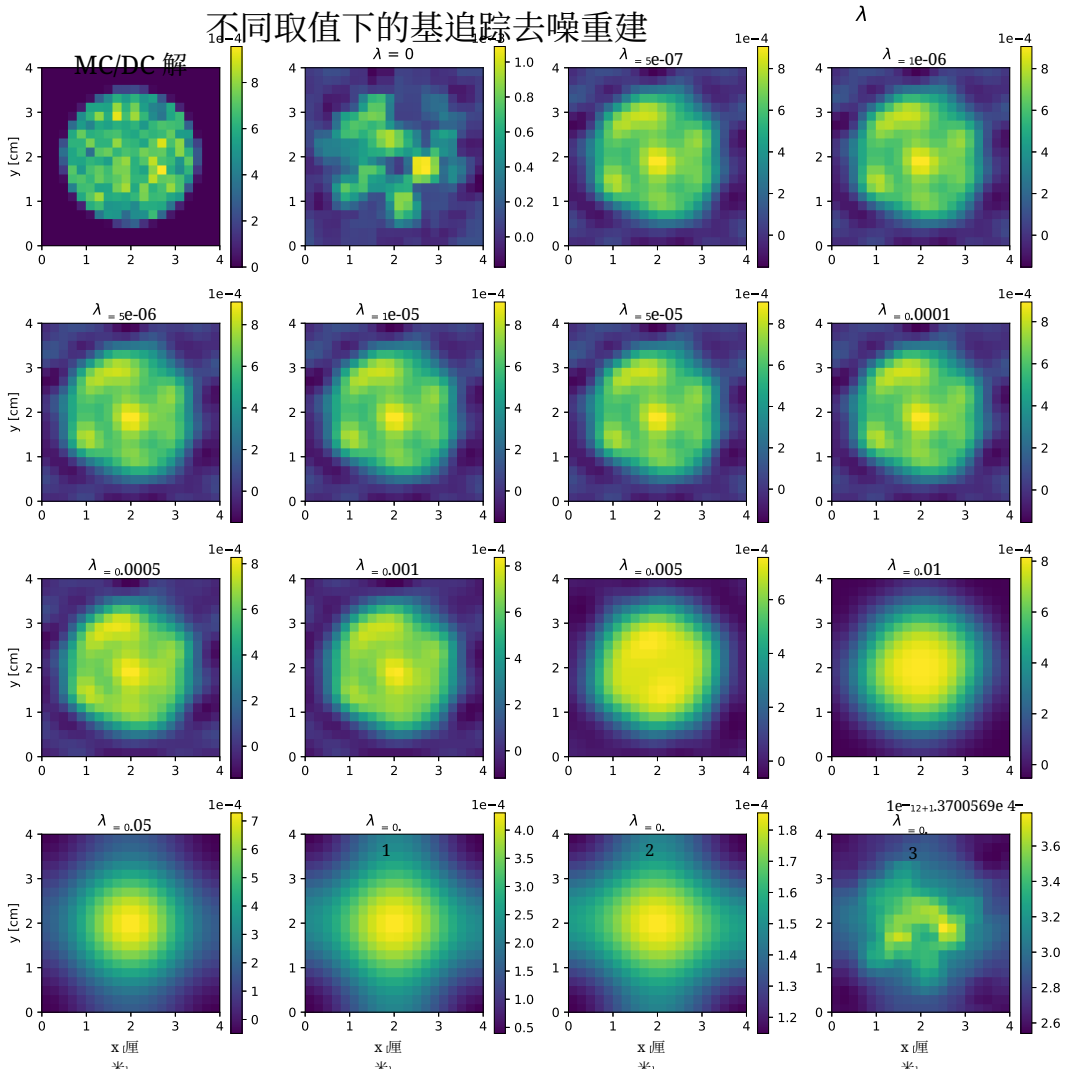


图7. 使用基追踪去噪得到的球体三维重建切片。结果采用300个分箱。所有分箱的大小均为 $3\times3\times3$ 像素。这些重建的误差绘于图10中。

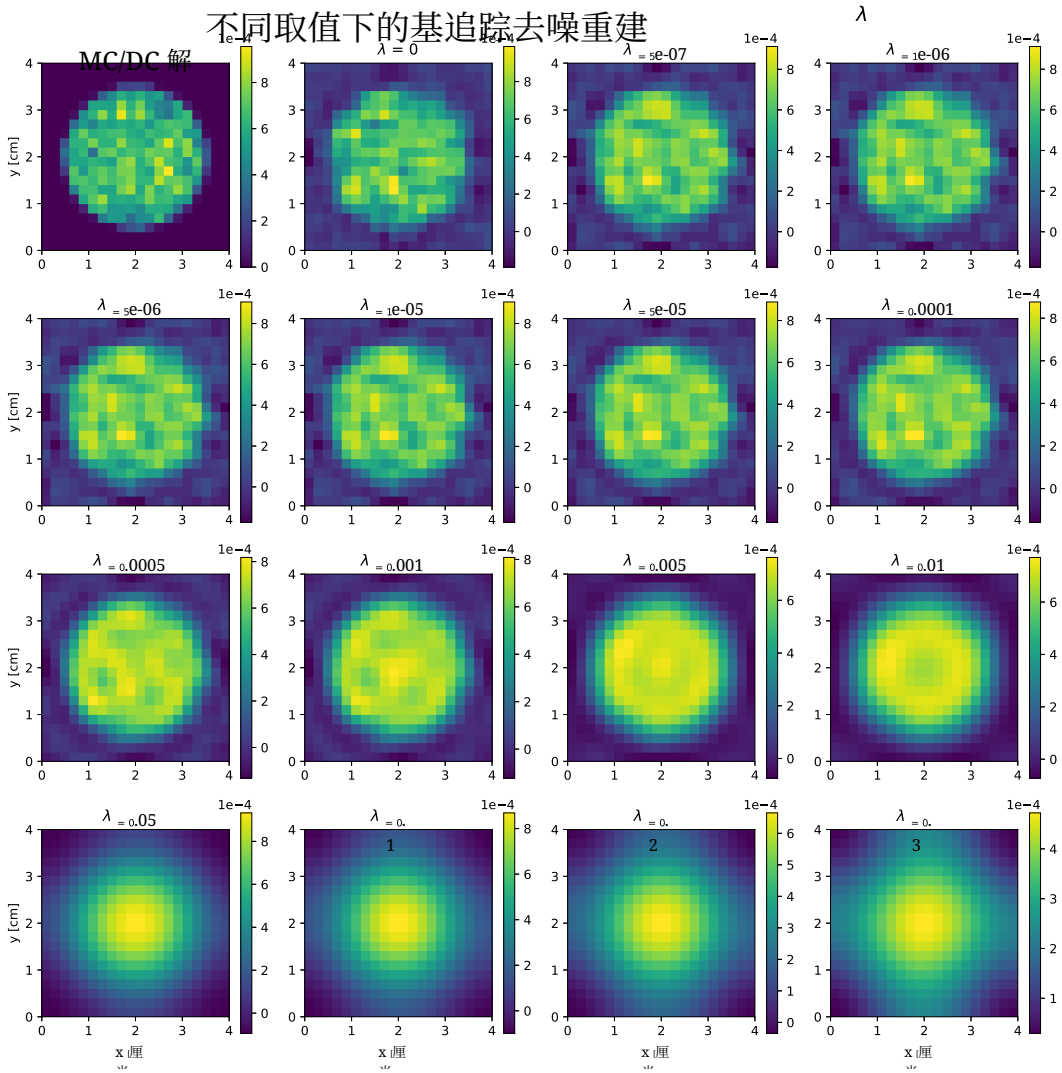


图8。 使用基追踪去噪得到的球体三维重建切片。结果采用1000个分箱。所有分箱的尺寸均为 3×3×3像素。这些重建的误差绘制在图10中。

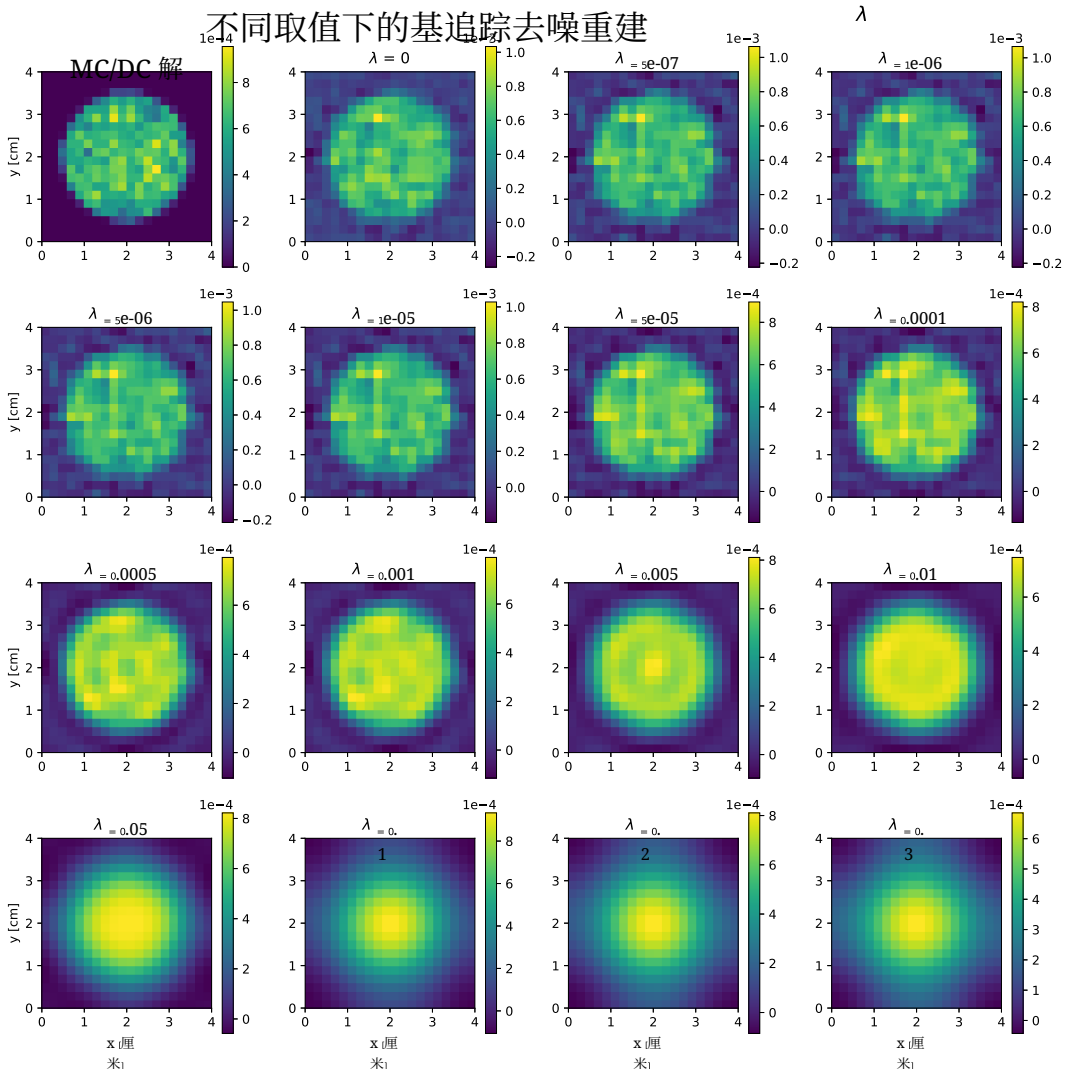
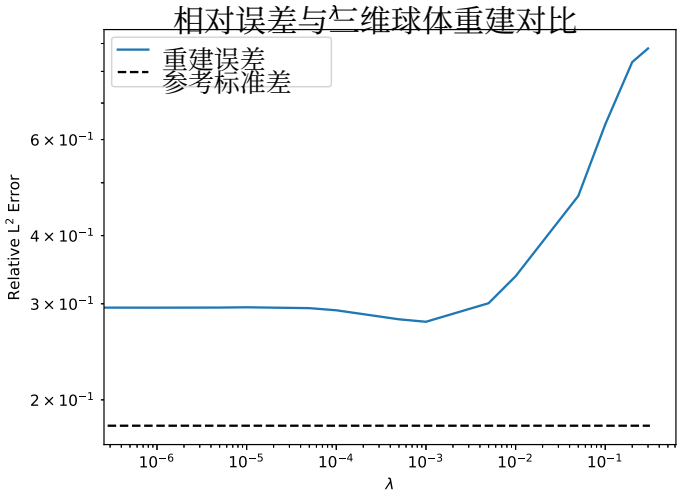
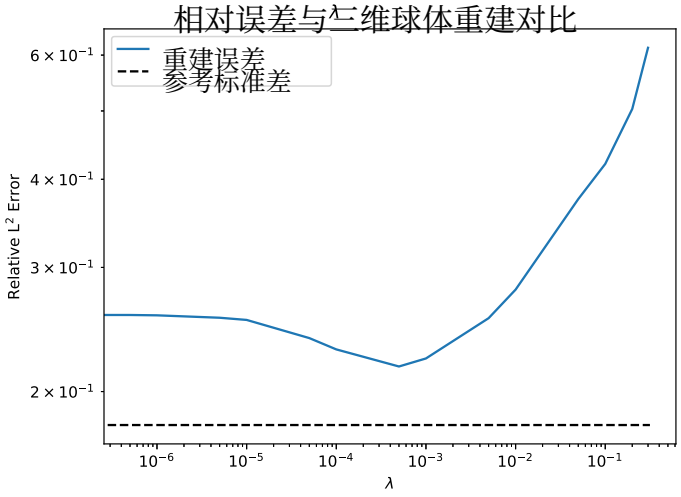


图9. 使用基追踪去噪对球体进行三维重建后的切片。结果采用1600个分箱。所有分箱的大小均为 3×3×3像素。与这些重建相关的误差绘制在图10中。





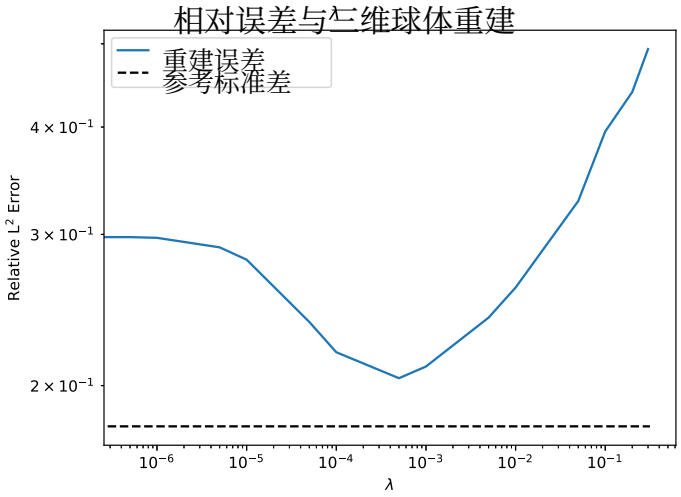


图10。 图7、图8和图9所示球体重建的相关误差。顶部： 300 3×3×3 分箱，内存缩减96.25%。中部： 1000 3×3×3 分箱，内存缩减87.5%。底部： 1600 3×3×3 分箱，内存缩减80%。

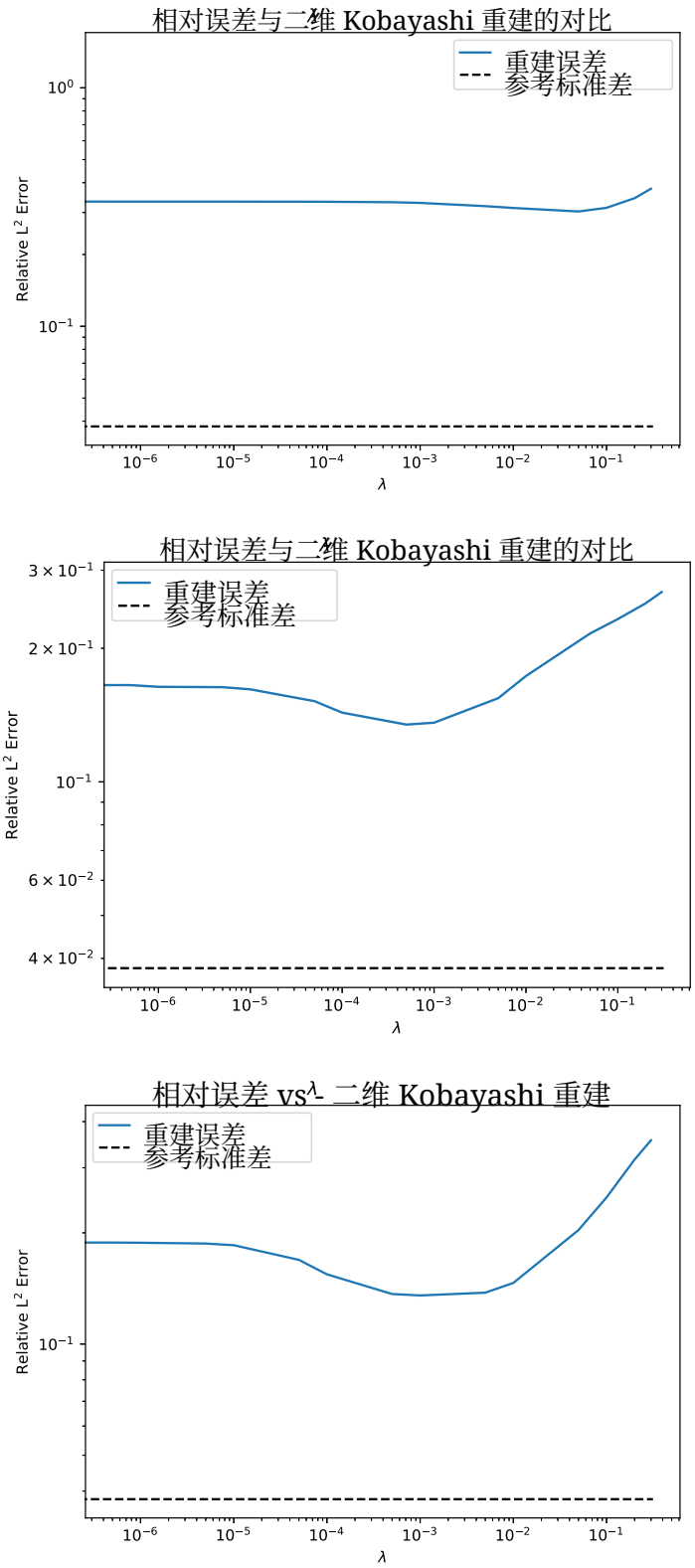


Figure 11. 与 Kobayashi 问题重建相关的误差。上： 300 3×3 分箱， 80% 内存 reduction。中： 1000 3×3分箱， 33.3% 内存缩减。下： 1000 2×2 分箱， 33.3% 内存缩减。

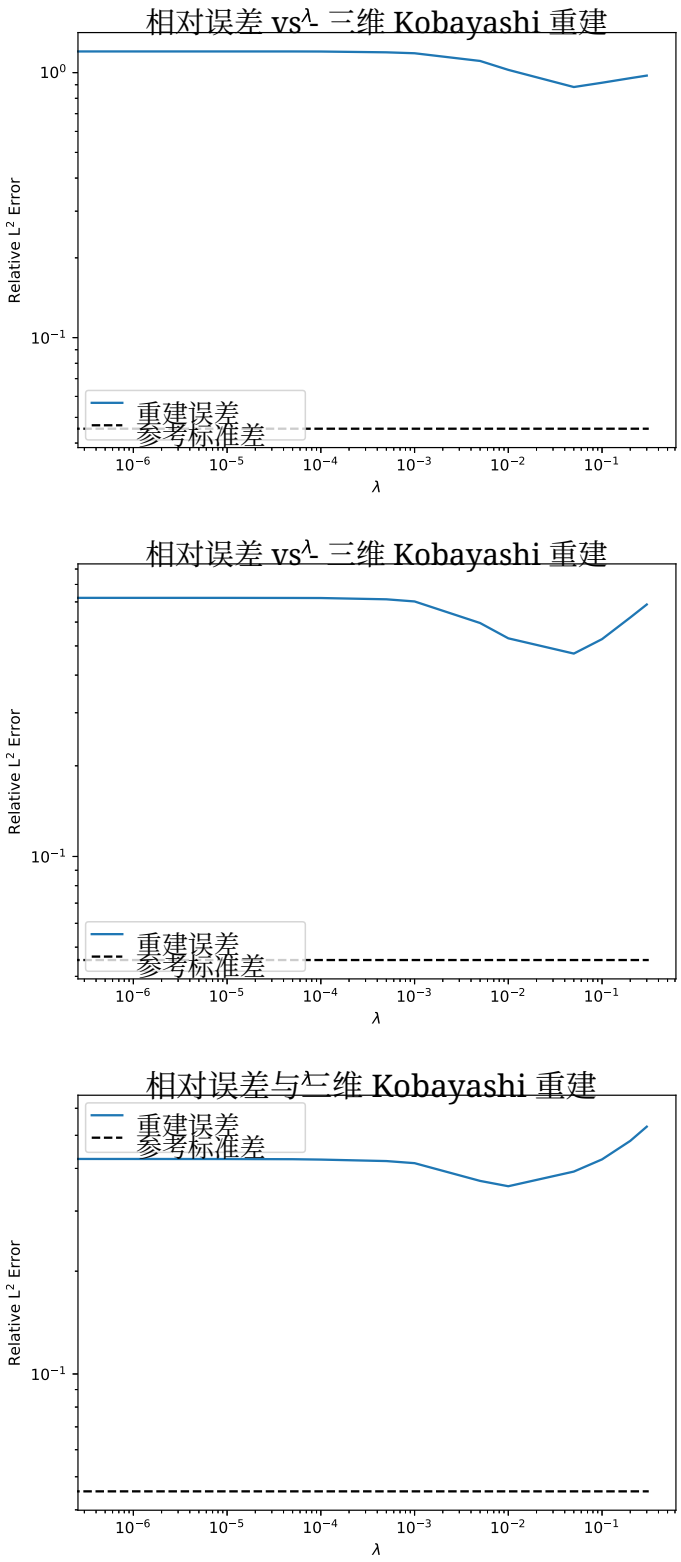


图12. Kobayashi 问题重建的相关误差。顶部： 300 2×2×2 个分箱，内存缩减 94.6%。中部： 1000 2×2×2 个分箱，内存缩减 82.22%。底部： 1500 2×2×2 个分箱，内存缩减 73.3%。

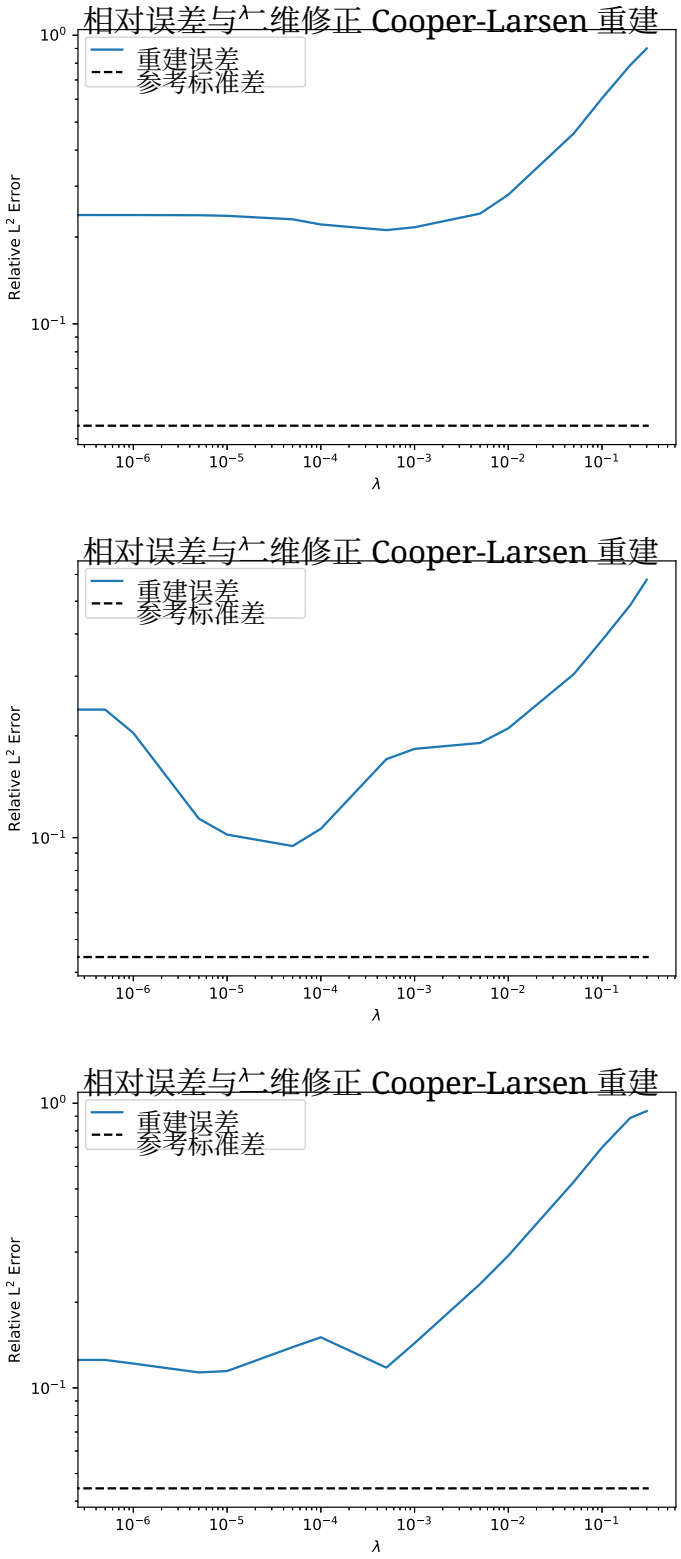


图 13. 修正 Cooper-Larsen 问题重建的相关误差。顶部： 300 3×3 个分箱，内存缩减 81.25%。中部： 1000 3×3 个分箱，内存缩减 37.5%。底部： 1000 2×2 个分箱，内存缩减 37.5%。

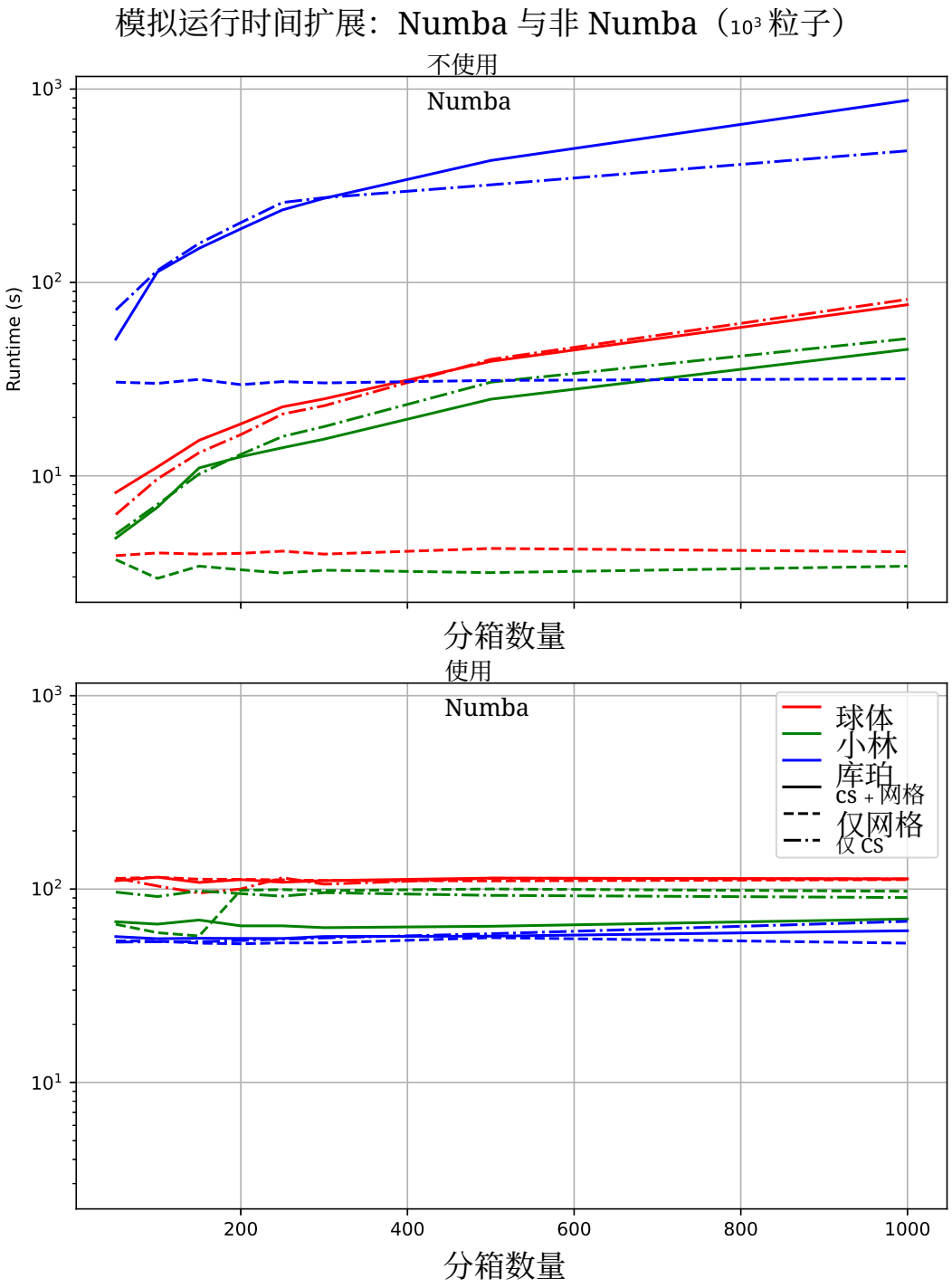
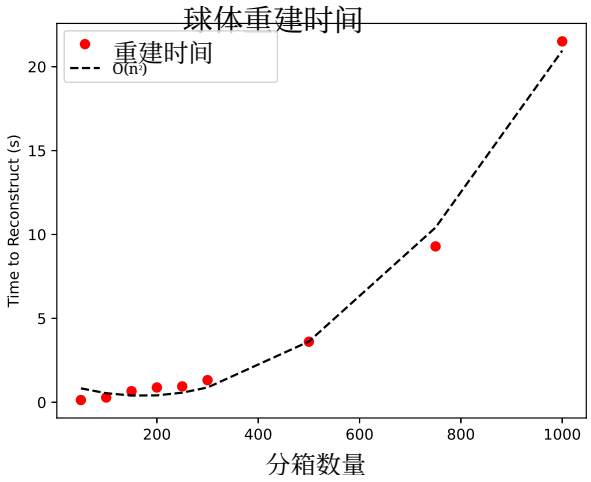
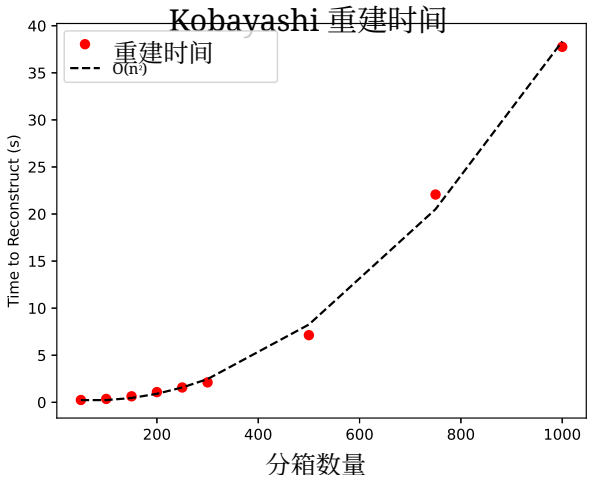


图 14。 多种情况下使用和不使用 Numba 的 MC/DC 模拟运行时间。





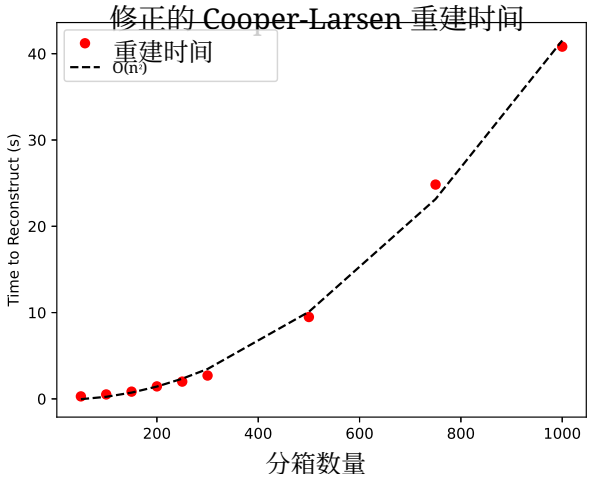


图 15. 查找重建所需的时间随粗分箱数量的变化。图中绘出了二次增长的拟合曲线。