arXiv:2303.08196v1 [physics.comp-ph] 14 Mar 2023

采用窗化多极形式的共振散射处理

Gavin Ridley、Benoit Forget、Timothy Burke

2023年3月16日

1 摘要

本文提出一种直接对共振上散射效应进行抽样的新方法。以往的方法依赖效率低下的拒绝抽样技术，或需要大量存储相对速度的表格数据。这些方法都需要逐点能量数据，因此并不特别适用于窗口化多极截面表示。新方法称为多极解析共振散射（MARS），它通过从靶核相对速度分布进行逆变换抽样来克服这些限制，其中截面以多极形式表示。相对速率分布的闭式表达式包含一种新颖的特殊函数，我们称之为不完全法德耶娃函数；本文还首次给出了该函数高效数值求值的结果。

2 引言

早期的连续能量蒙特卡罗中子输运程序在模拟热运动中原子的散射时，假设散射截面在散射核内基本保持恒定 [17]。然而，正如 [32, 30]详细说明的那样，由恒定截面近似所隐含的散射核，可能与散射共振附近的实际双微分截面相差甚远。正如 [24, 29], 所示，由此产生的误差可能导致压水堆中的多普勒反馈系数在最坏情况下被低估 11%，而在高温气冷堆问题中，偏差甚至更大。

PRAGMA项目 [8, 9], 表明，在图形处理器（GPU）架构上，传统的此效应处理方法仍有改进空间；而图形处理器构成了当今领先超级计算机大部分的计算能力。由于GPU架构独特，对标准蒙特卡罗中子追踪算法进行修改，能够显著加快计算速度 [9]。同样地，本文提出了一种适用于GPU的方法，在采用窗化多极（WMP） [23] 形式表示截面时处理共振上散射。具体而言，实现快速GPU代码的启发式原则是避免拒绝抽样和访问内存中相距较远的位置，而新方法做到了这一点。为介绍这种新方法，我们首先回顾一些用于模拟共振上散射的传统方法。

多普勒展宽拒绝校正（DBRC） [4]是早期提出的一种技术，用于处理散射中子能量附近相互作用截面的剧烈变化；而在此之前，人们曾使用 $S(\alpha,\ \beta)$ 表。[11]此后，该方法已被应用于许多连续能量蒙特卡罗中子输运程序 [4, 39, 35, 18], ，并被证明能够成功模拟共振上散射效应。然而，对于共振附近能量的中子，DBRC的拒绝概率最高可达99.995%。[31]

权重校正法（WCM） [24] 也能成功模拟共振对热运动原子核双微分截面的影响。即使采用恒定截面双微分自由气体分布，该方法也会调整粒子权重，以获得数值上正确的结果。WCM 同样具备我们新提出方法的优点，无需额外的拒绝循环或表格；然而，调整粒子权重会显著增加整体蒙特卡罗模拟的方差，从而降低对关注量的估计精度 [8]。

另一种称为靶核运动抽样（TMS） [36]的技术可用于模拟共振上散射效应。不过，这种方法仅适用于采用δ追踪技术的蒙特卡罗中子输运程序。遗憾的是，δ追踪技术在 GPU 上的性能似乎不尽如人意 [33]。

相对速度抽样（RVS）方法 [38, 31] 旨在降低用于模拟共振上散射的拒绝算法中常见的高拒绝率。从本质上说，这些方案抽样的概率分布与 $f(x)g(x)$成正比，其中$f(x)$是一个分布，而 $g(x) \in [0, 1]$。随后，从 $f(x)$中抽样，并以 $g(x)$的概率接受该样本。RVS 方法将直接抽样从热运动项转移到截面项，因此平均情况下的拒绝率会变高，但最坏情况下的拒绝率则大幅降低。

相对速率制表法（RST） [8] 旨在解决拒绝抽样对 GPU 性能造成的影响。RST 是首种无需拒绝循环或双变量散射分布表的共振上散射建模技术。RST 的核心依据是：靶核速度分布可分解为两部分，一部分是包含共振信息的边缘相对速率分布，另一部分是在给定靶相对速率条件下的简单靶核极角分布。因此，可以在逐点截面的特定区域内对单变量相对速率累积分布进行制表，随后直接抽样条件极角分布，无需任何额外数据或拒绝步骤。

尽管 RST 方法简单且在 GPU 上效果显著，但也存在一些明显的缺点。如果对问题中的所有核素都处理共振上散射效应，那么在逐点截面表示中，针对每个能量点和温度存储相对速率累积分布（CDF）将占用数 GB 的额外内存。因此，出于实际考虑，为避免内存用量过高，双微分截面中的共振影响只能针对问题中的少数几种核素进行处理。此外，该方法会引入一定的温度离散化误差，不过已有研究表明，这种近似是合理的。

如果能够避免使用预先制表的相对速度分布，就能大幅减少内存需求，并避免代价高昂的发散内存访问。减少 GPU 上的串行化内存访问通常会显著提升速度。

我们提出了一种新方法，能够实现这一目标，并通过窗化多极（WMP）截面表示 [23] 来实现这一点。我们的方法引入了一种新颖的特殊函数，称为不完全法德耶娃函数，用于编码共振的温度依赖影响在双微分散射分布中的表现。该方法依赖于对窗化多极形式下相对速度分布解析表示的数值反演，并以与 RST 方法相同的方式进行极角抽样，但无需任何预计算表。

这与 [27], 中提出的基于 WMP 的靶速度抽样技术形成对比；该技术在本质上与本文新提出的方法相似。然而， [27] 的方法依赖于将靶核速度分布拆分为零开尔文截面分量和麦克斯韦–玻尔兹曼分量。因此，该方法需要在拒绝循环中对积分散射截面函数进行数值反演，这与 [31] 类似，但它采用的是积分截面的函数表示，而非表格表示。

[6] 中提出的算法展示了如何在窗化多极框架中对相对速率进行抽样，这与我们的方法类似。然而， [6] 通过拟合高斯函数之和来取代式 4 中的极点，从而将零开尔文散射截面表示为：

$$\sigma(E) = \frac{1}{E}\sum_{k}\sum_{j}\left[h_{s,k,j}e^{\left(\sqrt{E}-u_k\right)^2/w_{s,k,j}^2} + h_{a,k,j}e^{\left(\sqrt{E}-u_k\right)^2/w_{a,k,j}^2}\right] \tag{1}$$

虽然高斯函数能否用于近似窗化多极数据库中出现的所有极点仍有待验证，但这种近似会给问题引入相当多的自由度，每个极点对应十八个未知数。因此，所面临的优化问题高度非线性且非凸，导致拟合困难。此外，无法直接对 [6] 的形式体系中的散射核关于窗化多极参数求导，而进行灵敏度分析需要这种求导。

我们提出的新方法——多极解析共振散射（MARS）——只需使用与不处理共振上散射效应的计算相同的窗化多极数据，因此无需额外的表格或拟合步骤。我们证明，与其他用于在 CPU架构上采样共振上散射效应的方法相比，该方法的性能开销可以忽略不计；未来将进一步探索其在 GPU 上的优化实现。

3 理论

Rothenstein 等人 [32] 阐述了严格的多普勒展宽双微分截面。由于实验室坐标系散射分布的精确表达式相当复杂，提出其建模算法的作者通常会舍弃实验室坐标系下的表达式，转而根据靶粒子速度与相对于入射粒子运动方向的方向余弦的联合分布进行推导。采样靶粒子速率和方向余弦后，采用标准的弹性散射二体碰撞运动学，

其中质心角分布取自核数据文件。由此得到的物理结果与 [32] 中的复杂表达式一致。

同样考虑碰撞靶粒子速度的分布，该联合分布为：

$$f(V,\mu) = Cv_r\sigma(v_r)M(T,V)f'(\mu) \tag{2}$$

其中 M 为靶粒子速率的麦克斯韦-玻尔兹曼分布：

$$M(T,V) = \frac{4}{\sqrt{\pi}}\beta^3 V^2 e^{-\beta^2 V^3} \tag{3}$$

具有速度倒数单位的变量 $\beta = \sqrt{\frac{A}{2kT}}$ 用于参数化靶粒子速度，而分布 $f'(\mu)$是在 -1 和 1 之间的均匀分布。其他变量为 C，即该分布的归一化常数； v_r，即中子相对于靶粒子的相对速率； σ，即零开尔文时的散射截面； A，即靶粒子质量； k，即玻尔兹曼常数； T，即绝对温度；以及 V，即靶粒子速率。

传统上，人们采用 $\sigma(v_r)$ 为常数这一近似。然而， [30, 24] 表明，该近似在共振附近并不成立；此时 $\sigma(v_r)$ 会变化数个数量级，从而更容易与相对速度更接近散射共振峰的靶粒子发生散射。上述各种技术都只是从公式 2 所示分布中进行抽样的方法，其中函数 $\sigma(v_r)$ 可以具有任意形式。

可以利用关于 $\sigma(v_r)$ 形式的更具体知识。大量研究已表明 [28, 13] ，截面可准确表示为若干极点与低阶洛朗展开（$N \approx 7$）之和，即：

$$\sigma(E) = \frac{1}{E}\left(\Re\left[\sum_j \frac{r_j}{p_j - \sqrt{E}}\right] + \sum_{n=0}^{N} a_n E^{n/2}\right) \tag{4}$$

事实上，为了对共振上散射效应进行采样，我们提出并将在后文通过数值计算加以证明：可实现的 v_r 范围很窄，因此在足够窄的能量范围内，零开尔文截面可以准确表示为一个极点和一个线性项：

$$\sigma(E) = \frac{1}{E}\Re\left[\frac{r_j}{p_j - \sqrt{E}}\right] + \sigma_0 + \sigma_1\sqrt{E} \tag{5}$$

σ_0 项和 σ_1 项通过线性化过程计算，以避免高阶多项式拟合带来的一些复杂性。第 3.4 节将介绍一种用于求取 σ_0 和 σ_1最佳值的算法。

采用这种近似，我们使用 [8] 中提出的技术：不尝试对靶粒子速率（V）和方向余弦（μ）进行采样，而是先对相对速度 v_r 进行采样，然后根据以 v_r 为条件的 μ 分布对 μ 进行采样。因此，入射粒子速率为 v 时的这种形式的分布为[8]:

$$P(v_r|v) = \left(e^{-\beta^2(v-v_r)^2} - e^{-\beta^2(v+v_r)^2}\right) v_r^2\sigma_0(v_r) \tag{6}$$

接下来，正如先前对窗化多极格式 [23]和原始完整多极格式 [21], 进行多普勒展宽时所示，我们注意到项 $e^{-\beta^2\ (v+v_r)}$ 可以忽略，并使用以下表达式：

$$P(v_r|v) \approx e^{-\beta^2(v-v_r)^2} v_r^2 \sigma_0(v_r) \tag{7}$$

此时，我们改用相对速度而非能量来表示多极截面。所用公式为：

$$\sigma(v_r) = \frac{2}{m_n v_r^2} \Re\left[\frac{r_j}{p_j - \sqrt{2m_n} v_r}\right] + \sigma_0 + \sigma_1 \sqrt{2m_n} v_r \tag{8}$$

若定义辅助变量 $x = \beta(v_r - v)$ 和 $y = \beta v$，并将式 8 代入以相对速度表示的目标碰撞率边缘分布（式 7），经过一些代数运算可得：

$$p(x|y) = e^{-x^2} \left(\Re\left[\frac{\beta r_j}{z - x}\right] + \beta^{-2}(x+y)^2 \left(\sigma_0 + \sigma_1 x\right) \right) \tag{9}$$

其中，$z = \beta p_j - y$ 表示麦克斯韦-玻尔兹曼分布中心与共振位置之间能隙的无量纲度量。

此时，我们可以考虑在给定 y （无量纲入射粒子速度）的条件下，随机变量 x （无量纲靶粒子速度）的累积分布函数。对式 9 积分可得：

$$CP(x|y) = \int_{-\infty}^{x} e^{-x'^2} \left(\Re\left[\frac{\beta r_j}{z - x'}\right] + \beta^{-2}(x'+y)^2 \left(\sigma_0 + \sigma_1 x'\right) \right) \mathrm{d}x' \tag{10}$$

其中，C 是归一化常数。展开积分并将算子 $\Re$ 与积分号交换后，可得：

$$CP(x|y) = \Re\left[\frac{r_j \pi}{i\beta^{-1}} w(z,x)\right] + \frac{\beta^{-2}\sigma_0}{4}(-2e^{-x^2}(x+2y) + \sqrt{\pi}(1+2y^2)(1+\mathrm{erf}(x))) + \frac{1}{2\beta^2}\sigma_1 e^{-x^2}\left(1 + (x+y)^2 + \sqrt{\pi} y (1+\mathrm{erf}(x))\right) \tag{11}$$

该分布的归一化常数为：

$$C = \Re\left[\frac{r_j \pi}{i\beta^{-1}} w(z)\right] + \frac{\sqrt{\pi}}{2\beta^2}\left(\sigma_0(1+2y^2) + \sigma_1 y\right) \quad , \tag{12}$$

我们指出，这不过是在温度 T 下采用单极点近似得到的多普勒展宽散射截面。

我们将这个新的特殊函数称为“不完全法德耶娃函数”，其定义如下：

$$w(z,x) = \frac{i}{\pi} \int_{-\infty}^{x} \frac{e^{-t^2}}{z - t}\, \mathrm{d}t \tag{13}$$

根据 $\Im[z] > 0$ 的法德耶娃函数定义，很容易看出 $w(z,\ \infty) = w(z)$：

$$w(z) = \frac{i}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z - t}\, \mathrm{d}t \tag{14}$$

确实，对于此应用，$\Im[z] > 0$，并且我们将在后续继续采用这一假设。我们开发了一种专用求根器，可快速反演此累积分布函数，从而对相对速度进行抽样。因此，这种方法无需拒绝抽样或大量查表，即可对靶粒子速度进行抽样。

我们方法的新颖之处完全在于零开尔文截面的处理，以及对相对速率累积分布的解析表示。按照 [8] 中的方法从相对速度分布中抽样后，我们必须根据相对速率对靶极角分布进行条件抽样。最后，为完整起见，我们给出靶粒子速率的累积分布函数：

$$C(V|v_r) \propto \begin{cases} 0 & V \leq |v_r - v| \\ 1 - e^{-\beta^2 V^2} & |v_r - v| < V < v_r + v \\ 1 & v_r + v \leq V \end{cases} \tag{15}$$

对此，[8] 提供了一种直接的抽样方法。剩余工作纯属数值计算，尤其需要一种高效且精度合理的不完全法德耶娃函数 $w(z,\ x)$ 算法。

3.1 不完全法德耶娃函数

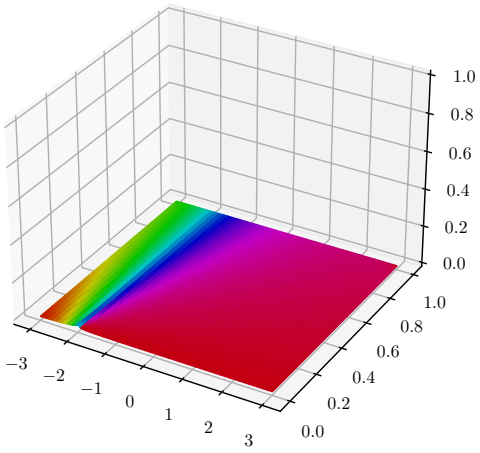
接下来的讨论将探究不完全法德耶娃函数的性质，尤其关注可用于构造其高效数值近似的性质。需要提醒读者的是，无须深入理解本节内容，也能掌握本文的论点和算法。

根据式 13 的定义，不完全法德耶娃函数为 $w\colon \mathbb{C} \times \mathbb{R} \to \mathbb{C}$。本节试图帮助读者直观理解该函数如何随 z 和 x 分别变化。在共振上散射处理中，z 描述共振相对于麦克斯韦-玻尔兹曼分布速度的位置、高度和宽度。$\Re[z] = 0$ 的取值对应于散射共振恰好位于麦克斯韦-玻尔兹曼分布所预测的、中子沿飞行方向的相对速度分量平均值处。$\Re[z] > 0$ 的取值对应于能量高于粒子能量的共振，因此会优先产生相对速度高于入射速度的散射；$\Re[z] < 0$ 则相反。x 描述无量纲靶相对速率。$\Im[z]$ 的值较小时，对应高而窄的共振；当 $\Im[z] = 0$ 时，极限情形为奇点，代表截面无限大的共振。$\Im[z]$ 的值较大时，则对应较宽、较弱的共振。

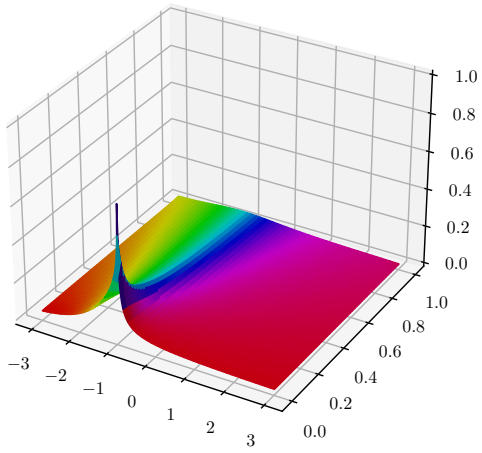
图 1 绘出了若干个 x 取值下，不完全法德耶娃函数关于 z 的变化曲线。这展示了当 $\Re[z] \approx x$时，该函数在 $\Im[z]$ 取值较小时变化迅速的特性，此时在实数轴附近，x 的取值之后会出现一个尖峰。此外，图中还显示了随着 $x \to \infty$，函数趋近于熟悉的 $w(z)$ 形状。图中表明，对于许多但并非所有的 z 取值，$w(z,\ x)$ 是关于 x的递增函数（即实部和虚部均递增）。这一行为可由以下渐近分析解释。

Humlicˆek [19] 给出了以下法德耶娃函数的渐近公式，但未作推导：

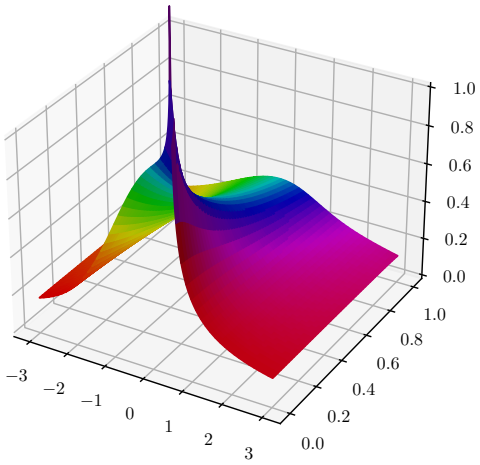
$$w(z) \approx \frac{i}{\sqrt{\pi} z} \quad . \tag{16}$$



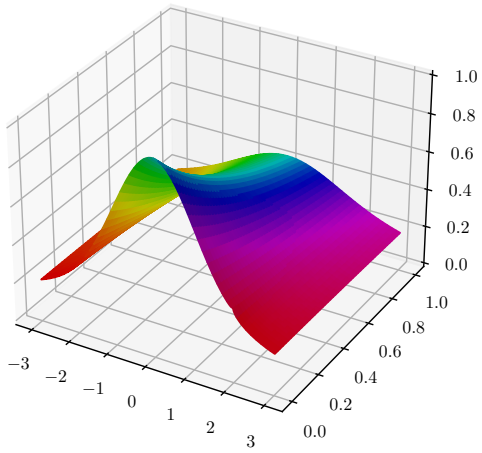
(a) $w(z, -2.0)$



(b) $w(z, -0.9)$

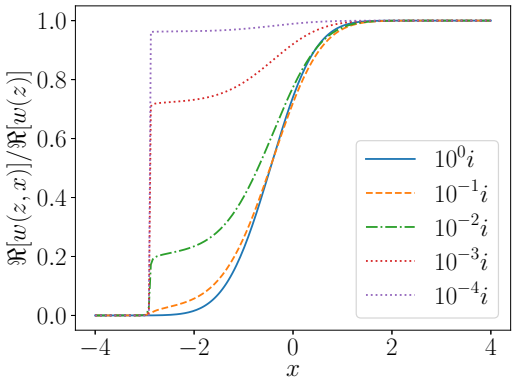


(c) $w(z, 0.5)$

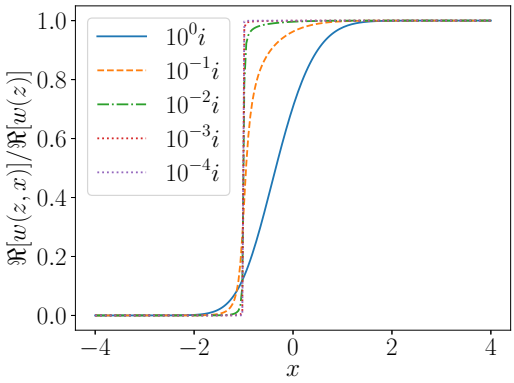


(d) $w(z, 2.0)$

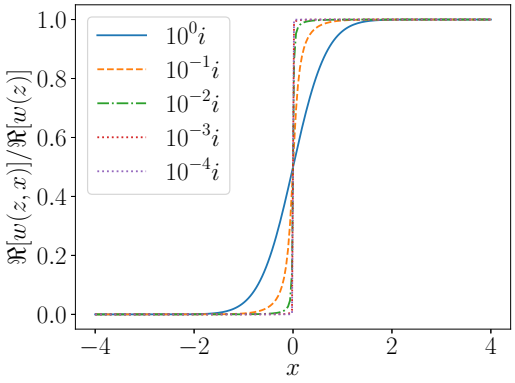
图 1: $w(z, x)$ 在若干个 x 取值下的结果。高度表示幅值，颜色表示相位。



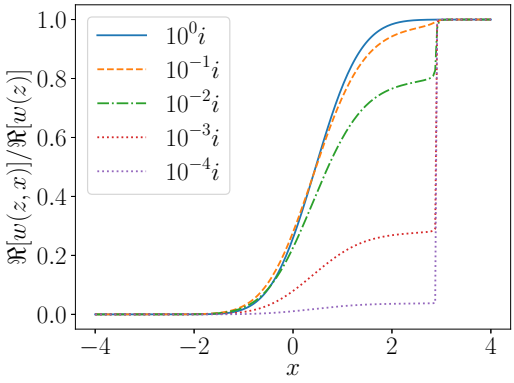
(a) $\Re[z] = -2.9$



(b) $\Re[z] = -1.0$



(c) $\Re[z] = 0.0$



(d) $\Re[z] = 2.9$

图2： $\Re[w(z,\ x)]$ 取若干个 z 值时的情况。图例表示在各图标题中指定的实部上加上虚数。绘制的量以 $\Re[w(z)]$ 归一化，因此随着 x 增大，所有曲线都趋近于 1。

再次考察式 14 中法德耶娃函数的定义。该近似基于这样的假设：如果 z 足够大，且只有 t 接近零的取值对积分有贡献，那么我们可以将 $w(z)$近似为：

$$w(z) \approx \frac{i}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z} \, \mathrm{d}t \tag{17}$$

由此立即得到式 16 的渐近估计。在不完全法德耶娃函数的情形下采用同样的近似，可得：

$$w(z,x) \approx \frac{i}{2\sqrt{\pi}z} \left(\mathrm{erf}(x) + 1 \right) \qquad \text{if} \quad |z| \gg 1 \tag{18}$$

这表明不完全法德耶娃函数在 x 处的行为与误差函数密切相关。事实上，以下恒等式将不完全法德耶娃函数与标准法德耶娃函数联系起来，证实了这一猜想：

$$w(z,x) = \frac{1}{2}(1 + \mathrm{erf}(x))w(z) + \frac{ie^{-x^2}}{\pi} \int_0^{\infty} \frac{e^{-t^2} e^{2itz} \, \mathrm{d}t}{i(x-z)+t} \tag{19}$$

这一关系的证明见附录 A。

对于较大的 $\Re[z]$，通过将极点项近似为以下形式，可以得到更准确的渐近估计：

$$\frac{1}{z-t} \approx \frac{1}{2z} \left(1 + e^{2t/z} \right) \quad . \tag{20}$$

该近似匹配了极点在 $t = 0$处相对于 t 的函数值、斜率和曲率。将其代回式 13，并调整表达式，使 $\Re[w(z,\ x)]$ 严格递增（如式 18 所示），同时匹配式 19 所示的 $w(z)$渐近值，可得

$$w(z,x) \approx \frac{1}{2} \left(1 + \mathrm{erf}\left(x - \Re[z]^{-1} \right) \right) w(z) \quad . \tag{21}$$

如图 3 所示，式 21 的精度足以满足实际计算的需要。在共振散射的背景下，式 21 表明，相对速度分布会向前偏移一个无量纲因子 $1/\Re[z]$ ，其影响按 $\Re[r_j w(z)]$ 乘以共振留数的比例变化，这与式 11 的提示一致。

不完全法德耶娃函数的另一个有用性质，是它在复平面中的导数与其函数值之间存在简单的关系。这与法德耶娃函数 [1] 的导数性质类似：

$$\frac{\mathrm{d}w(z)}{\mathrm{d}z} = \frac{2i}{\sqrt{\pi}} - 2zw(z) \tag{22}$$

我们得到的关系将其推广为：

$$\frac{\mathrm{d}w(z,x)}{\mathrm{d}z} = \frac{i}{\sqrt{\pi}} \left(1 + \mathrm{erf}(x) + \frac{e^{-x^2}}{\sqrt{\pi}(x-z)} \right) - 2zw(z,x) \tag{23}$$

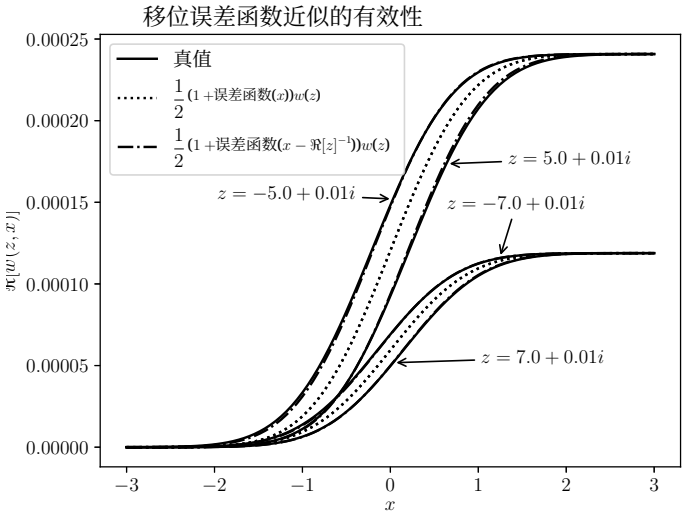


图3： $w(z,\ x)$近似对 $|\Re[z]| > 5$的准确性。图中还展示了未移位的误差函数以作比较，它代表了对 $w(z,\ x)$更朴素的渐近近似。

这显然与式 22 保持一致，如 $x \to \infty$ 所示。 $w(z,\ x)$ 的导数与其取值之间这一相当简单的关系，可用于高效地分析散射核对加窗多极参数的灵敏度。

下文将进一步介绍一些概念，以实现对不完全法德耶娃函数的高效数值求值。接下来，我们将式19中出现的第二个积分记为：

$$I(z, m) = \int_0^\infty \frac{e^{-t^2} e^{2itz}\,\mathrm{d}t}{im + t} \tag{24}$$

其中 $\mathbb{R} \ni m = x - z$。尽管半区间高斯-厄米特求积似乎可以很好地高效近似公式 24，但事实并非如此。首先，必须在每个求积点计算复指数函数。其次，随着 z 的实部增大，被积函数的振荡会更加明显。实际上，经常会遇到 $\Re[z] > 10$ 的值，而低阶求积无法捕捉这种振荡。此外，由于 $\Im[z]$ 的值较小，当 $x \approx \Re[z]$时，分母会变得近乎奇异。事实上，当 $\Im[z] = 0$时， $I(z,\ m)$ 在解释为柯西主值积分时，会在 x 中变成不连续函数。

公式 24 等价于不完全古德温-斯塔顿积分， [12]中将其称为该积分。然而，据我们所知，此前针对这类积分仅开展过渐近分析，尚未开发任何数值例程。金融领域的近期研究 [3] 给出了计算作者所定义的扩展不完全古德温 - 斯塔顿积分的结果，其中 $\nu = 1$ 情形正是本文讨论所关注的情况。遗憾的是，作者的数值方法适用于除 $\nu = 1$以外的所有情况，这表明公式 24 在本质上具有不同的性质。

根据我们的经验，采用驻相法 [5], 无法缓解大 $\Re[z]$ 带来的困难，因为这类方法往往会加剧极点行为，而且仍然存在类似的

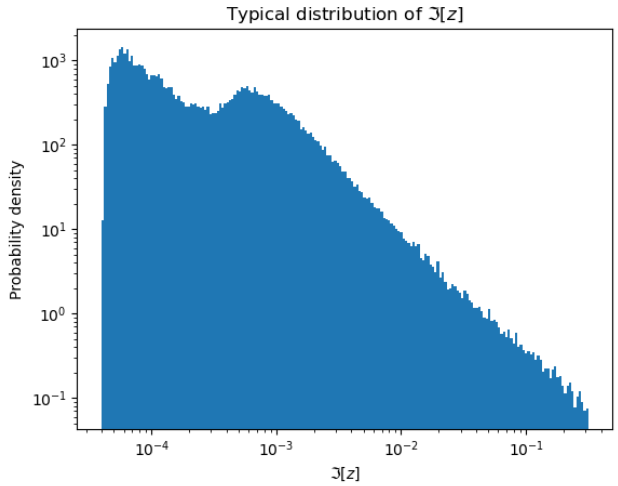


图 4：窗口化多极法中极点虚部的分布。这些数据收集自 OpenMC 回归测试数据集中的所有核素，该数据集基于 ENDFVII.1。

半区间高斯-厄米特求积困难。

在窗口化多极法中，$\Im[z]$接近于零，如图 4 所示。因此，我们可以预期在式 24 中经常遇到近奇异的被积函数。首先注意到 $I(z,\ m)$ 在复平面上满足以下微分方程，即可设计出一种能明确处理这种情况的高效数值技术：

$$\frac{\mathrm{d}I}{\mathrm{d}z} + 2zI = \frac{1}{x-z} \tag{25}$$

这可用于将点 z_0 处的值 $I(z_0,\ x)$ 与另一点 z_1处的值联系起来。积分因子法表明：

$$I(z_1, x) = \int_{z_0}^{z_1} \frac{e^{t^2-z_1^2}\,\mathrm{d}t}{x-t} + e^{z_0^2-z_1^2} I(z_0, x) \tag{26}$$

如果 z_0 与 z_1 之间的距离很小，则可将式 26 分子中的指数函数在 z_0 处展开为泰勒级数，从而得到一种高效的数值方案。

我们发现，$I(\Re[z],\ x)$ 的虚部可以用关于 x 的闭式不连续公式计算。由于 $I(z,\ x)$ 在 x 中的近似不连续行为是此处困难的主要来源，因此在计算出 $I(\Re[z],\ x)$ 的值后，可以利用式 26 准确地解析这种行为。$I(\Re[z],\ x)$ 的实部关于 x 连续，但无法用初等函数表示；不过，它很容易通过数值近似求得。总之，$I(\Re[z],\ x)$ 的实部和虚部截然不同：前者可通过级数或类似方法轻松进行数值近似，而后者是不连续的，因此不适合用级数或有理函数逼近，所幸它有精确公式。

当前的首要目标是计算实数 z 的 $\Im[I(z,m)]$ 。这可以写为：

$$\Im[I(z,m)] = \int_0^\infty \frac{e^{-t^2}\left(t\sin(2zt) - m\cos(2zt)\right)\mathrm{d}t}{t^2+m^2} \quad . \tag{27}$$

积分的线性可以分别作用于两个三角函数项。由此得到的各个积分可分别作为正弦和余弦变换积分求得，列于 [14] 中。结合正弦和余弦变换表中的结果可得：

$$\Im[I(z,m)] = \pi e^{-z^2+(m+z)^2}\left(\frac{1}{2}\mathrm{erf}(m+z) - \mathrm{sign}(m)\right) \quad z \in \mathbb{R} \tag{28}$$

接下来，我们可以求出 $I(z,m)$ 实部的一个近似：

$$\Re[I(z,m)] = \int_0^\infty \frac{e^{-t^2}\left(t\cos(2zt) + m\sin(2zt)\right)\mathrm{d}t}{t^2+m^2} \quad . \tag{29}$$

据我们所知，这个表达式既不能用初等函数表示，也不能用特殊函数表示。为了对其进行处理并得到数值表达式，请考虑辅助函数：

$$R(z,m) = \int_0^\infty \frac{e^{-t^2}\sin(2zt)\,\mathrm{d}t}{t^2+m^2} \tag{30}$$

当然，这个函数同样无法用初等函数或特殊函数表示。这指出了计算 $I(z,m)$ 实部的难点，因为：

$$\Re[I(z,m)] = mR(z,m) + \frac{1}{2}\frac{\partial R}{\partial z} \quad . \tag{31}$$

因此，我们寻求一个可直接求导的 $R(z,m)$ 近似。我们接下来对 $R(m,z)$ 进行数值近似的直觉依据是：对于 $m \gg 1$，在 e^{-t^2} 权重较大的范围内，t 相较于 m，在分母中的影响可以忽略，因此

$$R(m,z) \approx \frac{1}{m^2}\int_0^\infty e^{-t^2}\sin(2zt)\,\mathrm{d}t = \frac{1}{m^2}F(z) \tag{32}$$

其中 $F(z)$ 是道森 F 函数 [1]。根据标准渐近分析，在 $z=0$ 处匹配z 导数，并匹配道森 F 函数的 $z \gg 1$ 渐近线，可得到改进的估计：

$$R(m,z) \approx \frac{1}{m}\sqrt{e^{m^2}E_1(m^2)}F\left(mz\sqrt{e^{m^2}E_1(m^2)}\right) \tag{33}$$

在整个 m 值范围内，其精度约为 5%。不过，这种精度水平并不适用于工程计算。

为了进一步理解 $R(m,\ z)$，考察二者关于 z 的麦克劳林级数，即可看出它与道森 F 函数的密切关系：

$$F(z) = z - \frac{2}{3}z^3 + \frac{4}{15}z^5 - \frac{8}{105}z^7 + \cdots \quad , \tag{34}$$

$$R(m,z) = e^{m^2}\left(E_1(m^2)z - \frac{2}{3}E_2(m^2)z^3 + \frac{4}{15}E_3(m^2)z^5 - \frac{8}{105}E_4(m^2)z^7 + \cdots\right) \tag{35}$$

这进一步凸显了使 $R(m,\ z)$ 与其渐近形式 $F(z)$ 保持一致的价值。二者的级数相同，但在 $R(m,\ z)$ 中还乘有指数积分因子。

为了找到一种在渐近情况下仍能保持这种一致性的数值关系，可以通过交换微分与积分的顺序来变换公式 30。考虑以下广义函数：

$$R(m,z,\alpha)=e^{m^2}\int_0^\infty \frac{e^{-\alpha(t^2+m^2)}\sin(2zt)\,\mathrm{d}t}{t^2+m^2} \tag{36}$$

微分后可得：

$$\frac{\partial R(m,z,\alpha)}{\partial\alpha}=-e^{m^2}\int_0^\infty e^{-\alpha(t^2+m^2)}\sin(2zt)\,\mathrm{d}t=-\frac{e^{-m^2(\alpha-1)}}{\sqrt{\alpha}}F\left(\frac{z}{\sqrt{\alpha}}\right) \tag{37}$$

于是可以应用微积分基本定理：

$$R(m,z,\infty)-R(m,z,1)=e^{m^2}\int_1^\infty \frac{1}{\sqrt{\alpha}}e^{-m^2\alpha}F\left(\frac{z}{\sqrt{\alpha}}\right)\mathrm{d}\alpha \tag{38}$$

利用 $R(m,\ z,\ \infty)\ =0$ 这一事实，并进行变量替换，公式 30 可改写为：

$$R(m,z)=2e^{m^2}\int_1^\infty e^{-m^2t^2}F(z/t)\,\mathrm{d}t \tag{39}$$

这证实了我们关于 $F(z)$ 与 $R(m,\ z)$ 密切关系的猜想；它是道森 F 函数经过拉伸和缩放后的无限叠加。

我们成功地将 $F(z)$ 的一种近似代入公式 39。基于有理表达式和其他初等函数的 $F(z)$ 的知名近似 [26,25] 无法得出在数值上有用的表达式。然而，注意到：

$$\int_1^\infty e^{-m^2t^2}(z/t)^{(2k+1)}\,\mathrm{d}t=\frac{1}{2}z^{1+2k}E_{1+k}(m^2) \tag{40}$$

可以看出，鉴于指数积分的递推关系，采用幂级数形式近似 $F(z)$ 在计算上可能非常高效：

$$ne^{m^2}E_{n+1}(m^2)=(1-m^2e^{m^2}E_n(m^2)) \tag{41}$$

必须仔细考虑该关系 [16] 的浮点特性。计算误差的放大幅度可能变得任意大，后文将介绍一种专门的数值算法，以保证浮点稳定性。

因此，为了获得一个简单且高效的 $R(m,\ z)$ 近似，我们采用了 [20] 中给出的、适用于 $z\in[-5,5]$ 的切比雪夫展开。通过对 $F(z)$ 的切比雪夫展开进行更精细的分段划分（如 [10], 中所示），可以改进我们方法的结果；但为简化实现，我们采用了当前的方法。因此，如果将 $F(z)$ 的截断切比雪夫展开转换为幂级数基：

$$F(z)\approx\sum_{i=0}^{n}c_n(z/5)^{2*i+1} \tag{42}$$

我们得到如下形式的近似

$$R(m,z) \approx \sum_{i=0}^{n} \frac{c_n}{2} E_{1+i}(m^2)(z/5)^{2*i+1} \quad . \tag{43}$$

3.2 数值实现

3.2.1 稳定、高效地计算 an E_n 序列

指数积分从[16] 开始的向前递推关系中的误差放大倍数为:

$$|\rho_n| = \frac{x^n E_1(x)}{n! E_{n+1}(x)} \tag{44}$$

值得注意的是，反向递推关系的误差放大倍数是该值的倒数。此外， $|\rho_n|$ 是一个从 1 开始递增、达到最大值后再单调下降至低于 1 [16]的函数。因此，存在一个临界指标 n^* ，从该指标向两侧迭代可得到数值稳定的递归算法。对于式 43 的求值，这意味着将 z 中的多项式拆分为指标 n^* 以上和以下的部分。计算出 $e^{m^2} E_{n^*}(m^2)$后，使用霍纳法沿反向递推关系计算至 z阶项，并采用向前递归计算从 $2n^* + 1$ 到 $2n + 1$的多项式。

我们得到的一个简单结果是，满足以下条件的 n^* 的最小值:

$$\frac{x^{n^*} E_1(x)}{n^*! E_{n^*+1}(x)} < 1 \tag{45}$$

可以很好地近似为:

$$n^* \approx ex - \frac{1}{2}\log \pi \quad . \tag{46}$$

附录 B 展示了如何得到这一结果。图 5 展示了公式 46 的精度。利用这些信息，算法 1 明确给出了计算 $R(m,\ z)$ 及其关于 z 的导数的步骤。虽然从数值角度来看，使用幂基多项式并非最优，但在这种情况下，数值误差的主要来源是递归指数积分公式。该算法的说明假设已经提供了一种准确计算 $E_n(x)e^x$ 的方法；许多其他文献对此都有详尽的介绍。我们采用了 Cephes 库 [7], 所用连分数近似法的 C++ 适配版本，该方法详见 [1]。

3.2.2 跳跃积分

计算出 $I(\Re[z],\ m)$ 的值后，可以使用其后续的式 25 微分方程来计算 $I(z,\ m)$。由于 $I(z,\ m)$ 的行为几乎不连续，我们采用这种方法计算 $I(z,\ m)$；当 $z \in \mathbb{R}$ 时，它在 $m = 0$ 附近存在跳跃间断。由于窗函数多极库中 z 的虚部很小，因此其行为几乎不连续，通用近似技术无法高效捕捉这种行为；为此需要使用分辨率很高的表格或很高的多项式阶数。我们的方法

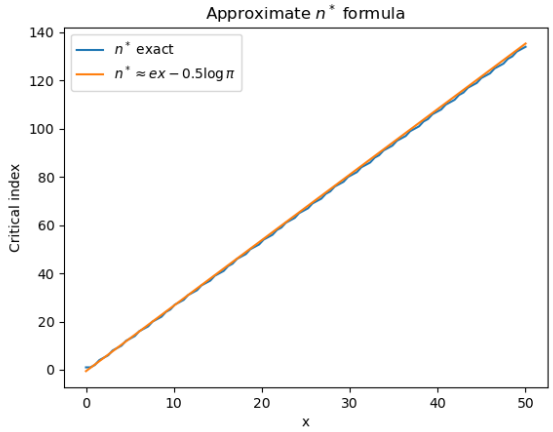


图5：求解首个满足 n 的值的近似解，其中 $x^n E_1(x)\ n!E_{n+1}(x) < 1$。

因此，通过分段函数式 28 精确解析不连续分量。式 26 的非平凡部分是超越积分：

$$J(z,x) = e^{-\Re[z]^2} \int_{\Re[z]}^{z} \frac{e^{t^2}\,\mathrm{d}t}{x-t} \tag{47}$$

我们将这一项称为跳跃积分，因为它可以让我们从实数轴上的 $I(z,m)$ 的值跳转到复平面中实数轴上方的值。乍看之下，由于又出现了一个超越积分（公式 47），我们对超越积分 $w(z,x)$ 的近似问题似乎仍未得到解决；但通过变量替换，可以将其化为适于数值近似的形式：

$$J(z,x) = \int_{0}^{i\Im[z]} \frac{e^{u^2+2u\Re[z]}\,\mathrm{d}u}{m-u} \tag{48}$$

其中同样有 $m = \Re[z] - x$。如图 4 所示，由于 $\Im[z]$ 很小，指数项中的自变量也同样很小。在该积分有定义的情况下（$m \neq 0$），可以将指数项展开为麦克劳林级数，并逐项积分：

$$e^{u^2+2u\Re[z]} = \sum_{n=0}^{\infty} \frac{a_n}{n!} u^n \quad . \tag{49}$$

经验证，系数 a_n 满足二项递推关系：

$$a_{n+1} = 2\Re[z]a_n + 2(n-1)a_{n-1}; \quad a_0 = 1; \quad a_1 = 2\Re[z] \tag{50}$$

接下来，逐项积分的形式如下：

$$\int_{0}^{i\Im[z]} \frac{u^n\,\mathrm{d}u}{m-u} = m^n B_{i\Im[z]/m}(1+n,0) \tag{51}$$

其中 $B_x(\cdot,\cdot)$ 是不完全贝塔函数，其定义为：

$$B_x(a,b) = \int_0^z t^{a-1}(1-t)^{b-1}\,\mathrm{d}t \quad . \tag{52}$$

由 [1] 中公式的一个特例推导出的递推公式，可以高效地依次计算这些高阶不完全贝塔函数值：

$$B_x(n+1,0) = B_x(n,0) - \frac{x^n}{n} \tag{53}$$

结合以下事实：

$$B_x(1,0) = -\log(1-x) \quad , \tag{54}$$

由此可得一种高效的数值方案，用于计算式 48 中指数函数的截断麦克劳林级数的积分。算法 2 详细说明了如何结合上述所有事实，高效近似 $J(m,\ z)$。对于具有 $|\Re[z]| \leq 5$ 的问题，这种近似效果很好；它涵盖了共振对温度下的双微分截面产生显著影响的散射事件范围。在此范围之外，积分的振荡性会逐渐增强，因此对 $|z| > 5$ 采用渐近近似。附录 D 介绍了这种近似。

最后，算法 3 给出了高效计算 $w(z,\ x)$ 的整体算法。该算法依赖于能够调用某种计算 $w(z)$ 的实现，例如采用宽松许可协议的 [22] ，它实现了多种近似方法以实现高精度；或者在允许较大误差时，采用各种有理近似方法之一 [34, 2, 19]。无论选择哪种 $w(z)$ 实现，我们的算法都能保持渐近一致性，即 $\lim_{x\to\infty} w(z,\ x) = w(z)$。本研究采用了一种近期针对 WMP 量身定制的近似方法 [15]。

3.3 极点采样近似

我们的方法之所以具有较高的计算效率，关键在于将相对速度分布中的多极截面视为混合分布。其理论依据是：如果存在极点，且这些极点与入射中子能量（具体而言为 $|z| < 20$ ）足够接近，那么忽略多项式贡献即可对相对速度概率密度函数作出良好近似：

$$P(x|y) \approx e^{-x^2}\Re\left[\sum_{j\in W(\beta^{-2}y^2)} \frac{\beta r_j}{z_j - x}\right] \tag{55}$$

该表达式并不用于实际采样散射分布。更恰当地说，应将其视为一种混合分布，其中每个极点的概率与以下量成正比：

$$\mathbb{P}[\sigma_s(x) = \frac{\beta r_j}{z-x} + \sigma_{0,j} + \sigma_{1,j}x] \propto \Re\left[r_j w(z_j)\right] \tag{56}$$

这定义了一个离散分布。为了避免需要辅助存储以及计算该分布的归一化常数，我们建议寻找

输入： $m \in \mathbb{R}$, $z \in \mathbb{R}$输出： $R(m, z)$， 来自公式 30 和 $\frac{\partial R}{\partial z}$ 数据：公式 42 的 c_n 系数，
$1 \leq n \leq n_{max} n^* \leftarrow \min(\max(em^2 - 0.57, 1)\ n_{max})$;
expnexp $\leftarrow E_{n^*}(m^2) e^{m^2}$; expnexp orig $\leftarrow$ expnexp; result $\leftarrow 0$; 导数 $\leftarrow 0$;
// Backward recurse with Horner schemefor $n \leftarrow n^* - 1$ to 1 do expnexp $\leftarrow (1 - n$ expnexp) $/m^2$; 导数 $\leftarrow$ 导数 $\cdot$ pow($z/5$, 2)+ result; result $\leftarrow$ (result)pow($z/5$, 2) $+$ expnexp $\cdot\, c_{n-1}$; end 导数 $\leftarrow 2$导数 $\cdot$ pow($z/5$, 2) ; 导数 $\leftarrow$ 导数 $+$ result ; result $\leftarrow$ result $\cdot\, z/5$;
// Forward recursing polynomial evaluation
x2 $\leftarrow$ pow($z/5, 2n^* - 1$); x1 $\leftarrow$ pow($z/5$, $2(n^* - 1)$); expnexp $\leftarrow$ expnexp orig; for $n \leftarrow n^*$ to n_{max} do result $\leftarrow c_{n-1}$·expnexp $\cdot$ x2; 导数 $\leftarrow c_{n-1}$· expnexp $\cdot$ x1 ·$(2n - 1)$; x2 $\leftarrow x2 \cdot$ pow($z/5$, 2); x1 $\leftarrow x1 \cdot$ pow($z/5$, 2); expnexp $\leftarrow (1 - m^2 \cdot$ expnexp)$/n$; end return (result, 导数);

A算法 1： 针对 $|z|$ 的稳定 $R(m, z)$ 近似 $\leq 5_\circ$ 附录 C 用于 $|z| > 5_\circ$

输入： $x \in \mathbb{R}$, $z \in \mathbb{C}$**输出：** $J(m,\ z)$ 根据公式 47 得到结果 $\leftarrow 0$; a0 $\leftarrow 1.0$; a1 $\leftarrow 2\Re[z]$; 得到结果 $\leftarrow$ b1 $\cdot$ a0; b 1 $\leftarrow b1 - i\Im[z]/(x - \Re[x])$;
// n_{max} adjusts the number of truncated series terms
对 $n \leftarrow 2$ **到** n_{max} **执行**tmp $\leftarrow 2\Re[z] \cdot$ a1 $+ 2(n-1)\ \cdot$ a0; a 0 $\leftarrow$ a1; a1 $\leftarrow$ tmp; result $\leftarrow$ result $+$ b1 $\cdot$ a$^{1\ {}^{m^n}_{n}\ !}$; b1 $\leftarrow$ b1 $- {}^{1}_{n+1}\left({}^{i\Im[z]}_{x-\Re[x]^{()}}\right)$;

结束返回结果;

算法 2： 适用于 $\Im[z] < 1$ 和 $|\Re[z]| < 5$的高效 $J(m,\ z)$ 近似。附录 D 介绍了针对 $|\Re[z]| > 5$ 的近似。

输入： $z \in \mathbb{C}$, $x \in \mathbb{R}$输出：根据公式 13 得到 $w(z, x)$ m $\leftarrow x - \Re[z]$; // $R(m, z)$ integral and its derivativermz, drmzdz $\leftarrow$ 调用（算法 1）；// $I(m, z)$ integralimz $\leftarrow m \cdot$ rmz $+ \frac{1}{2}$drmzdz $+ \frac{i\pi}{2} e^{-\Re[z]^2+x^2}$ (误差函数(x) $-$ 符号函数(m));// $J(m, z)$ integralji $\leftarrow$ 调用（算法 2）；imz $\leftarrow$ imz+ji；imz $\leftarrow$ imz$\cdot e^{\Re[z]^2-z^2}$；结果 $\leftarrow \frac{i}{\pi} e^{-x^2} \cdot$ imz；结果 $\leftarrow$ result $+ \frac{1}{2}$ (误差函数(x) $+ 1$) $w(z)$；返回结果；算法 3： $\Im[z] << 1$的高效 $w(z, x)$ 近似。

取 $-|r_j w(z_j)|/\log(\xi_j)$ 的最大值，其中 ξ_j 是针对每个极点各不相同的均匀随机数。与该表达式最大值对应的 j 符合所需的离散分布。我们还注意到，量 $\Re\left[r_j w(z_j)\right]$ 恰好是积分散射截面中的多普勒展宽贡献，因此，如果将此采样过程与 WMP 截面查找操作同时进行，就不会增加法德耶娃函数的计算开销。

最后，我们强调，极点采样近似与原始多极截面表示并非完全一致。相反，它利用了这样一个事实：多项式对截面的贡献对散射核影响可以忽略不计，而极点的影响则十分显著。

3.4 求取 σ_0 和 σ_1的值

乍看之下，式 5 中的多项式贡献似乎可以取自窗口内定义的多项式的首项，就像使用 [6] 时那样；但我们在实践中发现，这种选择与极点采样技术的近似不一致，并会导致截面值为负。

为解决这一问题，我们采用如下启发式方法：通过匹配共振谷附近的多项式值，使截面局部近似的相对误差最小化。散射共振谷的位置计算如下：

$$\sqrt{E}_{\text{trough}} = \frac{-b + \sqrt{b^2 - abc + a^2 d}}{a} \tag{57}$$

其中

$$a = -\Im[\left(r_j^* - \bar{r_j^*}\right)] \tag{58}$$

$$b = -\Im[\left(\bar{r_j^*} p_j^* - r_j^* \bar{p_j^*}\right)] \tag{59}$$

$$c = -\Re[\left(\bar{p_j^*} + p_j^*\right)] \tag{60}$$

$$d = |p_j^*|^2 \tag{61}$$

$$\tag{62}$$

此时，$\sqrt{E}_{\text{trough}}$ 的窗口索引按 [1] 计算。这可能与入射中子能量所在的窗口不同。随后，在 $\sqrt{E}_{\text{trough}}$ 处计算公式 4 的窗口化多极截面，但不包括抽样极点 p_j，即

$$\sigma_0 = \frac{1}{E_{\text{trough}}} \Re\left[\sum_{j^* \neq j \in W(E_{\text{trough}})} \frac{r_j}{p_j - \sqrt{E_{\text{trough}}}}\right] + \sum_{n=0}^{N} a_n E_{\text{trough}}^{n/2} \tag{63}$$

[1]式 57 中根号下的项有时会在非物理极点附近变为负值；这些极点是拟合过程产生的伪影，在这种情况下处理虚数并不可取。因此，在这种情况下，我们的计算改为对入射能量处的非极点截面进行线性化。

随后，在同一点计算 $\sigma_1 \approx \frac{\partial \sigma_{s(E)}}{\partial \sqrt{E}}$，这次仅包括多项式展开的贡献，不包括任何极点的贡献。研究发现，在处理共振间距很小的核素（例如 ^{235}U）时，这种线性化技术能够提高我们方法的准确性。为清晰起见，最终表达式为：

$$\sigma_1 = \sum_{n=0}^{N} a_n \frac{n-2}{2} E_{\text{trough}}^{n/2-2} \tag{64}$$

最后，已获得截面在 $\sqrt{E}$ 空间中的线性化结果。为了配合求根器，使用无量纲变量 β $(\sqrt{E} - \sqrt{E_{\text{incident}}})$更为合适，因此应适当平移 σ_0 并适当缩放 σ_1 。

3.5 反演相对速率累积分布函数

为了从相对速度概率密度函数中进行抽样，我们采用累积分布函数反演技术。一个简单的做法是先进行几步二分法求根，再进行少量牛顿类迭代。实践中，我们发现，五次二分迭代后接着进行三次 Halley-Newton 迭代，便能将根求解到可接受的容差范围内；不过，我们开发出了一种效率高得多的求根器，总共最多只需四次迭代，仅在处理罕见的边缘情况时才需要更多计算。

我们称之为自举步骤，它对于高效实现 MARS 至关重要。自举步骤能够以较低成本获得累积分布函数反演问题解的初始猜测值，再通过少量牛顿类迭代改善解。

为此，关键在于找到一个计算开销较低的近似，用于求一般极点参数下累积分布函数的反函数。为此，我们首先从$x \in (-\infty, \infty)$的求根空间转到非线性映射后的变量 $\tilde{x} = \frac{1}{2}($ $1 +$ 误差函数$x)$。采用这一变换空间的直觉依据是：随着共振变弱、入射中子能量升高，可以证明累积分布函数恰好等于 $\tilde{x}$，其取值范围为零到一。共振和低能自由气体效应只是对这一线性函数的扰动，因此可以从一个良好的初始值出发，近似求得根的位置。

改进映射空间中累积分布函数模型的下一步，是观察到共振产生的碰撞概率贡献在很大程度上不依赖于其虚部。增大共振的虚部会使其展宽并降低其宽度。因此，当 x 接近 $\Re[z]$ 时， $w(z, x)$的跃变量化了中子在共振峰附近发生碰撞的概率。对于较小的 $\Im[z]$ ， w (z, x)的跃变量为：

$$\lim_{\Im[z]\to 0} \int_{\Re[z]-\epsilon}^{\Re[z]+\epsilon} \frac{e^{-t^2}\,\mathrm{d}t}{z-t} = e^{-\Re[z]^2} \tag{65}$$

因此，共振导致的概率贡献约为：

$$p_{\text{jump}} = \mathbb{P}[x \approx Re[z]] \approx \beta r_{j^*} \pi e^{-\Re[z]^2}/C \tag{66}$$

其中， C 是由公式 12 给出的归一化常数。由于这是一种近似，公式 66 中的概率可能不在零到一之间，因此我们将其截断至

该范围内。实际应用中，此处给出的估计是准确的。我们发现，在区间 $[\Re[z]-\frac{3}{2}\Im[z], \Re[z]+\frac{3}{2}\Im[z]]$ 内，这一概率通常会使累积分布函数增加约 $\Re[z]$。显然，可以调整此估计以提高准确性。

我们用于启动求根过程的最后一个工具，与 $x=0$ 附近的累积分布函数值有关。在这种情况下，其表达式中多次出现的函数，例如 $\mathrm{erf}(x)$ 和 e^{-x^2}，都可以轻松计算。当 $x=0$ 时，不完全法德耶娃函数还有一个闭式表达式：

$$w(z,0)=\frac{1}{2}w(z)+\frac{i}{2\pi}e^{-z^2}E_1(-z^2) \quad . \tag{67}$$

由于在反演累积分布函数时已经计算并缓存了 e^{-z^2}，计算复指数积分便成了唯一的难点。这比计算更复杂的 $w(z,\ x)$ 容易得多，计算成本也低得多，因此这里可以采用任意现成的 $E_1(-z^2)$ 近似。

此外，$x=0$ 附近累积分布函数关于 x 的导数也很容易求得，我们利用这些导数进一步改进求根初值。目前为止，我们只纳入了一阶导数的信息，事实证明这已足够。

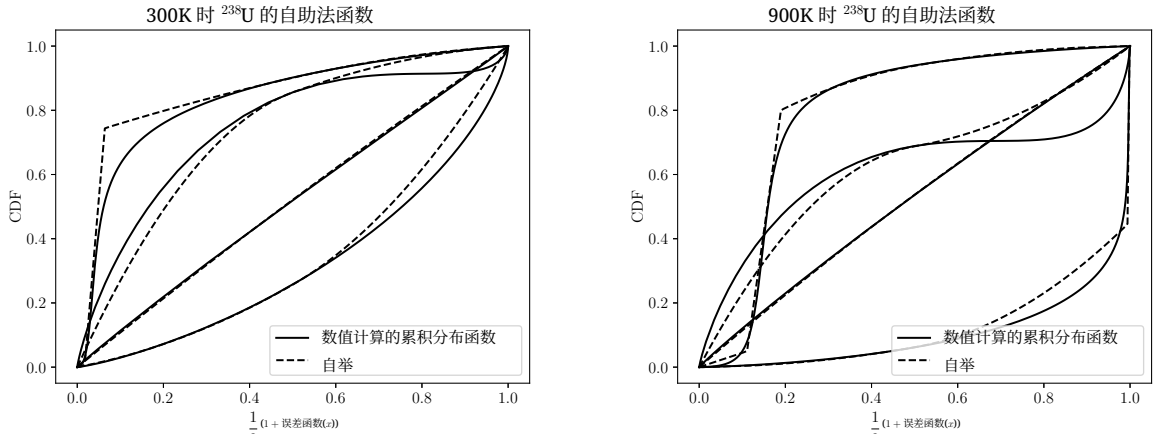
据此，我们可以利用以下信息提取根的估计值：共振所对应的概率、其宽度、$x=0$ 附近累积分布函数的值和斜率（即 $\tilde{x}=1/2$），以及累积分布函数在 0 和 1 处已知的端点值。因此，我们构造一个函数，使其在 $\tilde{x}=1/2$ 左右两侧分别为二次函数。这个二次区间延伸至端点 $\tilde{x}=0$ 或 $\tilde{x}=1$，或者延伸至共振概率增量的上限或下限；此处将其估计为 $\mathbf{x}\in[\Re[z]-\frac{3}{2}\Im[z], \Re[z]+\frac{3}{2}\Im[z]]$。注意，这个区间必须映射到 $\tilde{x}$ 空间中的一个区间。由于共振区间较小，可以利用变换 $x\mapsto\tilde{x}$ 的雅可比行列式计算其在 $\tilde{x}$ 空间中的范围；该雅可比行列式与 $e^{-\Re[z]^2}$ 成正比（这是一个已经计算出的量）。

有了这些知识，就可以在 $\tilde{x}$ 空间中较为准确地近似累积分布函数。尽管刚才描述的内容看起来相当复杂，但前一段所述函数的反函数可以通过简单的分支逻辑求解，最多只需解一个二次方程。由于将上述函数的反函数转换为代码需要花费不少功夫，附录 E 中提供了实现此功能的 C++ 代码。图 6 展示了两个示例，说明当共振影响散射分布时，在 $\tilde{x}$ 空间中对累积分布函数进行这种近似效果相当不错。

4 结果

4.1 计算 $w(z,\ x)$

为了检验算法 3 的准确性，我们使用 `scipy` 的 [37] 自适应求积程序 `scipy.integrate.quad` 计算了 $w(z,\ x)$ 的参考值，以求解积分表达式公式 13。在近似跳跃积分公式 47 时，仅保留级数的前五项。对于 $\log(x)$ 或 e^x 等函数，我们采用了 C++ 标准库实现。我们采用了来自 [22] 的 $w(z)$ 实现。这得到了图 7 所示的误差分布图，其中我们绘制了



(a) $T = 300$ 开尔文　　(b) $T = 900$ 开尔文

图 6：自举累积分布函数提供了一个相当准确且易于求逆的近似，用于启动求根过程，逼近真实的相对速度累积分布函数。从上到下排列的各对曲线分别表示入射中子能量为 35.25、36.25、38.25 和 66.25 eV 时的情况。

$\left(w_{\text{approx}}(z,\ x) - w(z,\ x)\right)/w(z)$ 的实部。由于共振上散射计算只关注实部，因此省略了虚部误差的结果。

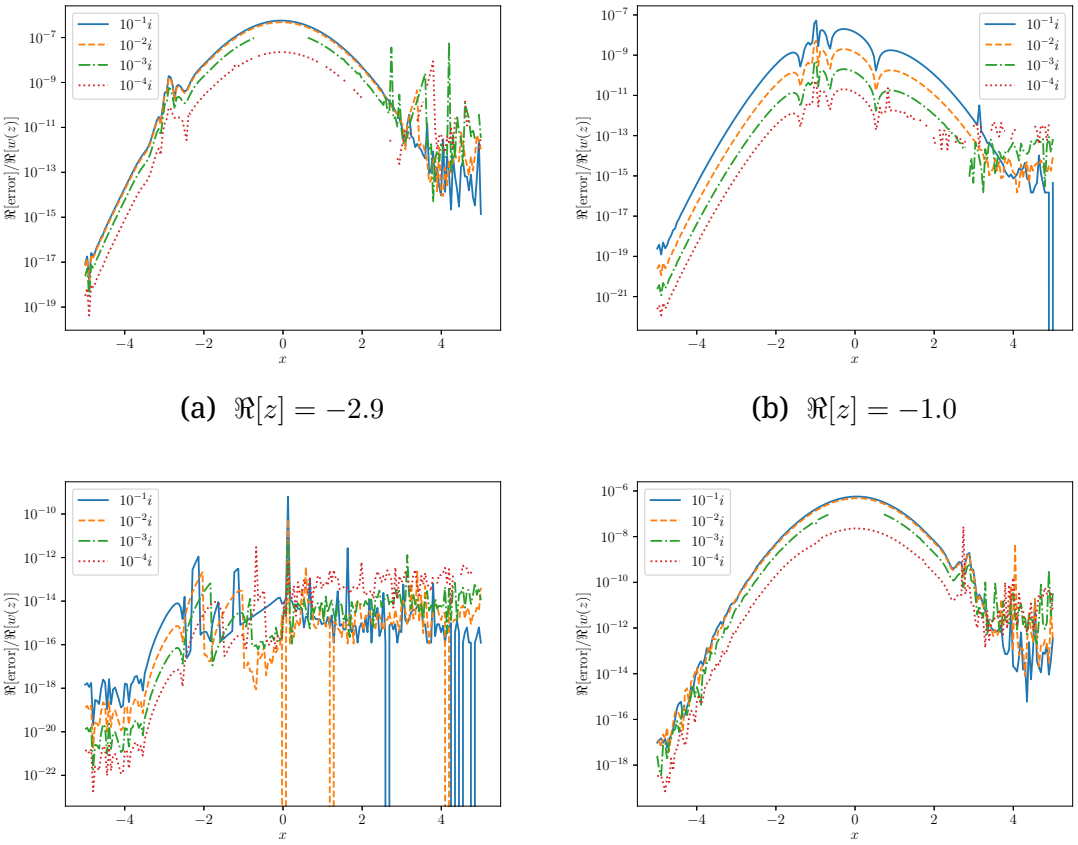
4.2 单能测试

我们首先在图 8 中展示了通过数值积分和 MARS解析累积分布函数计算得到的 ^{238}U 在两种不同能量和若干温度下的相对速度分布。这些能量分别对应于散射共振的谷底和峰值附近。图中清楚地显示了共振对双微分截面的影响；在超热能区，截面恒定的核素所具有的相对速度分布与误差函数非常接近。共振附近的相对速度分布会出现跳变，这种现象由 $w(z, x)$ 控制。它们与图 2 所示的 $w(z, x)$ 行为相似。

如果相对速度分布正确，那么由此得到的双微分散射分布也正确。图 9 显示，与 [31] 的 RVS 方法相比，我们的方法确实如此。这些结果由我们修改后的 OpenMC 版本得出，该版本可在 `github.com/gridley/openmc/tree/mars` 获取。图中还显示，极点抽样技术对于 ^{235}U 及其间距紧密的共振确实有效。

4.3 栅元反应性反馈

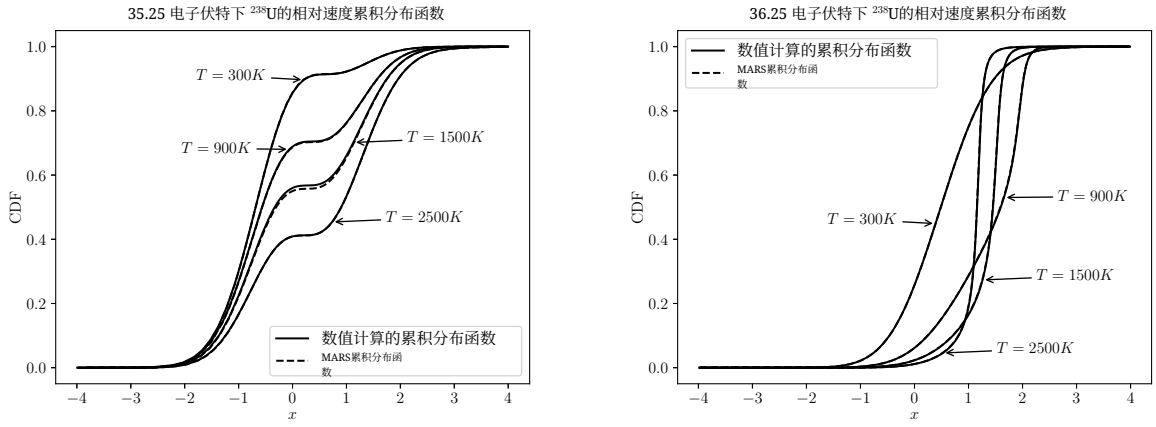
采用 OpenMC 示例问题集中的 2.4% 富集度压水堆燃料棒元件示例，使用四种不同模型计算多普勒反应性反馈效应。第一种和第二种模型使用了在 300、600、900、1200 和 2500 开尔文之间插值的逐点截面。模型的运行温度范围为 300 至 1800



(a) $\Re[z] = -2.9$ (b) $\Re[z] = -1.0$

(c) $\Re[z] = 0.0$ (d) $\Re[z] = 2.9$

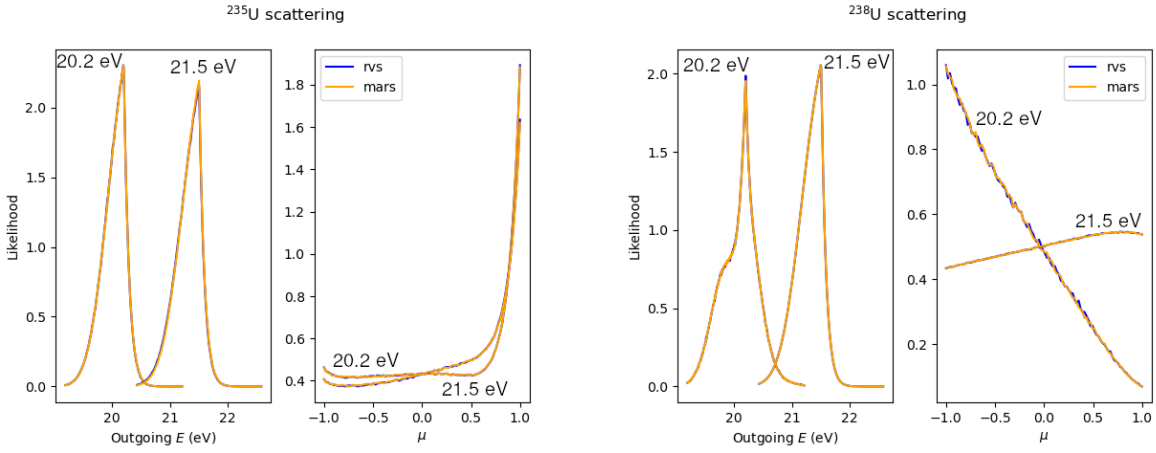
图 7: $\Re[w(z,x)]$ 在若干个 z取值下的误差。图例表示各图标题中指定的实部所加的虚数。绘制的误差 $w_{\text{approx}}(z,x) - w(z,x)$ 经 $\Re[w(z)]$ 归一化，以匹配图 2 的缩放比例。



(a) 35.25 电子伏特

(b) 36.25 电子伏特

图 8：MARS解析累积分布函数与数值积分得到的相对速率累积分布相匹配。1500开尔文、35.25 电子伏特时可以观察到一些误差；幸运的是，在共振谷发生散射的情况极为罕见。



(a) ^{235}U

(b) ^{238}U

图 9：在两个不同能量下，1200开尔文时的散射结果与RVS方法的结果吻合良好。这些能量与两种核素的共振发生相互作用。

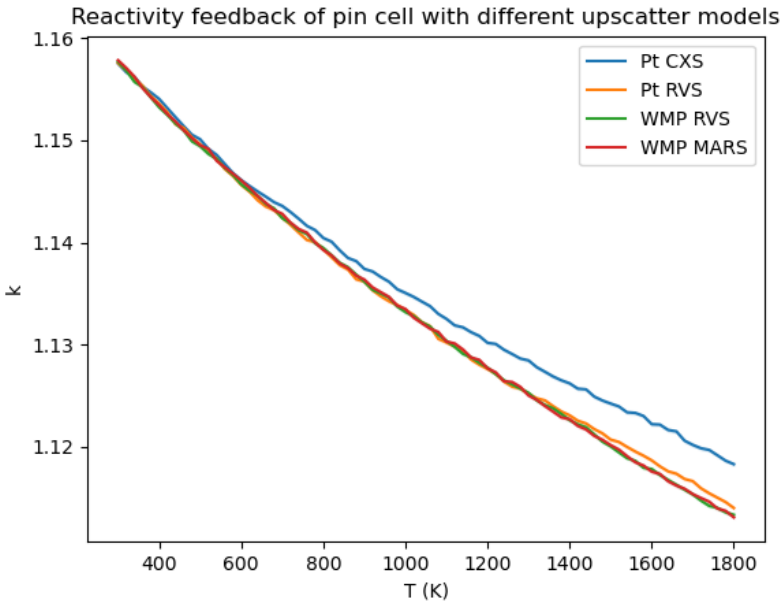


图10：对于富集度为2.4%的新鲜压水堆燃料棒元件问题，MARS所得的 k 本征值与RVS方法的结果相符。线宽表示均值估计标准差。

温度每次递增20开尔文。在两种采用点截面的算例中，一种采用传统的恒定截面自由气体散射近似，另一种采用RVS方法。另两种算例均采用窗化多极截面，其中一种采用RVS方法，另一种采用MARS。使用了ENDFB-VII.1核数据集。图10展示了这些算例的对比。共进行了200个循环，其中10个为非活跃循环，每个循环使用200,000个粒子。因此，每种情况下的 k_{eff} 均收敛至20 pcm。

可以看出，逐点截面表示在1200至1800开尔文之间会产生一定的插值误差。采用多极截面时，MARS方法所得结果与RVS结果相符。因此可以得出结论：对于具有强且间距较大的共振（^{238}U）以及间距较小且较弱的共振（^{235}U）的核素，在共振会影响该温度下双微分截面的整个能量范围内，新方法均能正确发挥作用。

4.4 对追踪速率的影响

最后，为了确定新方法的计算效率，我们在相同的压水堆燃料棒元件示例问题上进行了追踪速率比较。评估了非活跃循环和活跃循环的计算性能。对于活跃循环，采用 100x100 笛卡尔网格，通过径迹长度估计量对通量、裂变率和中子产生率进行计分。此外，还采用了空间均匀化能谱计分，其包含 500 个等宽的 lethargy 分箱。

计算由一台配备六个物理核心的 Intel Xeon W-2133 执行，并得到了表1所示的结果。这清楚地表明，MARS 相较于 RVS 和 DBRC 方法具有更高的计算效率。尽管在这种情况下它的性能不及 RVS，

方法	未激活	活跃
CXS	60.5	11.4
DBRC	57.0	11.1
RVS	58.3	11.3
MARS	60.3	11.1

表1：采用恒定截面处理和三种共振上散射模型得到的追踪速率，单位为千粒子/秒。MARS 的速度与广受认可的技术相当。

但未来的工作将探索其在向量计算机架构上的表现，我们预计它将在这些架构上表现更优。

计数所产生的计算开销往往使新方法对性能的影响尤其微不足道。可以在网格计数中使用碰撞估计器来提高追踪速率，但我们任意选择了径迹长度估计器。由于缓存利用率或分支预测等细微的硬件相关影响，比较活跃循环和非活跃循环时，三种共振上散射处理方法的追踪速率相对性能各不相同。未来的工作将探究多种架构上的详细性能结果。

5 结论

与点截面相比，多极形式具有诸多优势。除了在现代计算架构上有望提升计算效率之外，它还能够在不牺牲库大小的情况下实现精确的多普勒展宽 [23]，、优雅地量化灵敏度，并缩小R矩阵理论与截面表示之间的差距 [13]。本研究为窗化多极形式进一步拓展了一项优势：共振上散射闭式处理。

我们证明，新方法所得结果与其他共振上散射技术的结果一致。为此，我们推导了靶核相对速度分布的表达式，并发现了一种在此应用中普遍出现的新型特殊函数。我们推导、实现并测试了兼顾效率与精度的新型数值技术。结果表明，整体方案能够达到与其他共振上散射建模方法相同的追踪速率。

这种名为多极解析共振散射（MARS）的新方法克服了相对速度制表 [9]，所需的存储开销，并避免了其他常见方法采用的拒绝抽样。由于无需访问中间存储，因此可以减少 GPU 架构上的全局内存访问。此外，该方法也克服了 GPU 上拒绝抽样造成的线程间工作量差异。未来的工作将探索在 GPU 上实现并优化该方法。

6 致谢

本研究部分由美国能源部通过洛斯阿拉莫斯国家实验室提供支持。洛斯阿拉莫斯国家实验室由 Triad National Security, LLC 代表美国能源部国家核安全管理局运营（合同编号：89233218CNA000001）。本研究也部分获得综合大学项目研究生奖学金的支持。本研究还部分获得百亿亿次计算项目（17-SC-20-SC）的支持；该项目由美国能源部科学办公室和国家核安全管理局合作开展。

本出版物中表达的任何观点、研究结果、结论或建议均属于作者所有，不一定反映能源部核能办公室的观点。

参考文献

[1] Milton Abramowitz 和 Irene A. Stegun 编。《数学函数手册：公式、图表和数学表》。0009-修订版。纽约州纽约：多佛出版社，1965年6月1日。1046页。 ISBN: 978-0-486-61272-0。[2] S. M. Abrarov 和 B. M. Quine。“基于指数级数近似的 Voigt/-复误差函数高效算法实现”。载于：《应用数学与计算》218.5（2011年11月1日），第1894–1902页。 ISSN: 0096-3003。DOI: 10.1016/j.amc.2011.06.072。 URL: https://www.sciencedirect.com/science/article/pii/S0096300311009179 （访问于2023年1月10日）。[3] Ren´e A¨ıd、Luciano Campi 和 Nicolas Langren´e。《电力衍生品定价与对冲的结构性风险中性模型》。SSRN学术论文。纽约州罗切斯特，2013年7月1日。 DOI: 10.1111/j.1467-9965.2011.00507.x。 URL: https://papers.ssrn.com/abstract=2276600 （访问于2023年1月1日）。[4] B. Becker、R. Dagan 和 G. Lohnert。“理想气体能量相关散射核随机公式的证明与实现”。载于：《核能年鉴》36.4（2009年5月1日），第470–474页。 ISSN: 0306-4549。 DOI: 10.1016/j.anucene.2008.12.001。 URL: https://www.sciencedirect.com/science/article/pii/S0306454908003186 （访问于2022年10月29日）。[5] Carl M. Bender 和 Steven A. Orszag。《科学家与工程师的高级数学方法 I：渐近方法与微扰理论》。1999年第一版精装本的平装重印版。纽约、海德堡：施普林格，2010年12月1日。607页。ISBN: 978-1-4419-3187-0。[6] Elliott Biondo 等。“无拒绝抽样的自由气体弹性散射算法”。载于：数学与计算方法在核科学与工程中的应用国际会议。北卡罗来纳州罗利，2021年10月1日。 DOI: 10.13182/M&C21-33659。[7] *Cephes*。 URL: https://netlib.org/cephes/（访问于2023年1月3日）。

[8] Namjae Choi 和 Han Gyu Joo。“用于高效处理基于 GPU 的蒙特卡罗中子输运计算中共振散射的相对速度制表法”。载于：《核科学与工程》195.9（2021年9月2日），第954–964页。ISSN: 0029-5639。 DOI: 10.1080/00295639.2021.1887701。 URL: https://doi.org/10.1080/00295639.2021.1887701（访问于2021年9月12日）。[9] Namjae Choi、Kyung Min Kim 和 Han Gyu Joo。“面向基于 GPU 的连续能量蒙特卡罗计算的中子跟踪算法优化”。载于：《核能年鉴》162（2021年11月1日），第108508页。ISSN: 0306-4549。 DOI: 10.1016/j.anucene.2021.108508。 URL: https://www.sciencedirect.com/science/article/pii/S0306454921003844（访问于2021年9月1日）。[10] W. J. Cody、Kathleen A. Paciorek 和 Henry C. Thacher。“Dawson积分的切比雪夫近似”。载于：《计算数学》24.109（1970年），第171–178页。ISSN: 0025-5718。 DOI: 10.2307/2004886。JSTOR: 2004886。[11] R. Dagan。“关于将 S(a,b) 表用于具有显著共振的核素”。载于：《核能年鉴》32.4（2005年3月1日），第367–377页。 ISSN: 0306-4549。 DOI: 10.1016/j.anucene.2004.11.003。 URL: https://www.sciencedirect.com/science/article/pii/S0306454904002191（访问于2021年12月27日）。[12] Alfredo Dea˜no 和 Nico M. Temme。“余误差函数一种推广形式的解析与数值方面”。载于：《应用数学与计算》216.12（2010年8月15日），第3680–3693页。 ISSN: 0096-3003。 DOI: 10.1016/j.amc.2010.05.025。 URL: https://www.sciencedirect.com/science/article/pii/S0096300310005692（访问于2023年1月1日）。[13] Pablo Ducru 等。“R矩阵截面的窗口化多极表示”。载于：《物理评论C》103.6（2021年6月14日），第064610页。 DOI: 10.1103/PhysRevC.103.064610。 URL: https://link.aps.org/doi/10.1103/PhysRevC.103.064610（访问于2021年12月23日）。[14] A.；Magnus F. G.；Harry Bateman Erdelyi。《积分变换表》第1卷。第1版。麦格劳-希尔图书公司，1954年1月1日。411页。[15] Benoit Forget、Jiankai Yu 和 Gavin Ridley。“使用法德耶娃函数的有理分式逼近提高窗化多极形式的性能”。载于：2022年国际反应堆物理会议。宾夕法尼亚州匹兹堡，2022年5月17日。[16] Walter Gautschi。“某些积分的递归计算”。载于：《美国计算机协会期刊》8.1（1961年1月），第21–40页。 ISSN: 0004-5411, 1557-735X。 DOI: 10.1145/321052.321054。 URL: https://dl.acm.org/doi/10.1145/321052.321054（访问于2022年12月23日）。[17] Ely M. Gelbard。《VIM中的超热散射》。FRA-TM-123。美国伊利诺伊州阿贡：阿贡国家实验室，1979年12月5日。[18] S.W.D. Hart 等。“Keno中多普勒展宽拒绝修正的实现”。载于：《美国核学会汇刊》108（2013年1月1日），第423–425页。

[19] J. Humlı̌cek。“复概率函数的高效计算方法：Voigt函数及其导数”。载于《定量光谱学与辐射传递杂志》21.4（1979年4月1日），第309–313页。 ISSN：0022-4073。 DOI：10.1016/0022-4073(79)90062-1。 URL： https://www.sciencedirect.com/science/article/pii/0022407379900621（访问于2023年1月2日）。[20] David G. Hummer。“Dawson函数的切比雪夫多项式级数展开”。载于《计算数学》18.86（1964年），第317–319页。 ISSN：0025-5718。DOI： 10.2307/2003311。JSTOR：2003311。[21] R. N. Hwang。“多能级截面的严格极点表示及其实际应用”。载于《核科学与工程》96.3（1987年7月1日），第192–209页。 ISSN：0029-5639。 DOI：10.13182/NSE87-A16381。 URL： https://doi.org/10.13182/NSE87-A16381 （访问于2023年2月27日）。[22] Steven Johnson。Faddeeva软件包。 URL：http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package （访问于2023年1月10日）。[23] C. Josey等。“用于截面多普勒展宽的窗化多极”。载于《计算物理学杂志》307（2016年2月15日），第715–727页。 ISSN：0021-9991。 DOI：10.1016/j.jcp.2015.08.013。 URL： http://www.sciencedirect.com/science/article/pii/S002199911500532X（访问于2021年1月14日）。[24] Deokjung Lee、Kord Smith和Joel Rhodes。“铀-238共振弹性散射近似对热堆多普勒反应性的影响”。载于《核能年鉴》。PHYSOR 2008，36.3（2009年4月1日），第274–280页。 ISSN：0306-4549。DOI： 10.1016/j.anucene.2008.11.026。 URL：https://www.sciencedirect.com/science/article/pii/S0306454908003083 （访问于2021年12月23日）。[25] F. G. Lether和P. R. Wenston。“Dawson积分的初等近似”。载于《定量光谱学与辐射传递杂志》46.4（1991年10月1日），第343–345页。 ISSN：0022-4073。 DOI： 10.1016/0022-4073(91)90099-C。 URL： https://www.sciencedirect.com/science/article/pii/002240739190099C （访问于2022年11月9日）。[26] Frank G. Lether。“Dawson积分的受约束近极小极大有理逼近”。载于《应用数学与计算》88.2（1997年12月30日），第267–274页。 ISSN：0096-3003。 DOI： 10.1016/S0096-3003(96)00330-X。 URL： https://www.sciencedirect.com/science/article/pii/S009630039600330X（访问于2022年11月8日）。[27] Jingang Liang、Pablo Ducru和Benoit Forget。“利用窗化多极截面数据进行共振弹性散射的靶核速度抽样”。发表于2018年11月11日。[28] Shichang Liu等。“利用向量拟合技术生成窗化多极共振数据”。载于《核能年鉴》112（2018年2月1日），第30–41页。 ISSN：0306-4549。 DOI： 10.1016/j.anucene.2017.09.042。 URL：https://www.sciencedirect.com/science/article/pii/S0306454917303195（访问于2023年2月27日）。

[29] Takamasa MORI和Yasunobu NAGAYA。“MVP连续能量蒙特卡罗代码中新实现的共振弹性散射模型比较”。载于《核科学与技术杂志》46.8（2009年8月1日），第793–798页。 ISSN：0022-3131。 DOI： 10.1080/18811248.2007.9711587。 URL： https://doi.org/10.1080/18811248.2007.9711587（访问于2022年10月28日）。[30] Mohamed Ouisloumen和Richard Sanchez。“一种考虑热扰动效应的重同位素中子散射模型”。载于《核科学与工程》107.3（1991年3月1日），第189–200页。 ISSN：0029-5639。 DOI： 10.13182/NSE89-186。 URL： https://doi.org/10.13182/NSE89-186 （访问于2022年10月28日）。[31] Paul Romano和John Walsh。“用于自由气体弹性散射的改进靶核速度抽样算法”。载于《核能年鉴》114（2018年4月），第318–324页。[32] W. Rothenstein和R. Dagan。“重麦克斯韦气体中子散射的二体动力学处理”。载于《核能年鉴》22.11（1995年11月1日），第723–730页。 ISSN：0306-4549。 DOI： 10.1016/0306-4549(95)00002-A。 URL： https://www.sciencedirect.com/science/article/pii/030645499500002A （访问于2021年12月23日）。[33] Kelly Rowland等。“GPU加速的WARP蒙特卡罗中子输运代码中的Delta追踪”。载于国际数学与计算方法在核科学与工程中的应用会议。韩国济州，2017年4月。 URL： https://www.kns.org/files/int_paper/paper/MC2017_2017_2/P100S02-02RowlandK.pdf （访问于2021年4月27日）。[34] Franz Schreier。“Voigt函数与复误差函数：Huml ̀ıˇcek有理近似的推广”。载于《皇家天文学会月报》479.3（2018年9月21日），第3068–3075页。 ISSN：0035-8711。 DOI： 10.1093/mnras/sty1680。 URL： https://doi.org/10.1093/mnras/sty1680 （访问于2023年1月10日）。[35] Timothy H. Trumbull和Thomas E. Fieno。“将多普勒展宽拒绝修正方法用于低浓铀和混合氧化物燃料栅元耗减计算的影响”。载于《核能年鉴》62（2013年12月1日），第184–194页。 ISSN：0306-4549。DOI： 10.1016/j.anucene.2013.06.013。 URL： https://www.sciencedirect.com/science/article/pii/S0306454913003150 （访问于2022年10月29日）。[36] Tuomas Viitanen和Jaakko Lepp ̈anen。“连续能量蒙特卡罗追踪程序中热运动的显式处理”。载于《核科学与工程》171.2（2012年6月1日），第165–173页。 ISSN：0029-5639。 DOI： 10.13182/NSE11-36。URL： https://doi.org/10.13182/NSE11-36 （访问于2022年7月15日）。[37] Pauli Virtanen等。“SciPy 1.0：Python科学计算的基础算法”。载于《自然·方法》17.3（2020年3月3日），第261–272页。 ISSN：1548-7105。 DOI： 10.1038/s41592-019-0686-2。 URL： https://www.nature.com/articles/s41592-019-0686-2 （访问于2023年1月16日）。

[38] Jonathan A. Walsh、Benoit Forget 和 Kord S. Smith.《自由气体共振弹性散射核的加速采样》。载于：《核能年鉴》69（2014年7月1日），第116–124页。 ISSN：0306-4549。 DOI：10.1016/j.anucene.2014.01.017。URL：https://www.sciencedirect.com/science/article/pii/S0306454914000255（访问于2022年10月31日）。
[39] Andrea Zoia 等.《Tripoli- 4®中的中子弹性散射核多普勒展宽》。载于：《核能年鉴》54（2013年4月1日），第218–226页。 ISSN：0306-4549。DOI： 10.1016/j.anucene.2012.11.023。 URL：https://www.sciencedirect.com/science/article/pii/S0306454912004689 （访问于2021年12月23日）。

式（19）的推导

为简洁起见，并结合核工程期刊的语境，下面的讨论未作严格的数学处理。我们首先定义辅助复函数 F(z)：

$$F(z) = e^{z^2} w(z, x) \tag{68}$$

此时可应用复线积分定理，其中 $\Im[z] > 0$：

$$F(z) - F(0) = \int_0^z \frac{\mathrm{d}F}{\mathrm{d}z}|_{z=z'} \, \mathrm{d}z' \tag{69}$$

计算 $\mathrm{d}F \, \mathrm{d}z$ 并代入后可得：

$$e^{z^2} w(z, x) - w(0, x) = \int_0^z \left(2z' e^{z'^2} w(z', x) - e^{z'^2} \frac{i}{\pi} \int_{-\infty}^{x} \frac{e^{-t^2} \, \mathrm{d}t}{(z' - t)^2} \right) \mathrm{d}z' \tag{70}$$

这里用到了积分的线性，并且还交换了微分与积分的顺序。现在可以精确计算最内层的积分，并将 z' 代入定义不完全法德耶娃函数的积分中。由此得到：

$$e^{z^2} w(z, x) - w(0, x) = \frac{i}{\pi} \left(\int_0^z \frac{e^{z'^2} e^{-x^2}}{x - z'} \, \mathrm{d}z' - \sqrt{\pi}(1 + \mathrm{erf}(x)) \int_0^z e^{z'^2} \, \mathrm{d}z' \right) \tag{71}$$

回顾一下，法德耶娃函数可定义为

$$w(z) = e^{-z^2} \left(1 + \frac{2i}{\sqrt{\pi}} \int_0^z e^{t^2} \, \mathrm{d}t \right) \quad , \tag{72}$$

我们可以将 $w(z)$ 识别为式 71 的末项。项 $w(0, \ x)$ 必须按柯西主值来解释，因此会产生一个赫维赛德函数形式的贡献。于是得到以下表达式：

$$w(z, x) = e^{-z^2} \left(-\frac{1}{2}(\mathrm{Ei}(-x^2) + \frac{i}{\pi} e^{-x^2} \int_0^z \frac{e^{z'^2}}{x - z'} \, \mathrm{d}z' \right) + \frac{1}{2}(1 + \mathrm{erf}(x))(w(z) - e^{-z^2}) + h(x) e^{-z^2} \quad . \tag{73}$$

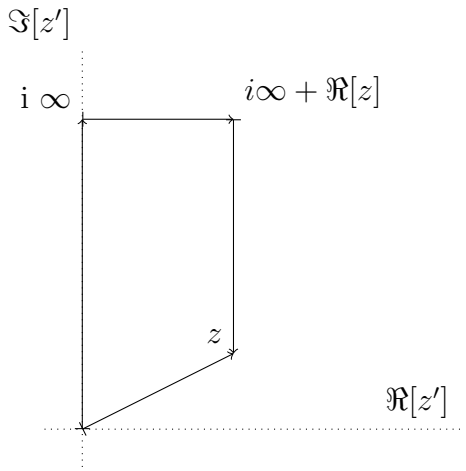


图11：用于抵消公式 73 中指数积分的修正围道。顶部线段的贡献为零。

这一结果或许可用于对 $w(z,\ x)$ 进行数值计算。不过，它有一个缺点：当 $x = 0$ 时，指数积分项趋于无穷大，而这一发散会被积分项抵消。然而， $w(z,\ x)$ 在 $x = 0$ 处有定义；若在数值例程中加入分支逻辑来处理这种情况，则会很繁琐。经过进一步简化，可以使该表达式更适合进行数值近似。

积分项的积分区间为 0 到 z；只要满足 $x \neq 0$，被积函数所围区域内就不包含下述路径的任何极点。因此，我们采用改变积分路径的方法：图11中的围道可以方便地抵消各项。由于e^{t^2} 项，围道的顶部线段积分为零，因此：

$$\int_0^z \frac{e^{t^2}}{x-t}\,\mathrm{d}t = \int_0^{i\infty} \frac{e^{t^2}}{x-t}\,\mathrm{d}t - \int_z^{i\infty} \frac{e^{t^2}}{x-t}\,\mathrm{d}t \tag{74}$$

经过一些代数运算可得：

$$\int_0^{i\infty} \frac{e^{t^2}}{x-t}\,\mathrm{d}t = \frac{1}{2}e^{x^2}\left(-i\pi(\mathrm{erf}(x) - \mathrm{sign}(x)) + \mathrm{Ei}(-x^2)\right) \tag{75}$$

将公式 74 代回公式 73 后，符号函数项最终会抵消海维赛德项。公式 74 中的第二项可以通过变量替换 $t' = -it$ 轻易转换为公式 19 中的积分，从而得到公式 19。

B 公式46的推导

目标是找到最小的 n ，使得

$$\frac{x^n E_1(x)}{n! E_{n+1}(x)} < 1 \tag{76}$$

公式 76 左侧函数随 n 增大时的变化情况（x 为常数）已证明如下：该函数先从 1 开始增大，达到最大值后便单调递减，且始终不会变为负值 [16]。首先，用等价的上不完全伽马函数替换指数积分：

$$E_n(x) = x^{n-1}\Gamma(1-n, x) \tag{77}$$

因此方程变为：

$$\Gamma(0, x) = n!\Gamma(-n, x) \tag{78}$$

利用上不完全伽马函数 $\Gamma(s,\ x) \to x^{s-1}e^{-x}$的渐近公式，可得以下近似方程：

$$x^n \approx n! \tag{79}$$

先代入斯特林公式，即可近似求解：

$$x^n \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{80}$$

取 n次方根后得到

$$ex \approx (2\pi n)^{\frac{1}{2n}} n \tag{81}$$

对于较大的n，可通过展开指数函数来近似右侧的第二项：

$$ex \approx (1 + \frac{1}{2n}\log(2\pi n))n \tag{82}$$

该方程可用朗伯 W 函数精确求解：

$$n \approx \frac{1}{2}W\left(\frac{e^{2ex}}{\pi}\right) \tag{83}$$

随后利用朗伯 W 函数的渐近性质 $W(z) \approx \log\ z - \log\log\ z$，得到公式 46。

C R（m，z）的渐近近似（公式 30）

通过对被积函数进行分组，可以得到一个高效的近似：

$$R(m,z) = \int_0^\infty f^{(0)}(t,m)\sin(2zt)\,\mathrm{d}t = -\frac{1}{2z}\int_0^\infty f^{(0)}(t,m)\frac{\partial}{\partial t}\left[\cos(2zt)\right]\mathrm{d}t \tag{84}$$

其中

$$f^{(0)}(t,m) = \frac{e^{-t^2}}{t^2+m^2} \quad .$$

对公式 84 中最右侧的表达式反复进行分部积分，可得如下形式的发散级数近似：

$$R(m,z) = \frac{f^{(0)}(0,m)}{2z} + \frac{f^{(2)}(0,m)}{8z^3} + \frac{f^{(4)}(0,m)}{32z^5} + \frac{f^{(6)}(0,m)}{128z^7} + \cdots \tag{85}$$

其中， $f^{(n)}(t,m)$ 表示 $f_{(0)}(m,t)$ 关于 t 的第 n 阶导数。以 $t=0$ 为中心求得的一些后续值为:

$$f^{(2)}(0,m) = -\frac{2(1+m^2)}{m^4} \tag{86}$$

$$f^{(4)}(0,m) = \frac{12(2+2m^2+m^4)}{m^6} \tag{87}$$

$$f^{(6)}(0,m) = -\frac{120(6+6m^2+3m^4+m^6)}{m^8} \tag{88}$$

$$f^{(8)}(0,m) = \frac{1680(24+24m^2+12m^4+4m^6+m^8)}{m^{10}} \tag{89}$$

这些表达式看似毫无规律，但仔细观察许久后，可以得到一个简单的递归公式来计算这些值：适用于计算机实现。考虑由下式定义的数列 $a_n,\ c_n \in \mathbb{R}$

$$a_0 = m^{-2}; \quad c_0 = 2 \tag{90}$$

and

$$a_{n+1} = \frac{2n(2n-1)a_n + c_n}{m^2}; \quad c_{n+1} = (4n+2)c_n \quad . \tag{91}$$

利用这一点，可以证明 $f^{(2n)}(0,\ m) = a_n$。这样便可轻松求得公式 85 的渐近级数。数值实验表明，这是一个非常精确的近似：当 $|z|=5$ 和 $|m|=1$ 时，仅保留五项的最大误差约为 10^{-6}。随着 $|z|\to\infty$ 和 $|m|\to\infty$，误差会迅速减小。

D 公式 47 中 $J(z,\ x)$ 的渐近近似

与 $R(m,\ z)$ 积分的渐近近似相比，据我们所知，目前没有适合编写简洁计算机代码的简明表达式。因此，求得渐近表达式需要借助计算机代数系统。首先，对公式 47 进行简单的变量替换，得到:

$$J(z,x) = \int_0^{\Im[z]} \frac{e^{-t^2} e^{2i\Re[z]t}\,\mathrm{d}t}{i(\Re[z]-x)-t} \tag{92}$$

由此可清楚看出，展开指数项所产生的数值困难源于 $e^{2i\Re[z]t}$ 调制。随着被积函数的振荡越来越强，必须将这一项分离出来，才能获得正确的渐近行为。随后即可应用标准的反复分部积分步骤。这纯粹是繁琐的工作，因此我们仅在下方给出计算五项的 C++ 代码。

```
const std::complex<double> pp1 = 6.0 + m2*(6.0 + 3.0*
m2) + z r *(m*( -3.0 - 3.0*m2) + z r *(m2*(2.0 + 2.0*m2) + z
r *( -2.0*m2*m + 4.0*m2*m2*z r ))) +
```

```
    zi*(zr*(3.0*ii +m2*(3.0*ii - 6.0*ii*m2) +zr*(m*(-4.
0*ii - 4.0*ii*m2) +zr*(m2*(6.0*ii - 4.0*ii*m2) - 16.0*i
i*m2*m*zr))) +zi*(6.0 +m2*(6.0- 12*m2) +zr*(m*(-3.
0 - 18.0*m2) +zr*(-2 - 4.0*m2*m2 +zr*(m*(6.0- 16.0*m2)
 - 24.0*m2*zr))) +zi*(40.0*ii*m2*m +zr*(3.0*ii+m2*(1
8.0*ii + 4.0*ii*m2) +zr*(m*(-4.0*ii + 16.0*ii*m2) +zr*(
-2.0*ii + 24.0*ii*m2 + 16.0*ii*m*zr))) +zi*(3.0 +m2*(4
8.0 +4.0*m2) +zi*(m*(-24.0*ii - 16.0*ii*m2) +zi*(-4.
0 - 24.0*m2 +zi*(16.0*ii*m + 4.0*zi + 4.0*ii*zr) +(-16.
0*m - 4.0*zr)*zr) +zr*(-24.0*ii*m2 +(-16.0*ii*m - 4.0*i
i*zr)*zr)) +zr*(m*(6.0 +16.0*m2) +zr*(-2.0 + 24.0*m2 +
zr*(16.0*m + 4.0*zr))))))); const double pp2 =(-6.0 +
m2*(-6.0 - 3.0*m2) +zr*(m*(3.0 + 3.0*m2) +zr*(m2*(-2.0 -
2.0*m2) +zr*(2.0*m2*m - 4.0*m2*m2*zr))))/(m2*m2*m);
```

// 注释: std::exp(zi*(zi - 2.0*ii*zr)) =exp(-z^2)/exp(-zr^2) 结果 =

(pp2 +pp1/(cache.emz2/cache.emrz2 *std::pow(m -ii*zi,5)))/(8.*std::pow(zr,5));

E 求根 BootstrapFunction

```
double rootfinding_bootstrap_guess(double xi,
        double apprx0cdf, // x 处的近似累积分布函数=0double
        dcdx, // x 处的近似概率密度函数=0double jump, // 共振处
        的概率跃变 std::complex<double>z) {

  // 注释: 可近似长度为 e^{-z^2} * 3Im [z ]double yjumplo
   = 0.5 *(std::erf(z.real() -1.5 *z.imag()) + 1.
  0); double yjumphi = 0.5 *(std::erf(z.real() +1.
  5 *z.imag()) + 1.0);

  若(xi <=apprx_0cdf) {若(yjumphi > 0.
  5 && yjumplo < 0.5){
```

跳转 *=(0.5 -yjumplo)/(yjumphi -yjumplo);
yjumphi = 0.5;} 否则若(yjumplo > 0.5){yjumplo
= 0.0; yjumphi = 0.0; 跳转 = 0.0;}

如果（跳跃 >appr x0 cdf）跳跃 =appr x0 cdf; 常量双精度 d = yjumphi -yjumplo; 常量双精度 sout =（appr x0 cdf -jump）/（0.5 -d）; 常量双精度 inv =jump > 0.0 ? d/jump : 0.0; 如果（xi >=sout *yjumplo +jump） {常量自动 r =sout *yjumplo +jump; 常量自动 a =（r -dcdx *（yjumphi - 0.5） -appr x0 cdf）/std::pow（yjumphi - 0.5, 2）; 返回 0.5 *（-dcdx +std::sqrt（std::pow（dcdx, 2） -4.0 *（appr x0 cdf -xi） *a））/a + 0.5; } 否则如果（xi >sout_ *yjumplo） {返回（xi -sout *yjumplo） *sinv +yjumplo; }否则{如果（sout > 0.0）返回 xi/sout; 否则返回 0.5 *yjumplo; }} 否则{// xi >appr x0 cdf 如果（yjumplo < 0.5 && yjumphi > 0.5）{jump *=（yjumphi - 0.5）/（yjumphi -yjumplo）; yjumplo = 0.5; }否则如果（yjumphi < 0.5）{yjumplo = 1.0; yjumphi = 1.0; jump = 0.0; }

// 若跳变量较大，则适当裁剪 if(jump > 1.0 -approx 0 cdf) jump = 1.0 -approx 0 cdf; const auto dyjumphi -yjumplo; const auto sout =(1.0 -jump -approx 0 cdf)/(0.5 -d); const auto sinv =jump > 0.0 ? d/jump : 0.0; const auto thresh 1 =sout *(yjumplo - 0.5) +jump +approx 0 cdf; const auto thresh 2 =sout *(yjumplo - 0.5) +approx 0 cdf;

如果 (xi >=阈值1) 返回 (xi -阈值1) / 输出 +yjumphi; 否则如果 (xi >输出 *(yjumplo - 0.5) +近似0cdf)

返回 (xi − 阈值2) ∗ 反正弦 + yjumplo；否则{常量自动 a =(阈值 2 − dcdx ∗(yjumplo − 0.5) − 近似0cdf) / std::pow(yjumplo − 0.5, 2)；返回 0.5 ∗(− dcdx + std::sqrt(std::pow(dcdx, 2) − 4.0 ∗(近似0cdf − xi) ∗ a)) / a + 0.5;}}不可达();}