Vlasov–Poisson 方程的数值流迭代

Matthias Kirchhart* R. Paul Wilhelm*

摘要

我们介绍用于求解 Vlasov–Poisson 方程的数值 fl流迭代（NuFI）。在本文稍后所述的某种意义上，NuFI 能够以无限 分辨率表示分布函数。NuFI 精确保留正性、所有 L^p-范数、电荷和熵。数值实验表明，该方法不存在能量漂移。NuFI 速度快，所需内存比传统方法少几个数量级，而且能够在 GPU 集群上高效并行化。低 fl保真度模拟在较长时间内也能给出良好的定性结果，并可在低成本工作站上运行。

1 引言

1.1 Vlasov–Poisson 方程

我们所关注的问题是一个用于描述等离子体在无碰撞极限下演化的简化模型，例如核聚变装置中的等离子体。在无量纲形式下，它由以下系统给出：

$$\partial_t f + v \cdot \nabla_x f - E \cdot \nabla_v f = 0, \tag{1}$$

$$E \coloneqq -\nabla_x \varphi, \tag{2}$$

$$-\Delta_x \varphi = \rho, \tag{3}$$

$$\rho(t,x) \coloneqq \bar{\rho} - \int_{\mathbb{R}^d} f(t,x,v)\,\mathrm{d}v. \tag{4}$$

这里，$f = f(t,\ x,\ v)$ 是电子分布函数，即，$f(t,\ x,\ v) \geq 0$描述了在时刻 $t \in \mathbb{R}$，具有速度 $v \in \mathbb{R}^d$ 且位于位置 $x \in \mathbb{R}^d$ 的电子的概率密度。等离子体的 Vlasov–Poisson 方程通常针对 $d \in \{1,\ 2,\ 3\}$ 进行研究。在本文中，我们假设 x 上满足周期性边界条件，即 $\Omega \coloneqq [0,L]^d$ with $L > 0$，使得对于所有笛卡尔基向量 $e_i \in \mathbb{R}^d$ 和所有 $k \in \mathbb{Z}$，都有 $f(t,\ x + Lke_i,\ v) = f(t,\ x,\ v)$。

为了获得适定问题，需要提供 f 的非负初始数据，通常取在时刻$t = 0$:

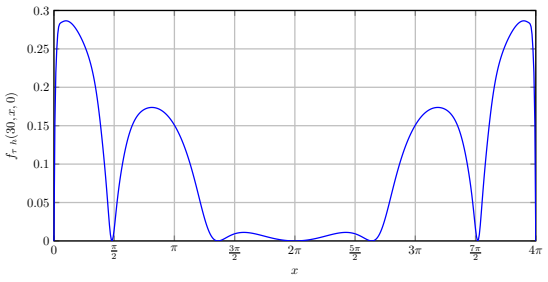
$$f(t=0,x,v) = f_0(x,v) \text{ for a given } x\text{-periodic } f_0 \in L^1(\Omega \times \mathbb{R}^d) \cap L^\infty(\mathbb{R}^d \times \mathbb{R}^d),\ f_0 \geq 0 \text{ a. e.} \tag{5}$$

在许多计算基准测试中，f_0 是一个光滑函数，通常表示为相对简单的表达式。通常，f_0 会在 $|v| \to \infty$时呈指数衰减，因此在计算实践中，只要 $|v| \geq v_{\max}$对于某个用户定义的参数 $v_{\max}$成立，便假定$f(t,\ x,\ v) = 0$。

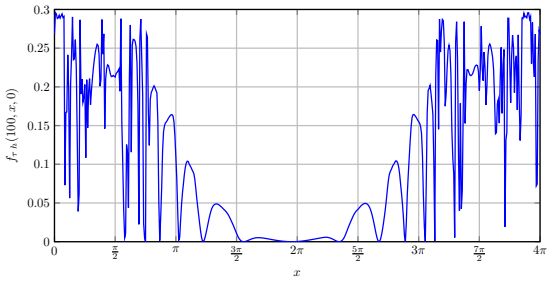
方程（4）定义了电荷密度 ρ；参数 $\bar{\rho}$ 源于均匀离子背景这一假设，需要对其进行选择，以保持总体电中性，即 $\forall t \geq 0$：$\int_\Omega \rho(t,\ x)\,\mathrm{d}x = 0$。

忽略碰撞和磁 fl场后，Vlasov 方程（1）描述了 f在自洽电fl场 fl的作用下的演化，电场 $E = E$（t，x）由电势$\varphi = \varphi$（$t,\ x$）给出，而电势则是泊松方程（3）的解。

*App应用与计算数学，亚琛工业大学，Schinkelstraße 2，52062 亚琛，德国
E-邮箱：kirchhart@acom.rwth-aachen.de 和wilhelm@acom.rwth-aachen.de



$t = 30$



$t = 100$

图1：截面 $v = 0$ 处的分布函数 $f(t,\ x,\ v)$ 的近似结果，展示的是 $d = 1$中的双流不稳定性基准测试，见第 4.2 节，对应时刻 $t = 30$ 和 $t = 100$。使用下文所述的数值 fl流迭代计算，采用 $N_x = 64$ 和 $N_v = 256$。可以清楚地看到 fl丝状结构不断增长，且随时间推移变得更加 fl细。由于分辨率有 fl限，所有直接离散 f 的基于网格的方法最终都会无法准确重现这些 fl丝状结构。

1.2 C Vlasov–Poisson 方程数值算法面临的挑战

表面fl上，数值求解 Vlasov–Poisson 方程似乎很简单：暂且假设电 fl场 E 已知，Vlasov 方程（1）是一个线性平流方程，可以用经典数值格式求解，例如采用迎风离散的 fl有限差ff分法。通过这种近似可以直接计算电荷密度 ρ；求解泊松方程（3）也是一个经典问题，有许多高fl效算法可供使用。然后可将得到的电势近似值反馈给 Vlasov 求解器，向前推进一个时间步长。

虽然理论上可行，但这种简单方法在实践中会遇到许多问题。直接离散化 f 的基于网格和基于粒子的方法会遭遇维数灾难： $2d$ 维的（$x,\ v$）空间会带来极大的内存需求。对于这类方案，模拟三维情况（$d = 3$）通常只有在超大型高性能计算机上才可行，即使如此也未必能实现。同时，这类方案的算术强度通常很低，也就是说，每字节浮点运算数很少。现代计算机体系结构难以为这类方案提供良好的性能。

即使能够克服这些问题，我们仍想特别强调由于所谓的fl丝*fl*状结构，即解中随时间演化而形成的非常 fl细微结构。这一点如图1所示，其中展示了一个知名基准问题的数值解在两个不ff同时间 t的截面。解中不断增加的 fl丝状结构表现为振荡，其频率会随时间迅速增加 t。当这些 fl丝状结构变得对于给定分辨率而言过于细小时，传统离散化往往开始违反第 2.5 节中讨论的若干守恒定律。从这个意义上说，这些格式会产生不符合物理规律的结果。例如，对于许多高阶格式而言，避免 f 在 fl丝状结构附近出现fl过冲和负值非常困难。

1.3 概述

本文其余部分的结构如下。 第 2 节 介绍数值 fl流迭代，并讨论数学和实际问题。 第 3 节 阐述其与文献中其他方法之间的关系。数值实验将在第 4 节中讨论。最后，我们将在第 5 节中展望未来可能的扩展。

2 数值流迭代

要理解 NuFI，我们首先需要介绍与 Vlasov–Poisson 方程相关的精确流。接下来，我们将介绍 NuFI 的组成部分，并讨论若干重要的守恒性质。

2.1 Vlasov–Poisson 方程的精确流与解

假设我们知道某个虚构“粒子”在时间 $s \in \mathrm{R}$ 时的速度 $v \in \mathrm{R}^d$ 和位置 $x \in \mathrm{R}^d$。如果进一步假设所有时刻 t 的电场 field E 都已知，那么我们可以通过求解以下初值问题，追踪该虚构粒子随时间向前（$t > s$）和向后（$t < s$）演化的状态 $\big(\hat{x}(t), \hat{v}(t)\big)$：

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\hat{x}(t) &= \hat{v}(t), & \hat{x}(s) &= x, \\ \frac{\mathrm{d}}{\mathrm{d}t}\hat{v}(t) &= -E\big(t, \hat{x}(t)\big), & \hat{v}(s) &= v. \end{aligned} \tag{6}$$

方程（6）是fiVlasov方程（1）特征线的定义。假设 E 足够fiu光滑，根据 Picard–Lindel¨of 定理，众所周知，对于任意的（s, x, v）$\in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$。由此我们得到以下定义fi:

定fi义 2.1（流）。*Vlasov–Poisson* 方程的 *fl*流 $\Phi_s^t(x, v)$ 定义为初值问题（(6)）的唯一解:

$$\Phi_s^t(x, v) \coloneqq \big(\hat{x}(t), \hat{v}(t)\big).$$

We w有时会使用精确 fi流 这一术语，以便 Φ_s^t 与近似的数值 *fl*流相区别 $s.$

换句话说： $\Phi_s^t(x, v)$ 告诉我们某个时刻 t 处于给定状态（x, v）的粒子在时刻 s的位置和速度。注意，系统（6）是辛的，而下面的引理是一个众所周知的推论，之后将发挥重要作用。

引理 2.2（精确流的体积保持性）。该 *fl*流 Φ_s^t： $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ 是一个保持相空间体积的微分同胚:

$$\forall s, t \in \mathbb{R}: \ \forall (x, v) \in \mathbb{R}^d \times \mathbb{R}^d: \qquad \det \nabla_{(x,v)} \Phi_s^t(x, v) \equiv 1.$$

借助该 fi流 Φ_s^t，Vlasov–Poisson 系统的精确解具有非常简单的形式。

引理 2.3（解公式）。设非负分布函数 f 在初始时刻 $t = 0$给定，即假设已知 f_0（x、v）≥ 0，满足 f（0、x、v）$= f_0$（x、v）$= f_0$对所有 （x、v）$\in \mathbb{R}^d \times \mathbb{R}^d$成立。那么，*Vlasov–Poisson* 方程的（精确）解为:

$$\forall (t, x, v) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d: \qquad f(t, x, v) = f_0\big(\Phi_t^0(x, v)\big).$$

2.2 数值流与 Vlasov–Poisson 方程的解 n

在 NuFI 中，精确数值 fi流 Φ 被数值近似 Ψ所替代。暂时我们仍假设电 fi场 E 在所有时刻 t均为已知。为简化说明，我们将范围限制在 $\Phi_t^0 \approx \Psi_{n\tau}^0$、 $n \in \mathrm{N}_0$, 的情形，即只考虑离散时间 $t = n\tau \geq 0$，其时间步长 $\tau > 0$由用户定义。不过，扩展到任意 t 当然也是可能的。

由于系统 (6) 具有辛性，因此很自然地使用常微分方程的 Φ_t^0 辛单步方法来近似ff。本文将采用著名的 St¨ormer–Verlet 方法。[1] 注意到我们是在沿时间反向推进，用数学符号表示为:

$$\Psi_{n\tau}^0 = \Psi_\tau^0 \circ \Psi_{2\tau}^\tau \circ \Psi_{3\tau}^{2\tau} \circ \Psi_{4\tau}^{3\tau} \circ \cdots \circ \Psi_{n\tau}^{(n-1)\tau}, \tag{7}$$

其中，单步 $\Psi_{k\tau}^{(k-1)\tau}(x, v)$, $k \in \{1, \ldots, n\}$，定义如下:

$$\begin{aligned} v_{k-\frac{1}{2}} &\coloneqq v + \tfrac{\tau}{2} E(k\tau, x), \\ x_{k-1} &\coloneqq x - \tau v_{k-\frac{1}{2}}, \\ v_{k-1} &\coloneqq v_{k-\frac{1}{2}} + \tfrac{\tau}{2} E\big((k-1)\tau, x_{k-1}\big), \\ \Psi_{k\tau}^{(k-1)\tau}(x, v) &\coloneqq (x_{k-1}, v_{k-1}). \end{aligned} \tag{8}$$

我们认为，值得指出的是， $\Psi_{n\tau}^0$ 可以轻松高效地在计算机硬件上实现，每个时间步长只需对 E 求值一次。在下面的 C++ 代码片段中， `vec`表示由 d个浮点值组成的向量。

```
void numerical_flow( size_t n, double tau, vec &x, vec &v )
{
    if ( n == 0 ) return;

    // Omit the following single line for Psi_tilda:
    v += (tau/2)*E(n*tau,x);

    while ( --n )
    {
        x -= tau * v;           // Inverse signs; we are
        v += tau * E(n*tau,x); // going backwards in time!
    }

    x -= tau*v;
    v += (tau/2)*E(0,x);
}
```

此代码还提到了一个密切相关的函数 $\tilde{\Psi}^0_{n\tau}$，它定义为fi满足 $\Psi^0_{n\tau}(x,\ v) = \tilde{\Psi}^0_{n\tau}\big(x,\ v + \frac{\tau}{2}E(n\tau,\ x)\big)$，稍后我们会用到它。

St¨ormer–Verlet 方法在时间上具有二阶精度：$|\Phi^0_{n\tau}(x,\ v) - \Psi^0_{n\tau}(x,\ v)| = O(\tau^2)$。此外，该方法是对称的，即沿时间反向运行该方法，等同于对时间反演后的 Vlasov–Poisson 方程运行正向方法。数值 fl流构造完成后，自然可以定义如下形式的近似：

$$\boxed{f(n\tau, x, v) = f_0\big(\Phi^0_{n\tau}(x,v)\big) \approx f_0\big(\Psi^0_{n\tau}(x,v)\big) =: f_\tau(n\tau, x, v).} \tag{9}$$

以下几点值得注意：

- Both $\Psi^0_{n\tau}$ and $\tilde{\Psi}^0_{n\tau}$ 均可在任意位置 $(x,\ v) \in \mathrm{R}^d \times \mathrm{R}^d$处计算。St¨ormer–Verlet 方法仅对时间进行离散化，数值 fl流的 $(x,\ v)$ 分辨率不受限制。正是在这个意义上，我们说 NuFI 中的分布函数f 在相空间中具有无*fi*限分辨率 。
- 无需显式存储 f，只需能够访问初始数据 f_0 以及电 fi场 E 。
- 为了计算 $\Psi^0_{n\tau}$，只需在离散时间步 $i\tau$、 $i = 0$、 …、 n处获取电 fi场 E。
- 为了计算 $\tilde{\Psi}^0_{n\tau}$，只需在之前的时刻 $i\tau$、 $i = 0$、 …、 $n-1$获取电 fi场，即当前的 $E(n\tau$、·)并不需要。

在实际应用中，精确的电 fi场 $E(t$、 $x)$将由数值近似$E_{\tau,\,h} \approx E$替代； h 的含义将在稍后说明。由此得到的完全离散数值 fi流$\Psi^0_{n\tau,\,h} \approx \Psi^0_{n\tau}$ 也将通过附加指标 h加以区分。然而，无论$E_{\tau,\,h}$存在何种误差，与精确fi流 $\Phi^0_{n\tau}$一样，辛积分器的显著关键性质之一是，数值 fi流 $\Psi^0_{n\tau}$及相应的$\Psi^0_{n\tau,\,h}$ 也保持相空间中的体积不变。为完整起见，我们重复这一简单证明。

引理2.4（数值流的体积保持）。无论电场*field* $E_{\tau,h} \approx E$中存在何种误差，*St¨ormer–Verlet* 方法的数值 *flows* $\Psi^0_{n\tau}$ *and* $\Psi^0_{n\tau,h}$ 都是微分同胚，并且保持相空间中的体积：

$$\forall n \in \mathbb{N}_0:\ \forall (x,v) \in \mathbb{R}^d \times \mathbb{R}^d:\qquad \det \nabla_{(x,v)} \Psi^0_{n\tau}(x,v) \equiv \det \nabla_{(x,v)} \Psi^0_{n\tau,h}(x,v) \equiv 1.$$

证明。我们仅说明 $\Psi^0_{n\tau}$ 的证明，因为$\Psi^0_{n\tau,h}$ 的证明完全相同。根据链式法则和行列式的性质，有：

$$\det \nabla_{(x,v)} \Psi^0_{n\tau} = \prod_{k=1}^{n} \det \nabla_{(x,v)} \Psi^{(k-1)\tau}_{k\tau}. \tag{10}$$

同样，一个步骤 $\Psi^{(k-1)\tau}_{k\tau}$ 由式（8）给出的三个子步骤组成。例如，第 fi一个子步骤的雅可比矩阵为：

$$\nabla_{(x,v)} \begin{pmatrix} x \\ v + \frac{\tau}{2}E(k\tau, x) \end{pmatrix} = \begin{pmatrix} I & 0 \\ \frac{\tau}{2}\nabla_x E(k\tau, x) & I \end{pmatrix}. \tag{11}$$

这是一个三对角线为单位矩阵的角矩阵，因此其行列式也是 1。对于其他子步骤也是如此，因此 $\det \nabla_{(x,v)} \Psi^0_{n\tau} \equiv 1$。 □

2.3 数值流迭代的结构

到目前为止，我们一直假设电场 E 是已知的。然而在实际应用中，情况当然并非如此，需要根据电荷密度 ρ 计算电场；而电荷密度本身定义于（4）中。因此，实际只能得到某种近似 $E_{\tau,h} \approx E$ 其中 h 表示一个离散化参数，稍后将对此进行说明。

在初始时刻 $t = 0$，电荷密度 $\rho(t = 0, x)$ 可以通过沿 f_0 对 v 进行积分来计算。例如，可以使用数值求积得到近似值 $\rho_{\tau,h}(0, x) \approx \rho(0, x)$。然后，通过求解泊松方程（3）得到近似电场 $E_{\tau,h}(0, x)$；泊松方程本身则可以采用某种合理的数值方法进行离散化。

从这里开始，我们按时间逐步迭代。因此，假设我们已在时刻 $i\tau$, $i = 0, \dots, n-1$ 计算出 $\rho_{\tau,h}$, $\varphi_{\tau,h}$ 和 $E_{\tau,h}$，其中 $n \in$ 为某个 N。我们希望使用数值近似 $f(n\tau, x, v) \approx f_0\big(\Psi^0_{n\tau,h}(x, v)\big)$ 来计算 $\rho_{\tau,h}(n\tau, x)$。然而，由于 $E_{\tau,h}(n\tau, x)$ 仍未知，无法求出 $\Psi^0_{n\tau,h}$。幸运的是，可以借助 $\tilde{\Psi}^0_{n\tau,h}$ 和简单的变量变换来解决这一循环依赖：

$$\begin{aligned}\rho(n\tau, x) &= \bar{\rho} - \int_{\mathbb{R}^d} f_0\big(\Phi^0_{n\tau}(x, v)\big)\,\mathrm{d}v \\ &\overset{\Phi \rightsquigarrow \Psi_{\tau,h}}{\approx} \bar{\rho} - \int_{\mathbb{R}^d} f_0\big(\Psi^0_{n\tau,h}(x, v)\big)\,\mathrm{d}v = \bar{\rho} - \int_{\mathbb{R}^d} f_0\big(\tilde{\Psi}^0_{n\tau,h}(x, v + \tfrac{\tau}{2} E_{\tau,h}(n\tau, x))\big)\,\mathrm{d}v \\ &\overset{\hat{v} := v + \frac{\tau}{2} E_{\tau,h}(n\tau,x)}{=} \bar{\rho} - \int_{\mathbb{R}^d} f_0\big(\tilde{\Psi}^0_{n\tau,h}(x, \hat{v})\big)\,\mathrm{d}\hat{v}. \quad (12)\end{aligned}$$

换言之，在计算 ρ 时，我们可以放心地将 $\Psi^0_{n\tau,h}$ 替换为 $\tilde{\Psi}^0_{n\tau,h}$，而不会引入任何额外误差。此外，与 $\Psi^0_{n\tau,h}$ 不同，它的同伴 $\tilde{\Psi}^0_{n\tau,h}$ 可以计算！因此，用合适的求积公式替换最后一个积分后，我们便可计算 $\rho_{\tau,h}(n\tau, x) \approx \rho(n\tau, x)$ 并继续进行。这就是数值流迭代，可概括如下：

1. 沿 v 对 f_0 积分，并使用数值求积计算 $\rho_{\tau,h}(t = 0, x)$。
2. 使用 $\rho_{\tau,h}(t = 0, x)$ 计算并存储 $\varphi_{\tau,h}(t = 0, x)$。
3. 对于 $n = 1, 2, 3, \dots$：

   a) 使用 (12) 和数值求积计算近似值 $\rho_{\tau,h}(n\tau, x) \approx \rho(n\tau, x)$。b) 使用 $\rho_{\tau,h}(n\tau, x)$ 计算并存储 $\varphi_{\tau,h}(n\tau, x)$。

在极限 $h \to 0$ 下，(12) 中的最后一个积分可以无误差地计算，泊松方程 $-\Delta\varphi_{h,0} = \rho_{h,0}$ 也可以精确求解。

定义finition 2.5（半离散近似）。半离散近似 $f_{\tau,0}$ 定义为

$$f_{\tau,0}(n\tau, x, v) := f_0\big(\Psi^0_{n\tau,0}(x, v)\big),$$

其中在每个个时间步长中，（12）中的最后一个积分以及泊松方程 $-\Delta\varphi_{\tau,0} = \rho_{\tau,0}$ 均被精确求解 *tly.*

注意，除了初始时刻 $t = 0$之外，由于时间离散引入的误差，我们通常仍有$\Psi^0_{n\tau} \neq \Psi^0_{n\tau,0}$、$E \neq E_{\tau,0}$、$\varphi \neq \varphi_{\tau,0}$，以及$\rho \neq \rho_{\tau,0}$。因此，我们通常也有$f_{\tau,0} \neq f_\tau$，其中 f_τ 来自（9），该式使用了精确的 field E。

2.4 ρ 和 φ的数值近似

虽然分布函数 f 会随时间增加振荡，但数值证据表明，电势 φ 和场 E通常并非如此，电荷密度 ρ 在一定程度上也同样如此。图示见 图 2。

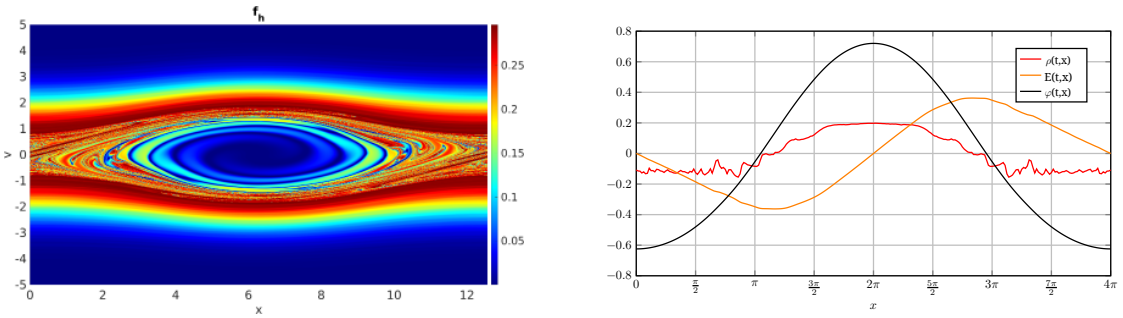


图 2：即使分布非常强烈地 fi起伏 f，电场 fi和 E 电势 φ 通常仍相对平滑。此例展示了双流不稳定性基准测试（ f 左图）的 ρ、 E以及 φ右图）的近似结果（$d = 1$）在 $t = 100$。更多详情请参见 第 4.2 节 。

因此，与 f 不同，电荷密度 ρ 和尤其是电势 φ 可以在相对粗糙的网格上进行高效近似。在本研究中，我们在网格宽度为 $h_x = \frac{L}{N_x}$ 的笛卡尔网格上对 ρ 的取值进行采样，其中 $N_x \in$ 为某个 N。对于每个网格节点 x_i、 $\overline{i = 0}$、 …、 $N_x^d - 1$，我们使用式 (12) 近似计算 $\rho(n\tau, x_i)$。积分采用中点法则在笛卡尔网格上进行近似，该网格在 v 的每个 d 分量上均从 $-v_{\max}$ 延伸至 $v_{\max}$ ，网格宽度为 $h_v = \frac{2v_{\max}}{N_v}$:

$$\rho(n\tau, x_i) \approx \rho_{\tau,h}(n\tau, x_i) := \bar{\rho} - h_x^d h_v^d \sum_{j=0}^{N_v^d - 1} f_0\big(\tilde{\Psi}^0_{n\tau,h}(x_i, v_j)\big). \tag{13}$$

这里， v_j是笛卡尔网格在 v 方向上各单元格的中点。这要求在 $N_x^d N_v^d$ 个位置（x_i， v_j）处计算 f_0 。 $\tilde{\Psi}^0_{n\tau,\ h}$，这是 NuFI 计算开销最大的步骤。不过请注意，计算 f_0 。 $\tilde{\Psi}^0_{n\tau,\ h}$ 是一项极易并行化的操作。因此，可以在图形处理器上使用所谓的“原子加法”高效实现求和。还可以通过对 $\rho_{\tau,\ h}$的部分结果进行简单归约，将其并行化到整个图形处理器集群上。

原则上可以使用任何求积公式，而中点法则的优点是，对紧支撑光滑函数能够实现指数收敛。[2] 正如引言中所述，在许多情况下， f_0 是一个光滑函数，并且会随着 $|v| \to \infty$ 呈指数衰减，因此可以有效地视为紧支撑函数。因此，在 fi丝化发生之前、 f 仍然光滑时，我们可以期待得到非常精确的结果。同时，中点法则简单且稳定：求和中只出现正项。

因此，我们定义了网格点处的$\rho_{\tau,h}$ ，并将在这些点之间进行插值。本文假设 f，因此 ρ 和 φ 在 x中也都是周期的。因此，使用三角插值是合理的:

$$\rho_{\tau,h}(n\tau, x) = \sum_{\alpha} c_\alpha e^{i\alpha \frac{2\pi x}{L}}, \tag{14}$$

其中， α 是一个多重指标，系数 c_α 可以通过快速傅里叶变换（FFT）根据节点 x_i 处的点值$\rho_{\tau,h}(n\tau, x_i)$ 高效计算。在这种表示下，求解泊松方程$-\Delta\varphi_{\tau,h} = \rho_{\tau,h}$ 相当于对系数 c_α进行简单的缩放。利用帕塞瓦尔恒等式，也可以轻松计算电能$\frac{1}{2}\|\nabla_x \varphi_{\tau,h}\|^2_{L^2(\Omega)}$ 。

尽管从数学角度来看，$\varphi_{\tau,h}$ 的谱表示几乎是最优的，但在实际应用中却存在一个困难。计算数值 fi流 $\Psi^0_{n\tau,h}$ （或其相对$\tilde{\Psi}^0_{n\tau,h}$）时，必须经常在任意位置 x计算电 fi场，因此也需要计算$-\nabla_x \varphi_{\tau,h}$ 。这项计算代价极高，因为必须对所有模态求和 α 。因此，我们采用另一种方法。

相反，我们计算逆傅里叶变换，从而高效地获得网格节点 x_i, $i = 0, \ldots, N_x^d - 1$ 处的点值$\varphi_{\tau,h}(n\tau, x_i)$。随后，我们计算四阶笛卡尔周期张量积样条插值函数（分段、逐坐标的三次多项式，具有全局逐坐标光滑性 C^2）。这种表示同样节省内存，但单次求值

对 $-\nabla_x\varphi$ 只需恒定的时间，与分辨率 h_x 无关，因此可显著加速。

定义 2.6 （全离散近似）。全离散近似 $f_{\tau,\,h}$ 定义如下：

$$f_{\tau,h}(n\tau,x,v) \coloneqq f_0\big(\Psi^0_{n\tau,h}(x,v)\big),$$

其中 $\Psi^0_{n\tau,\,h}$ 是通过对 $\varphi_{\tau,\,h}$ 的样条插值函数求梯度，并利用所得数值近似 $E_{\tau,\,h}\approx E$ 计算得到的数值流 fl。

2.5 守恒性质

NuFI 真正与众不同之处，在于其出色的守恒性质。为避免与 x-周期情形相关的技术细节，本小节改为考虑 $\bar\rho = 0$ 下的全空间情形，但请注意，这些结果同样适用于周期情形。

定理 2.7（守恒量）。设 $F(n\tau,x,v)$ 表示精确解 $F(n\tau,x,v) = f_0\big(\Phi^0_{n\tau}(x,v)\big)$，或离散近似 $F(n\tau,x,v) = f_{\tau,h}(n\tau,x,v) = f_0\big(\Psi^0_{n\tau,h}(x,v)\big)$，$\tau>0$，$h\ge 0$。那么，无论计算 $\Psi^0_{n\tau,h}$，F 时使用的近似电场 $\mathit{field}\,E_{\tau,h}\approx E$ 存在何种误差，对于所有 $n\tau$，$n\in\mathbb{N}_0$，以下等式均成立：

- 最大值原理：$0\le F(n\tau,x,v)\le \|f_0\|_{L^\infty(\mathbb{R}^d\times\mathbb{R}^d)}$，
- 所有 $L^p(\mathbb{R}^d\times\mathbb{R}^d)$ 范数的守恒，$1\le p\le\infty$：

$$\|F(n\tau,\cdot,\cdot)\|_{L^p(\mathbb{R}^d\times\mathbb{R}^d)} = \|f_0\|_{L^p(\mathbb{R}^d\times\mathbb{R}^d)}, \tag{15}$$

- 动理学熵守恒：

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} F(n\tau,x,v)\ln\big(F(n\tau,x,v)\big)\,\mathrm{d}v\mathrm{d}x = \iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} f_0\ln f_0\,\mathrm{d}v\mathrm{d}x, \tag{16}$$

- 更一般地，对于满足以下积分有定义的任意 函数 $g:\mathbb{R}\to\mathbb{R}$：$\mathit{fi}$定：

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} g\big(F(n\tau,x,v)\big)\,\mathrm{d}v\mathrm{d}x = \iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} g\big(f_0(x,v)\big)\,\mathrm{d}v\mathrm{d}x. \tag{17}$$

证明。为简洁起见，我们记 $\Xi\in\{\Phi^0_{n\tau},\Psi^0_{n\tau,h}\}$。最大值原理可直接由 $F = f_0\circ\Xi$ and $f_0\ge 0$这一事实推出。L^∞-范数的守恒是因为 Ξ 是一个微分同胚，因此 $\Xi(\mathbb{R}^d\times\mathbb{R}^d) = \mathbb{R}^d\times\mathbb{R}^d$。熵（$g(x) = x\ln x$）和其他 L^p-范数（$g(x) = x^p$）的守恒则由最后一条结论推出。对于最后一条结论，我们对积分作变量变换，并利用 Ξ的体积保持性，得到：

$$\begin{aligned}\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} g\big(F(n\tau,x,v)\big)\,\mathrm{d}v\mathrm{d}x &= \iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} g\left(f_0\big(\Xi(x,v)\big)\right)\,\mathrm{d}v\mathrm{d}x = \\ &\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} g\big(f_0(x,v)\big)\underbrace{\left|\det\nabla_{(x,v)}\Xi^{-1}(x,v)\right|}_{\equiv 1}\,\mathrm{d}v\mathrm{d}x = \iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} g\big(f_0(x,v)\big)\,\mathrm{d}v\mathrm{d}x.\end{aligned} \tag{18}$$

□

该半离散近似 $f_{\tau,\,0}$ 还守恒动量。

定理 2.8（动量守恒）。设 $F(n\tau,x,v)$ 表示精确解 $F = f_0\big(\Phi^0_{n\tau}(x,v)\big)$，或表示定义 2.5 中的半离散近似 $F(n\tau,x,v) = f_{\tau,0}(n\tau,x,v)$。则 $F\mathit{fulfils}$ 满足动量守恒：

$$\forall n\in\mathbb{N}_0:\qquad \iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} vF(n\tau,x,v)\,\mathrm{d}v\mathrm{d}x = \iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} vf_0(x,v)\,\mathrm{d}v\mathrm{d}x. \tag{19}$$

证明。精确解的结论属于经典问题；因此我们仅证明 $f_{\tau,\,0}$ 的情形。对于 $n=0$，结论显然成立，因为 $f_{\tau,\,0}(0,\,x,\,v)\equiv f_0(x,\,v)$。现在假设该结论对某个 $n\in\mathbb{N}_0$，成立，我们将证明它对 $k:=n+1$ 也成立。简记 $\Xi:=\tilde{\Psi}^0_{k\tau,\,0}$，可得：

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} v f_{\tau,0}(k\tau,x,v)\,\mathrm{d}v\mathrm{d}x=\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} v f_0\big(\Xi(x,v+\tfrac{\tau}{2}E_{\tau,0}(k\tau,x))\big)\,\mathrm{d}v\mathrm{d}x. \tag{20}$$

现在，我们执行与 (12) 中相同的变量变换，不过需要注意，由于乘积 $vf_{\tau,\,0}$情况会稍微复杂一些。令 $\hat{v}=v+\frac{\tau}{2}E_{\tau,\,0}\big(k\tau,\;x\big)$，并注意到本节中我们假设处于满足 $\bar{\rho}=0$ 的全空间情形，可得：

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d}\hat{v}f_0\big(\Xi(x,\hat{v})\big)\,\mathrm{d}\hat{v}\mathrm{d}x+\frac{\tau}{2}\int\limits_{\mathbb{R}^d}E_{\tau,0}(k\tau,x)\rho_{\tau,0}(k\tau,x)\,\mathrm{d}x. \tag{21}$$

注意，$E_{\tau,\,0}=-\nabla\varphi_{\tau,\,0}$ 且 $\rho_{\tau,\,0}=-\Delta\varphi_{\tau,\,0}$。此外，对于任何足够光滑且在无穷远处衰减足够快的函数 φ，利用分部积分、交换微分顺序以及拉普拉斯算子是自伴的这一事实，可得：

$$\int_{\mathbb{R}^d}\nabla\varphi\Delta\varphi\,\mathrm{d}x=-\int_{\mathbb{R}^d}\varphi\nabla\Delta\varphi\,\mathrm{d}x=-\int_{\mathbb{R}^d}\varphi\Delta\nabla\varphi\,\mathrm{d}x=-\int_{\mathbb{R}^d}\Delta\varphi\nabla\varphi\,\mathrm{d}x. \tag{22}$$

因此，该积分等于其相反数，所以 (21) 中的第二个积分为零。至于第 fi一个积分，我们再进行一次变量变换，并令 $\hat{x}=x-\tau\hat{v}$：

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d}\hat{v}f_0\big(\Xi(x,\hat{v})\big)\,\mathrm{d}\hat{v}\mathrm{d}x=\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d}\hat{v}f_0\big(\Psi^0_{n\tau,0}(\hat{x},\hat{v}+\tfrac{\tau}{2}E_{\tau,0}(n\tau,\hat{x}))\big)\,\mathrm{d}\hat{v}\mathrm{d}\hat{x}. \tag{23}$$

因此，通过进 行 final 变量变换，并对 $\hat{v}$ 重复上述相同的论证，我们得到 in 如下：

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} v f_{\tau,0}\big((n+1)\tau,x,v\big)\,\mathrm{d}v\mathrm{d}x=\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} v f_{\tau,0}\big(n\tau,x,v\big)\,\mathrm{d}v\mathrm{d}x. \tag{24}$$

因此，由归纳法可得结论。 □

最后，精确解也满足能量守恒。遗憾的是，目前尚不清楚 NuFI 在多大程度上也具有这一性质，我们将在本文末尾介绍相关数值实验。

定理 2.9（能量守恒）。精确解 $f(t,x,v)=f_0\big(\Phi^0_t(x,v)\big)$ 和电 fi场E 满足能量守恒：

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d}v^2f(t,x,v)\,\mathrm{d}v\mathrm{d}x+\frac{1}{2}\int\limits_{\mathbb{R}^d}E(t,x)^2\,\mathrm{d}x\right)=0. \tag{25}$$

2.6 复杂度

2.6.1 内存复杂度

在本文中，我们将假设 f_0 以紧凑数学表达式的形式给出，因此基本上可以零成本存储。在这种情况下，NuFI 基本上只需要为每个时间步长存储 $\varphi_{\tau,h}$。

必须强调，由于$\varphi_{\tau,\,h}$不依赖于 v，将 φ的样条系数fi存储在网格上所需的内存，远少于存储 f所需的内存，差距可达数个数量级。尤其是当 $d=3$时，内存消耗可减少数百万倍。令 $n\in\mathbb{N}$表示需要计算的时间步数。那么，我们需要存储$\varphi_{\tau,\,h}(k\tau,\,\cdot)$，其中 $k=0,\,1,\,\ldots,\,n$。由此可得：

$$\text{Memory Requirement}=(n+1)N_x^d\times\text{size of one floating point value}. \tag{26}$$

d	$N_x = N_v$	每步 $\varphi_{\tau,\ h}$ 所需内存	用于存储 f 的内存
2	128	0.125 兆二进制字节	2 048 兆二进制字节
2	256	0.5 兆二进制字节	32 768 兆二进制字节
2	512	2 兆二进制字节	524 288 兆二进制字节
3	128	0.015 吉二进制字节	32 768 吉二进制字节
3	256	0.125 吉二进制字节	2 097 152 吉二进制字节
3	512	1 吉二进制字节	134 217 728 吉二进制字节

表1：在 d 维网格上存储电势 φ 所需的内存，与在 $2d$ 维网格上直接存储分布函数 f 所需的内存进行比较，针对多种示例离散化。

虽然 NuFI 中的内存需求会随时间步数增加，但单个时间步所需的数据，与将 f 存储在 $2d$ 维网格上所需的数据相比微不足道，后者需要 $N_x^d N_v^d$ fl个浮点值。假设每个 fl浮点值占 8 字节，不同的内存需求如表1所示。

这意味着，即使对于具有 $d = 3$ 和 $N_x = 256$ 的模拟，$\varphi_{\tau,\ h}$ 也能完整存储在现代图形处理器的内存中，持续数百个时间步。

2.6.2 计算复杂度

NuFI 中计算开销最大得多的操作，是使用数值求积计算 $\rho_h(n\tau,\ x_i)$。所有其他操作的开销都可以忽略不计：数值求积在 $2d$ 维的（x，v）空间中进行，而所有其他操作则在维数更低的 x 空间中进行。

为简单起见，我们假设 f_0 可在 O(1) 的成本下高效求值。在每个时间步$k = 1,\ 2,\ \ldots,\ n$，都需要在 $N_x^d N_v^d$ 个求积节点处计算数值通量 flux $\Psi^0_{k\tau,\ h}$。而计算$\Psi^0_{k\tau,\ h}$ 又需要对近似电场 field $E_{\tau,\ h}$求值 $k+1$ 次；因此总成本约为：

$$N_x^d N_v^d \sum_{k=1}^{n} (k+1) = \mathcal{O}\left(\frac{n^2}{2} N_x^d N_v^d\right) \tag{27}$$

次求值 $E_{\tau,\ h}$。因此，必须具备高效计算 $E_{\tau,\ h}$ 的程序。采用样条函数的方法中，单次计算 $E_{\tau,\ h}$ 的成本与 n、N_x 和 N_v 无关。

因此，与传统方法不同，NuFI 的成本会随时间步数呈二次增长。这可能会让这种方法显得缺乏吸引力。然而，与直接存储 f 的方法相比，NuFI 的浮点运算/字节比高得多，因此在现代计算机系统上能实现高得多的性能。此外，由于存储需求低，它也很容易在集群上并行化。最后，据作者所知，没有其他方法像 NuFI 一样能够守恒如此多的物理性质。或许正因如此，正如数值实验将firm证实的那样，NuFI 只需相对较粗的分辨率 N_x、N_v，就能获得定性上良好的结果。

3 与相关文献和方法的相似之处与差异

NuFI 的一个核心思想是对相-fl流进行近似并追踪相-fl流 Φ_s^t，同时利用特征线法，充分发挥沿相-fl流 f 的值能够精确保守这一特性。从某种意义上说，这与拉格朗日格式（如光滑粒子流体动力学（SPH）[3],[4],[5] 、粒子网格法（PIC）[5],[6],[7] 以及基于插值和近似的粒子方法[8],[9] ）的思路类似。PIC 等粒子方法与 NuFI 的主要差异在于，NuFI 直接对相空间进行离散，并且沿特征线向前追踪，而不是向后追踪。

粒子方法尝试通过对初始分布函数进行采样，直接离散化相空间：在相空间中选取一组点，称为“粒子” f_0。随后沿各自的特征线追踪这些粒子。对于非线性情形，需要根据这些粒子及其携带的相应 f值计算电 fl场。不同的拉格朗日格式主要在这一点上有所ff异ff

硬件	CPU	GPU
工作站笔记本电脑	英特尔至强 E-2276M @ 2.8 GHz，6 个核心，每个核心 2 个线程	英伟达 Quadro T1000 Mobile，4 GB DDR5 显存
Claix GPU 集群	2×英特尔至强 铂金版 8160 ‘Skylake’，2.1 GHz，24 个核心 每个节点	2×英伟达 Volta V100-SXM2，每个节点 16 GB HBM2 内存

表 2：数值实验所用硬件配置概览。

有所不同，即如何根据采样点计算电 fi场。对于常用的 PIC 格式，将粒子映射到 x中的网格上，从而计算 ρ 的近似值，再通过同一网格上的泊松求解器求得 E 。

原则上，PIC 具有若干理想的守恒性质，但它难以应对由细丝化和解的陡峭梯度造成的过量噪声。这使得每隔几个时间步就必须进行一次重新网格化。重新网格化能降低数值模拟中的噪声，但会引入一定的数值扩散，这与半拉格朗日方法类似。由于存在陡峭梯度，直接近似解的方法会在数值解中产生过冲。

与拉格朗日方法相关的，尤其是 PIC 的方法，是“半拉格朗日”方法。与直接离散分布函数并更新网格节点上数值的纯欧拉方法类似，半拉格朗日格式也采用相空间上的完整网格离散化。然而，为避免过于严格的时间步长限制并减少数值扩散，它们会沿特征线追踪“网格点的运动”，以便利用上一个时间步对分布函数的近似值，在当前时间步进行插值或近似。NuFI 同样可以采用较大的时间步。

如第 2.6.1 节所述，通过网格在整个相空间中直接进行离散化的代价很高。克服这一问题的方法，例如 Kormann 关于将解分解为张量列格式的论文，以及 Einkemmer 关于比较间断 Galerkin 方法与基于样条的半拉格朗日方法的论文，均对此进行了探讨。[19],[20]

4 数值实验

本节介绍若干数值实验的结果。为此，我们使用了两种不同的ff硬件配fi置：本地工作站笔记本电脑，以及亚琛工业大学的 Claix GPU 集群，参见表 2。

4.1 弱朗道阻尼 ($d = 1$)

第 fi 个测试案例通常称为弱朗道阻尼。初始条件为

$$f_0(x,v) := \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} \big(1 + \alpha \cos(kx)\big), \quad (x,v) \in [0,L] \times \mathbb{R} \tag{28}$$

其中 $k = 0.5$、 $\alpha = 0.01$、 $L = 4\pi$。速度空间截断于 $v_{\max} = 10$。我们采用不同的空间分辨率N_x 和 N_v，但所有模拟使用相同的时间步长 $\tau = \frac{1}{16}$。

初始状态（28）是对麦克斯韦分布的微小扰动

$$f_M(v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} \tag{29}$$

它是 Vlasov–Poisson 方程的一个平衡态（1）。已知这种特定扰动会以弱意义周期性地衰减至零，当 $t \to \infty$。因此，该基准测试旨在复现电 fi场的正确阻尼率。所谓的 Landau 阻尼率为 $\gamma_E = 0.153359$, 是相应电能阻尼率的一半 $\gamma_{\text{energy}} = 0.306718$。[19]

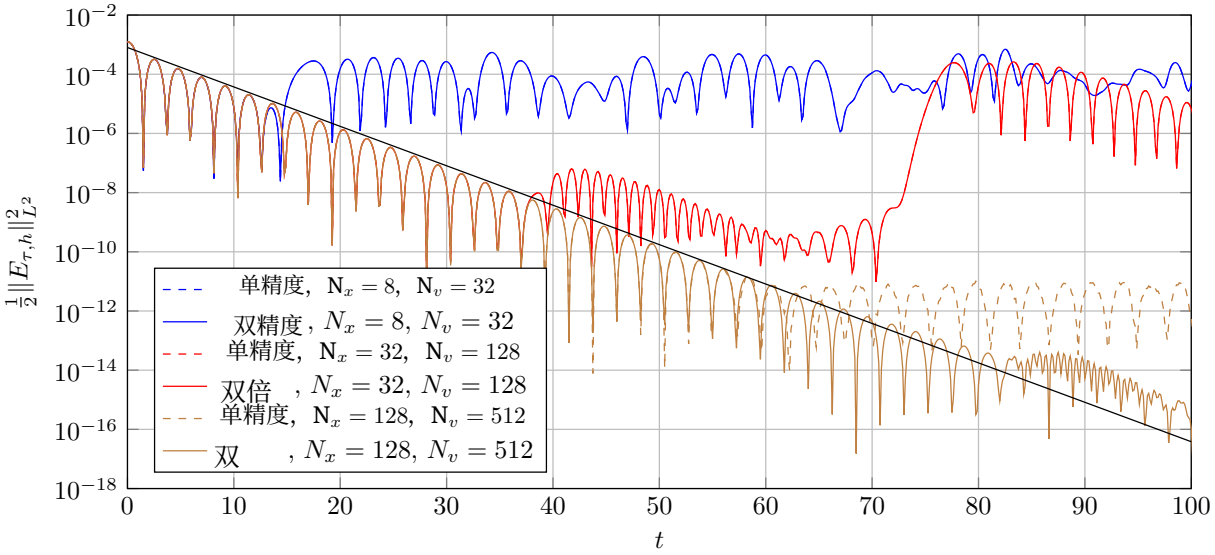


图3：弱朗道阻尼基准测试（$d = 1$）中电能的变化，展示了采用单精度和双精度算术进行的数值模拟结果。除 fi 嵌套分辨率外，单精度和双精度的结果无法区分。

对于 $d = 1$，仍可在工作站笔记本电脑上进行有意义的数值模拟。对于此类硬件，单精度计算通常快得多，而且在 $d = 1$ 时精度足够。在最高分辨率下，使用单精度完成全部 1 600 个时间步共耗时 1.7 秒。由于这台笔记本电脑仅配备消费级图形处理器芯片，双精度计算明显较慢，耗时 28.3 秒。

结果如图3所示。在低分辨率和中等分辨率下，单精度和双精度计算的结果无法区分。只有在高分辨率下，才会观察到在ff之后出现不同的行为： $t \approx 60$一方面，采用单精度计算时，在ff之后观察不到阻尼效 $t \approx 62$ 应，因为电能已达到相对于单精度而言的机器精度，这表明 Landau 阻尼最多只能解析到所选的 fi浮点精度。另一方面，双精度下阻尼效ff应仍可观察到 $t \approx 80$，但衰减速率略慢。单精度下未观察到回归现象：相反，电能在 $t \approx 62$之后以相同振幅振荡。双精度数值模拟则在 $t \approx 83$处显示出轻微的回归。

所有模拟都再现了正确的阻尼率和振荡频率。最高分辨率的模拟能够捕捉到正确的阻尼率，直到 γ_{energy} $t \approx 62$。较低分辨率的模拟只能在各自发生复现的时刻之前捕捉到正确的阻尼率。

4.2 双流不稳定性 ($d = 1$)

接下来，我们研究双流不稳定性基准测试。初始条件为

$$f_0(x, v) := \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} v^2 \big(1 + \alpha \cos(kx)\big), \quad (x, v) \in [0, L] \times \mathbb{R} \tag{30}$$

其中 $k = 0.5$、 $\alpha = 0.01$、 $L = 4\pi$。速度空间截断于 $v_{\max} = 10$，时间积分采用的时间步长为 $\tau = \frac{1}{16}$。我们再次考虑低分辨率 ($N_x = 8$、 $N_v = 32$)、中分辨率 ($N_x = 32$、 $N_v = 128$) 和高分辨率 ($N_x = 128$、 $N_v = 512$) 的模拟。

此基准测试模拟两束速度相反、且均略微偏离平衡态的电子流发生碰撞的过程。经过初始阻尼阶段后，两束粒子流开始混合，行为逐渐变得更加湍流。特别是，分布函数 f 中的 fi丝化现象随时间愈发明显，并形成一个“涡旋”。

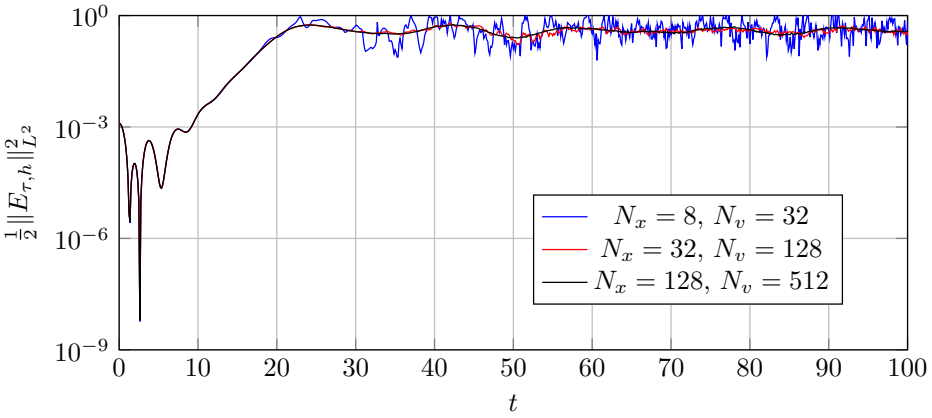


图4：双流不稳定性的电能基准测试（$d = 1$），展示了使用双精度并采用时间步长 $\tau = \frac{1}{16}$运行的模拟。

图4 展示了电能随时间的演变，直至 $T = 100$。在 $t \approx 10$ 之前，阻尼对ff两束粒子的影响仍主导着整体动力学。在$t \approx 10$ 和 $t \approx 25$ 之间，相空间中开始形成一个涡旋。到了$t \approx 25$ 之后，混合过程占据主导地位，此时解中的湍流和fi丝化现象使得对 f 进行解析变得愈发复杂。直到 $t \approx 25$ 之前，三种模拟都能正确捕捉电能中潜在的动力学特征。到了 $t \approx 25$ 之后，低分辨率模拟仍能定性地捕捉这些动力学特征，但误差显fi著。到了 $t \approx 40$之后，中等分辨率模拟也开始偏离高分辨率模拟，但偏离程度小得多。中等和高分辨率模拟能够解析 $t \approx 25$之后电能的缓慢振荡。高分辨率模拟能够重现电能，直至 $t = 100$，且图中没有显fi著的数值噪声。

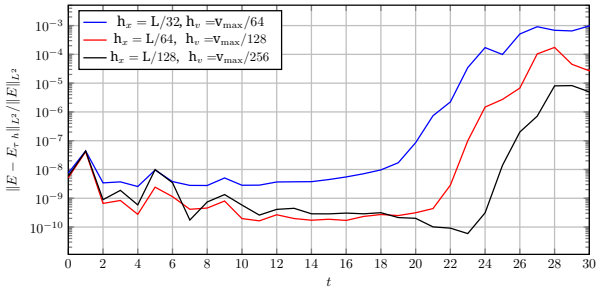
图6 展示了分布函数 $f_{\tau,\ h}$ 在若干时刻 t的情况。在 $t \approx 5$，电 fi场阻尼不再占主导地位。此时，与初始数据 f 的差异几乎难以察觉。在 $t \approx 25$附近，相空间中涡旋的形成开始显现，同时电能达到全局最大值。此后 $t \approx 25$，涡旋呈周期性旋转，为之后的 t产生越来越多的 fi丝状结构。低分辨率也能够在定性意义上正确捕捉总体动力学，但可以观察到低分辨率模拟中的畸变不断增加。对于 $t = 100$，在低分辨率模拟中，涡旋似乎沿 x-轴移动，这是一种错误的动力学行为。

更高分辨率的数值模拟能够正确再现动力学过程，直至结束（$t = 100$）。此外，NuFI 能够再现解析解所预期的 fi细 fi丝结构。图6中可见的像素伪影是由于绘图分辨率有限（每个方向上有 2 048 个采样点）所致；若绘图分辨率足够高，这些伪影便会消失。

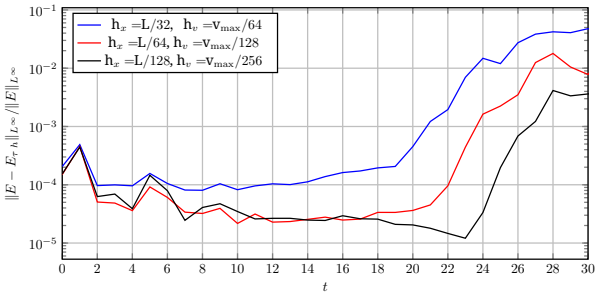
图5展示了在若干时间和相空间分辨率设置以及不同ff浮点数表示方式 fl下，数值模拟产生的相对误差。参考解使用 $N_x = 512$，$N_v = 2\,048$ 和 $\tau = \frac{1}{32}$计算得到。

改变相空间分辨率时可以观察到，直到 $t \approx 20$，所有误差都保持在大致相同的水平。随后，伴随着解中 fi丝结构不断增多，混合过程开始出现，导致所有分辨率下的误差都增大。经过 $t \approx 26$ 后，误差似乎再次稳定在一个fi定水平；单精度数值模拟的相对误差介于 10^{-3} 至 10^{-5} 之间，而相对于 L^∞ 范数的误差介于 10^{-1} 至 10^{-2} 之间。双精度下也能观察到类似的误差行为，这表明在此情况下，数值模拟进入湍流阶段后，所选的 fi浮点精度不再产生影响。但请注意，在早期时刻，双精度下的误差最多可小八个数量级。

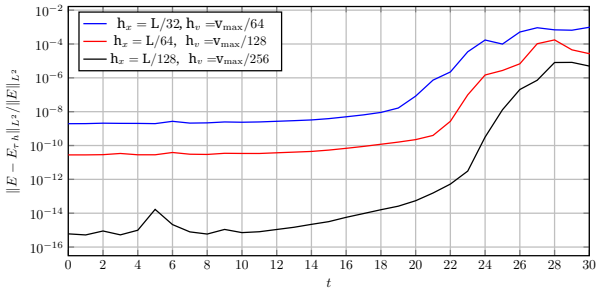
改变时间步长并固定相空间分辨率时，我们观察到误差在 $t = 0$ 和 $t \approx 2$之间出现了初始跃升。时间步长较小时，误差增长幅度大大减小。这表明，在数值模拟的这一阶段，数值求积和泊松求解器产生的误差已接近机器精度 10^{-8}，而占主导地位的是时间离散误差。经过 $t \approx 2$ 后，误差水平趋于稳定，直到约 $t \approx 26$时开始出现 fi丝化现象，误差才会进一步显著增大。



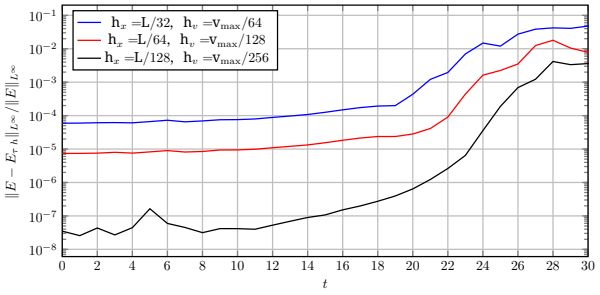
(a) L^2-误差，单精度



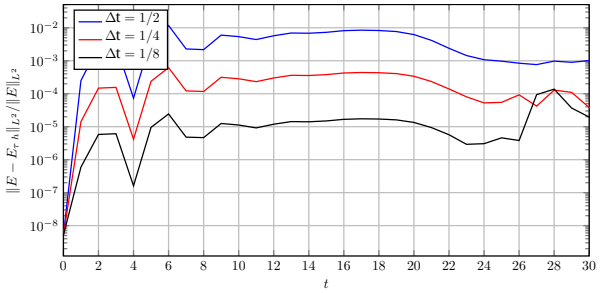
(b) L^∞ 误差，单精度



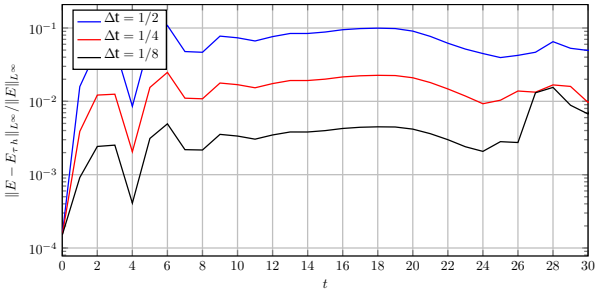
(c) L^2-误差，双精度



(d) L^∞-误差，双精度

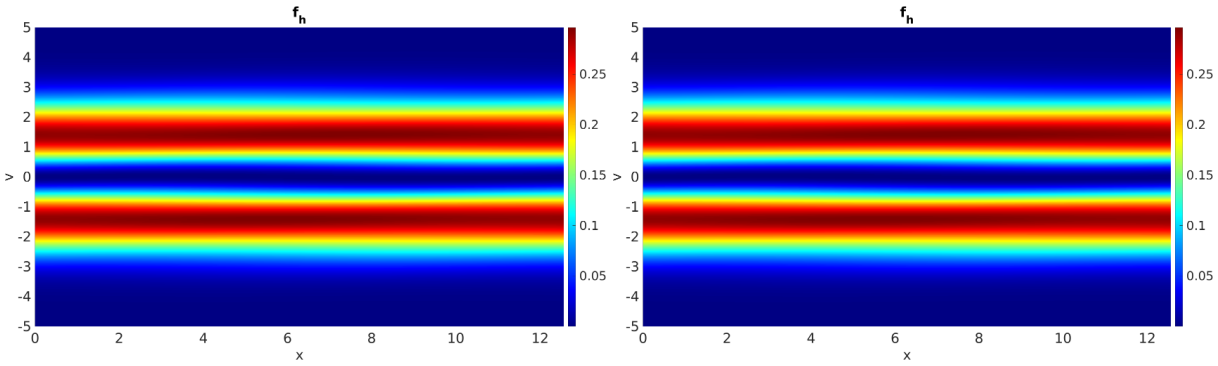


(e) L^2-误差，单精度

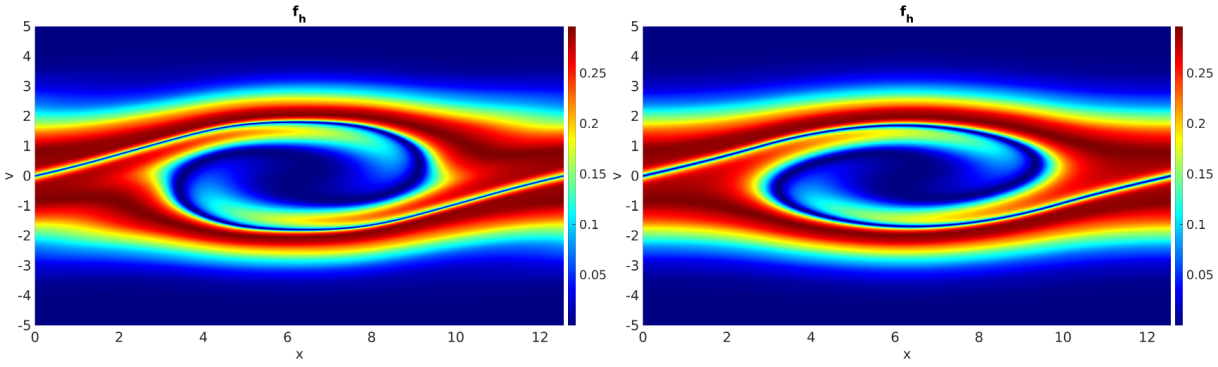


(f) L^∞-误差，单精度

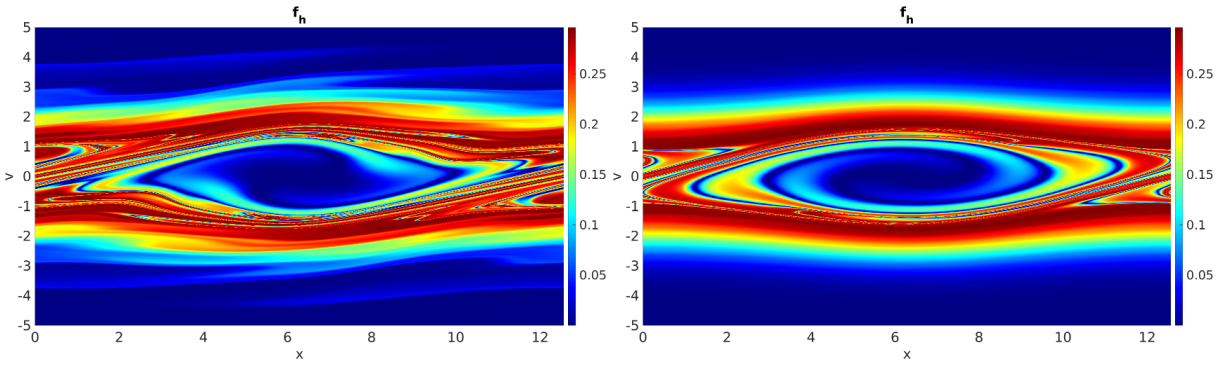
图5：电场中的相对误差 field，相对于 L^2- 和 $L^\infty-$范数，针对 双流不稳定性 基准测试，采用 $d = 1$。使用 $\tau = \frac{1}{16}$ 或 $N_x = 64$，$N_v = 256$ fixed，分别。参考解使用 $N_x = 512$，$N_v = 2\,048$ 和 $\tau = \frac{1}{32}$。



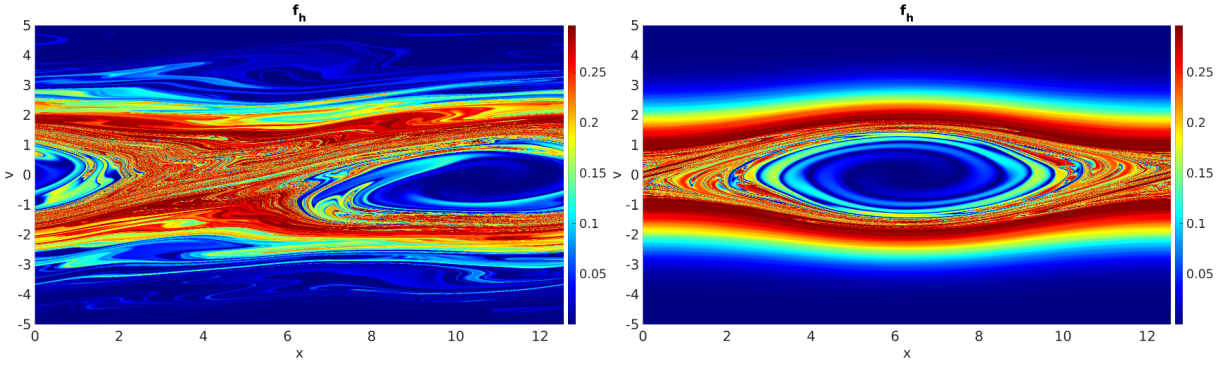
$t = 5$



$t = 25$



$t = 50$



$t = 100$

图6：分布函数$f_{\tau,\ h}$（t，x，v）是双流不稳定性基准测试的结果。左图 $N_x = 8$，$N_v = 32$ 和右图 $N_x = 128$，$N_v = 512$。两者均使用 $\tau = \frac{1}{16}$ 作为时间步长。绘图分辨率在 x方向和 v方向上均为2 048 个点。

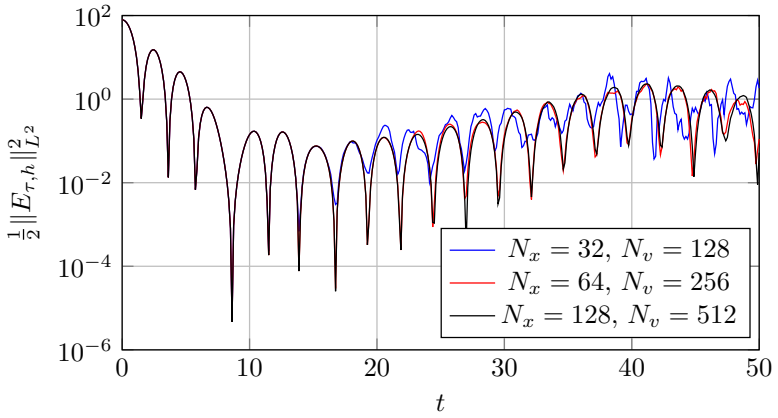


图 7：电　场能量：$d = 2$ 中的强朗道阻尼基准测试，时间步长设为 $\tau =$ $\frac{1}{16}$.

4.3 强朗道阻尼（$d = 2$）

对于二维情形，我们考虑强朗道阻尼或非线性 *Landau* 阻尼基准测试。初始条件为

$$f_0(x, y, u, v) := \frac{1}{2\pi} e^{-\frac{v^2+u^2}{2}} \big(1 + \alpha(\cos(kx) + \cos(ky))\big), \quad (x, y, u, v) \in [0, L]^2 \times \mathbb{R}^2 \tag{31}$$

其中 $k = 0.5$、$\alpha = 0.5$ 和 $L = 4\pi$。速度空间截断于 $v_{\max} = 10$，我们选择 $\tau = \frac{1}{16}$ 作为时间步长。所有计算均采用双精度算术。

初始数据是麦克斯韦分布函数的强扰动。因此，我们预期电场强度会大幅增长。从这个意义上说，与弱 *Landau* 阻尼 基准测试相比，本案例更接近双流不稳定性 。

图 7 展示了电能随时间的演化。经过一段初始的周期性阻尼后，直到$t \approx 10$，电能在 10^2 和 10^{-1} 之间变化，电场 fi再次受到激发。电能周期性地增长，直到 $t \approx 42$，之后又再次受到阻尼。

高分辨率模拟能够正确解析电能的动力学，直至$t = 50$；但激发开始后，低分辨率模拟便开始难以应对，在 $t \approx 20$ 时，与高分辨率解的偏差已显而易见。尤其是在 $t \approx 30$ 之后，低分辨率模拟也无法再捕捉电能正确的峰值和谷值。不过，它仍能定性地正确再现振荡特性及整体动力学。

在图8中，我们绘制了相空间中不同分辨率下的相对误差ff。参考解采用 $N_x = 256$、$N_v = 1\,024$ 和 $\tau = \frac{1}{32}$计算。

—

起初，高分辨率数值模拟中的相对 L^2-误差再次接近机器精度，与 $d = 1$ 情形相似。对于低分辨率，模拟误差在L^2-范数下约为 10^{-8}，在 L^∞-范数下为 10^{-4}。然而，与 $d = 1$ 双流基准测试相比，误差增长明显更快。误差持续增大，直到 $t \approx 20$，随后趋于平稳：低分辨率下，在 L^2-范数和 L^∞-范数中介于 10^{-1} 之间；高分辨率模拟下，在 L^2-范数中为 10^{-4}，在 L^∞-范数中为 10^{-2}。

图9和10 展示了高、低分辨率模拟在不同时间的电场 fi分布 t。在这些图中，两种模拟在定性或定量上均无可观察到的差异ff，这表明即使是低分辨率模拟，也能够正确捕捉电场中的动力学，至少持续到 fi $t = 50$。

4.4 双流不稳定性（$d = 3$）

对于三　　维情形下，我们再次考虑双流不稳定性基准测试。初始条件　　on is

$$f_0(x, y, z, u, v, w) := \frac{1}{2(2\pi)^{\frac{3}{2}}} \big(e^{-\frac{(v-v_0)^2}{2}} + e^{-\frac{(v+v_0)^2}{2}}\big) e^{-\frac{u^2+w^2}{2}} \big(1 + \alpha(\cos(kx) + \cos(ky) + \cos(kz)\big), \tag{32}$$

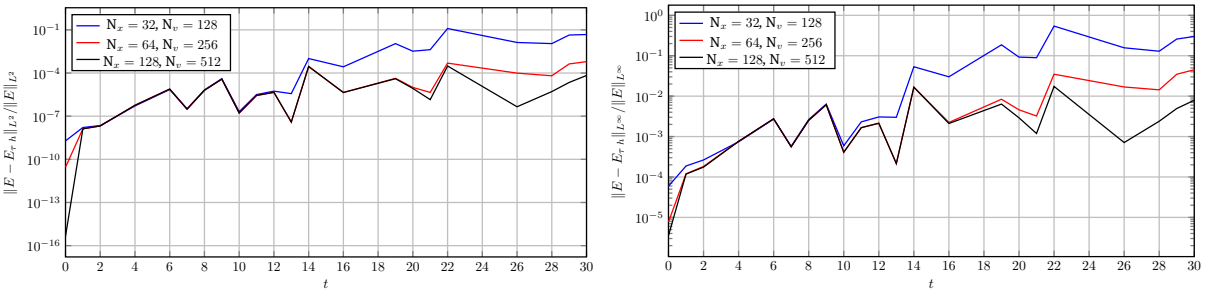


图8: L^2- 和 L^∞ 相对于以下问题参考解的误差：强朗道阻尼 基准测试，采用 $d = 2$计算。时间步长为 $\tau = \frac{1}{16}$。参考解使用 $N_x = 256$，$N_v = 1\,024$计算。

为 $(x, y, z, u, v, w) \in [0, L]^3 \times \mathbb{R}^3$ 其中 $k = 0.2$、$\alpha = 10^{-3}$、$L = 4\pi$ 和 $v_0 = 2.4$。这些参数与 Kormann 选用的参数相同。[19] 速度空间截断于 $v_{\max} = 10$，我们采用 $\tau = \frac{1}{16}$ 作为时间步长。我们在不同分辨率下进行数值模拟；最 fi的情形使用 $N_x = 32$、$N_v = 64$，因此相空间中的求积节点超过八十亿个。使用 Claix 集群的八个节点，每个节点配备两块图形处理器，计算全部 $n = 800$ 个时间步耗时 3 小时 58 分钟。同样，这里仅考虑双精度算术。

图11 展示了电能随时间的演化。与 $d = 1$ 基准测试不同，平衡态扰动的振幅较小，因此电场的激发以及涡旋形成均有所延迟。电能峰值出现在 $t \approx 35$。较低分辨率的数值模拟无法捕捉部分峰值出现时刻；在 $t \approx 30$ 之后，其振荡速度比高分辨率数值模拟更快。不过，即使空间方向只有 8 个点、速度方向只有 16 个点，NuFI 仍能以定性正确的方式捕捉整体动力学，直至 $t \approx 30$。

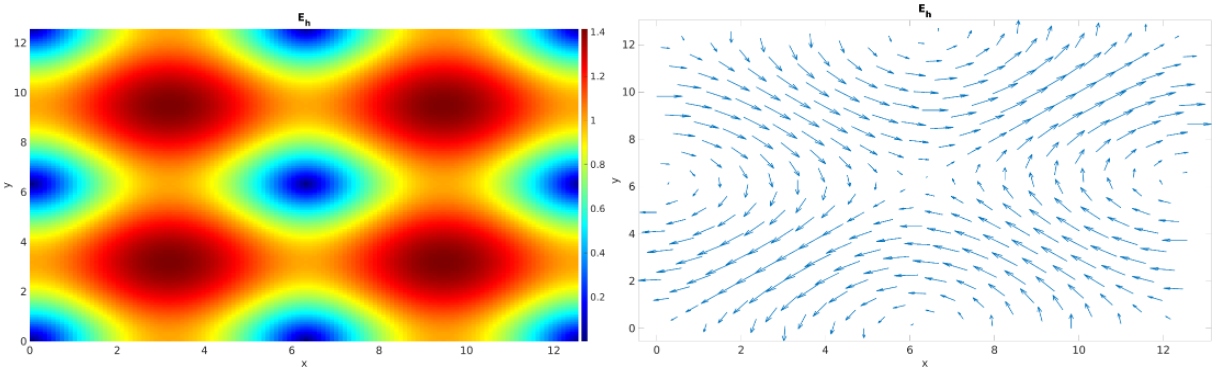
图12 展示了电场在 fi处的一个典型截面 $z = 0$。电场 fi沿 x轴排列，在约 $y > 15$ 处方向为正，在 $y < 15$处方向为负。

4.5 关于能量守恒

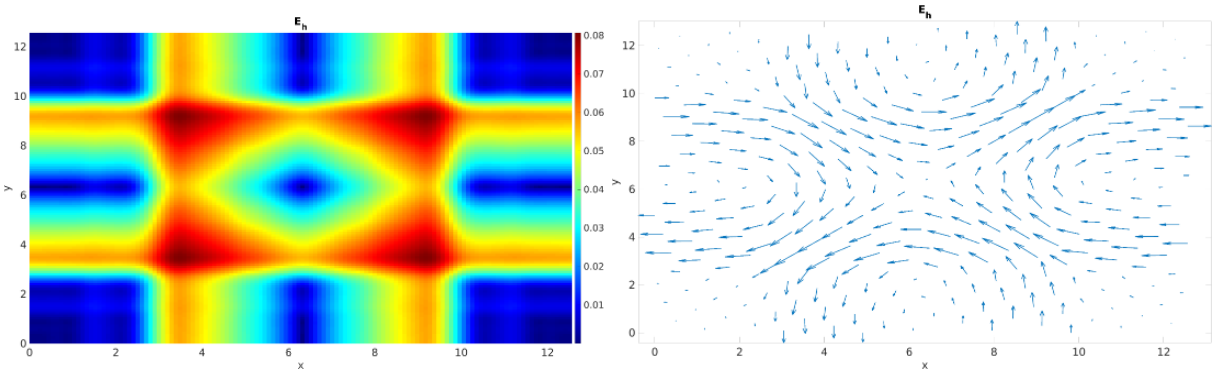
根据 第 2.5 节 可知，所有 L^p范数以及 $f_{\tau,\,h}$ 和动理学熵都由 NuFI 精确守恒。虽然守恒是精确的，但积分（15）和（16）的实际数值只能通过求积公式近似计算。

总能量也一样：其数值也只能通过数值求积近似计算。此外，由于存在各种离散误差，我们无法期望能量精确守恒。即使在半离散格式中，$h \to 0$ 采用固定的时间步长 fi时，τ也会出现这种情况。特别是，在许多力学系统中，辛时间积分器通常不能精确守恒能量；不过，总能量误差仍然有界，不会随时间增长。换言之，尽管这些格式通常不能精确守恒能量，却不会产生系统性的能量漂移。因此，NuFI 是否也有这种性质，便成了一个值得探讨的问题。

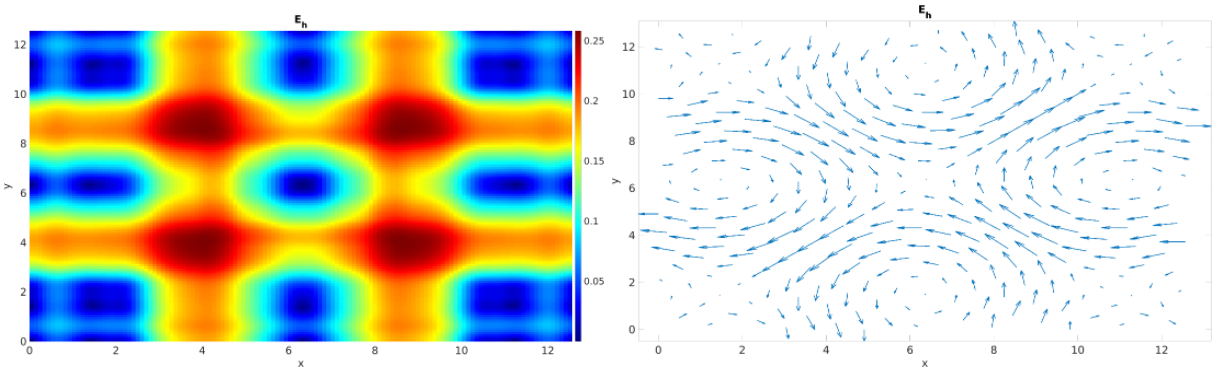
为了至少在一定程度上回答这个问题，我们考虑 $d = 1$中的$\tau = \frac{1}{32}$ 双流不稳定性基准测试。在 $t = 0$ 时，初始数据 f_0 及其对应的电荷密度 ρ 都非常光滑，因此可以通过数值求积准确计算守恒量和总能量。在 $t = 0$ 时得到的数值因而可作为参考值。接下来，我们按如下方式进行：在每个时间步长，均使用与（13）相同的求积公式计算积分。结果如图 13所示。对于熵和 L^p范数，相较于 $t = 0$ 时的值，偏差仅来自求积误差。总能量的数值则同时包含求积误差和离散误差。不过，我们观察到，总能量近似值的相对误差与其他守恒量的求积误差处于同一量级。因此，我们得出结论：总能量误差小于求积误差；如果存在系统性能量漂移，只有使用更多求积点才能检测到。由于所有测试的分辨率下情况皆如此，我们认为能量漂移不太可能存在。



$t = 0$

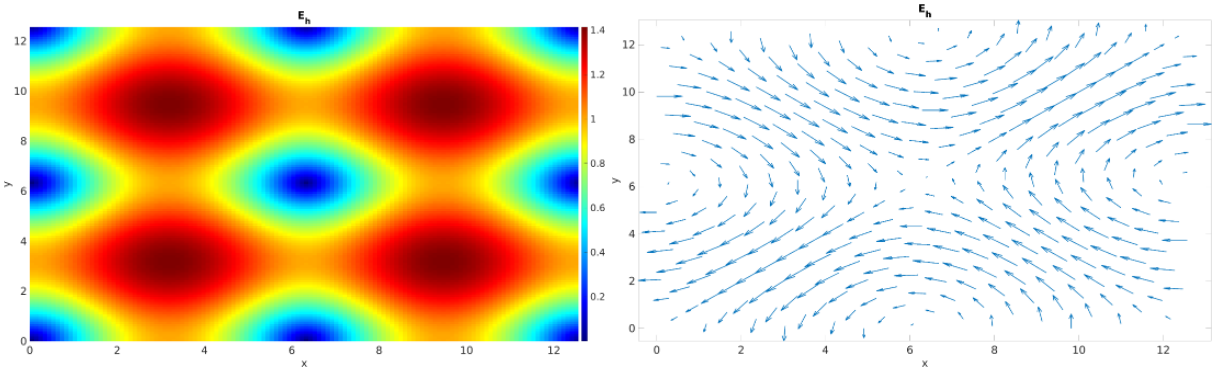


$t = 25$

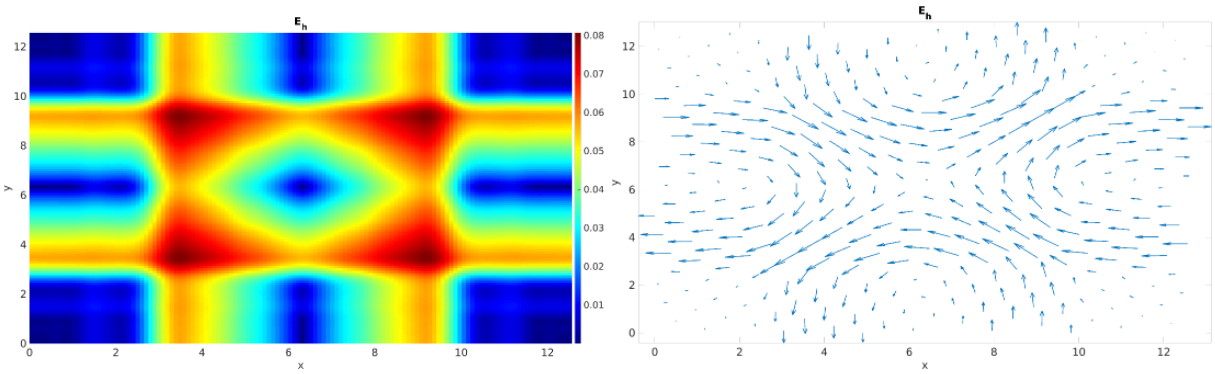


$t = 50$

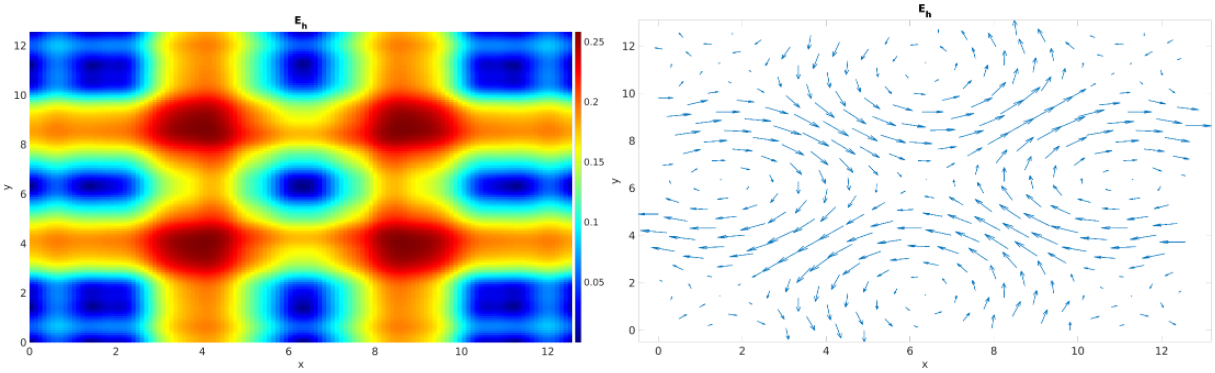
图9：电场 fi场$E_{\tau,\ h}$ 对于强朗道阻尼基准测试，位于 $d = 2$ 在不同ff时刻 $t_{\circ}$ 左侧显示电场的幅值 $|E(t,\ \ x)|$，右侧展示电场 fi场的方向。数值模拟采用 $\tau = \frac{1}{16}$，$N_x = 32$，$N_v = 128_{\circ}$



$t = 0$

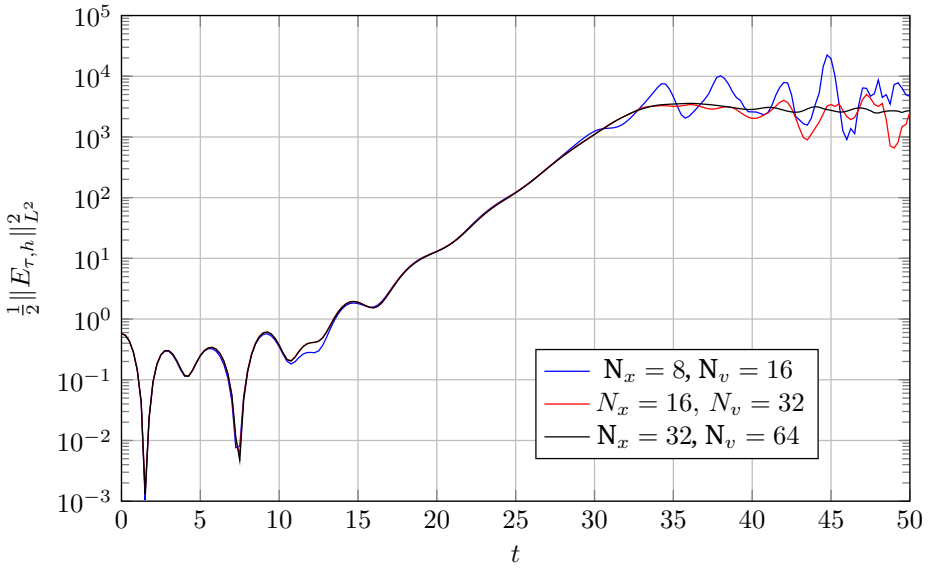


$t = 25$



$t = 50$

图10：与图9相同，但这次采用高分辨率 $N_x = 128$, $N_v = 512$。两次模拟定性地产生了相同的电 fi场。



Figu图 11： $d = 3$ 中双流不稳定性 基准测试的电能。时间步长 $\tau = \frac{1}{16}$ —.

4.6 计算时间与扩展性

如前所述，NuFI 可以在 GPU 集群上得到非常高效的实现。因此，本节将考察 NuFI 的计算时间和扩展行为。这些计算所用的代码可向作者索取。我们仅考虑 $d = 1$ 和 $d = 2$ 这两种情况，因为它们允许使用更多样本：对于 $d = 3$，求积节点数量增长过快，最终得到的图表中只会包含少数几个计算结果。

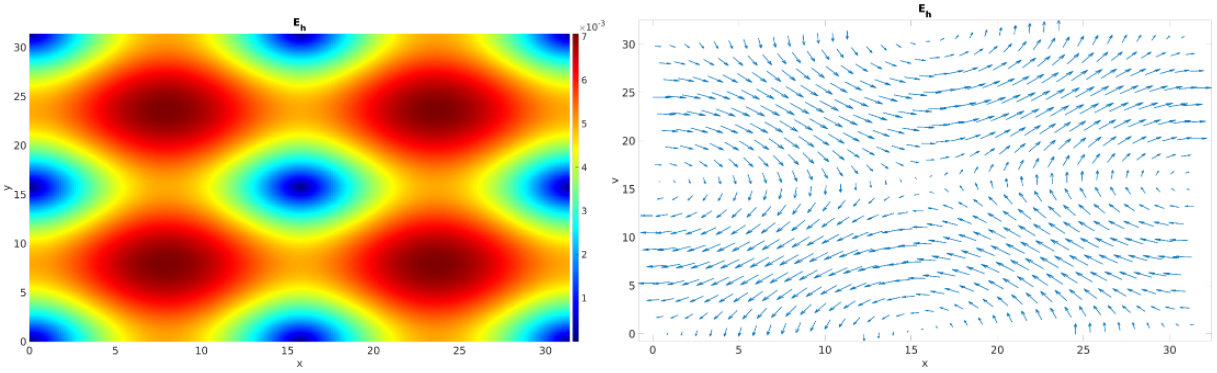
首先，我们考察 Claix 集群单个节点上的计算时间，见 表 2。对于 $d = 1$，仅使用一块图形处理器和单精度；对于$d = 2$，则使用两块图形处理器和双精度。从 $N_x = 8$ 开始， $N_v = 32$，我们测量了完成 $n = 480$ 个时间步的总计算时间， $\tau = \frac{1}{16}$，即在 $t_{\text{end}} = 30$ 时停止数值模拟。对于之后的每次模拟， N_x 和 N_v 的值都加倍。

计算时间显示在图14中。对于 $d = 1$，计算时间在求积节点数约达到一百万之前始终低于 10^{-1} 秒。在此之前，图形处理器中仍有空闲的执行单元，因此计算时间大致保持不变。之后，直到约十亿个点，计算时间呈线性增长。此时，图表的曲线再次出现向上的拐点。我们认为这是由缓存效应造成的：在达到这个点之前，单个时间步 fit 的 $\varphi_{\tau,\ h}$ 系数可以放入图形处理器中大小为 128 KiB 的缓存。对于 $d = 2$，我们观察到从第二个测量点开始呈线性增长。

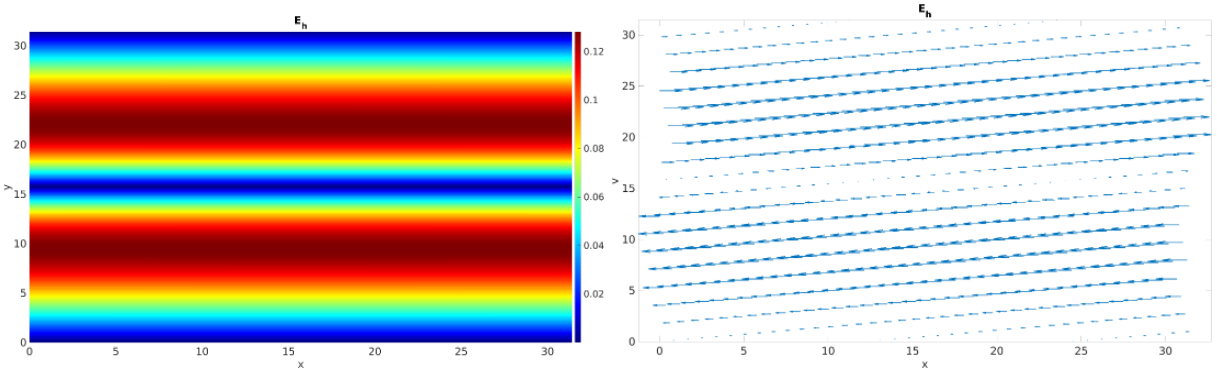
接下来，我们在 $d = 2$ 上进行一项强扩展实验，其中 $N_x = 64$， $N_v = 256$，并使用 $n = 480$ 个时间步长，以双精度进行计算。从单个节点开始，我们测量随着 Claix 集群中的计算节点数量增加而变化的计算时间。结果如 图15 所示。我们观察到几乎达到最优的强扩展性，即计算节点数量翻倍时，计算时间减半。在使用八个计算节点的测量点之后，图中出现了轻微拐点；我们认为这是由基础计算造成的，这些计算所需时间不受节点数量影响，例如泊松求解器采用的串行、非并行实现。待获得相应计算资源后，我们计划在更大型的机器上开展规模更大的模拟。

5 结论

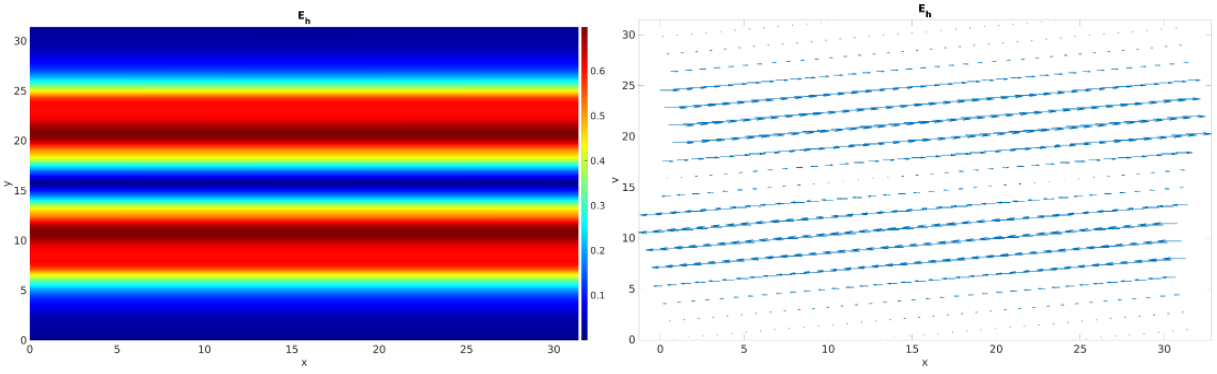
数值流迭代是一种用于数值求解 Vlasov–Poisson 方程的新方法，具有优异的守恒性质、较低的内存需求和较高的算术强度。它非常适合



$t = 0$



$t = 25$



$t = 50$

图12：电场 fi场$E_{\tau,\ h}$ 对于双流基准测试，在 $d = 3$，于不同ff时刻 t，针对 $z = 0$。左图：幅值$|E_{\tau,\ h}|^2$，右图：平面中的方向 xy。数值模拟采用 $\tau = \frac{1}{16}$，$N_x = 32$，$N_v = 64$，即使用了 8 589 934 592 个求积点。 —

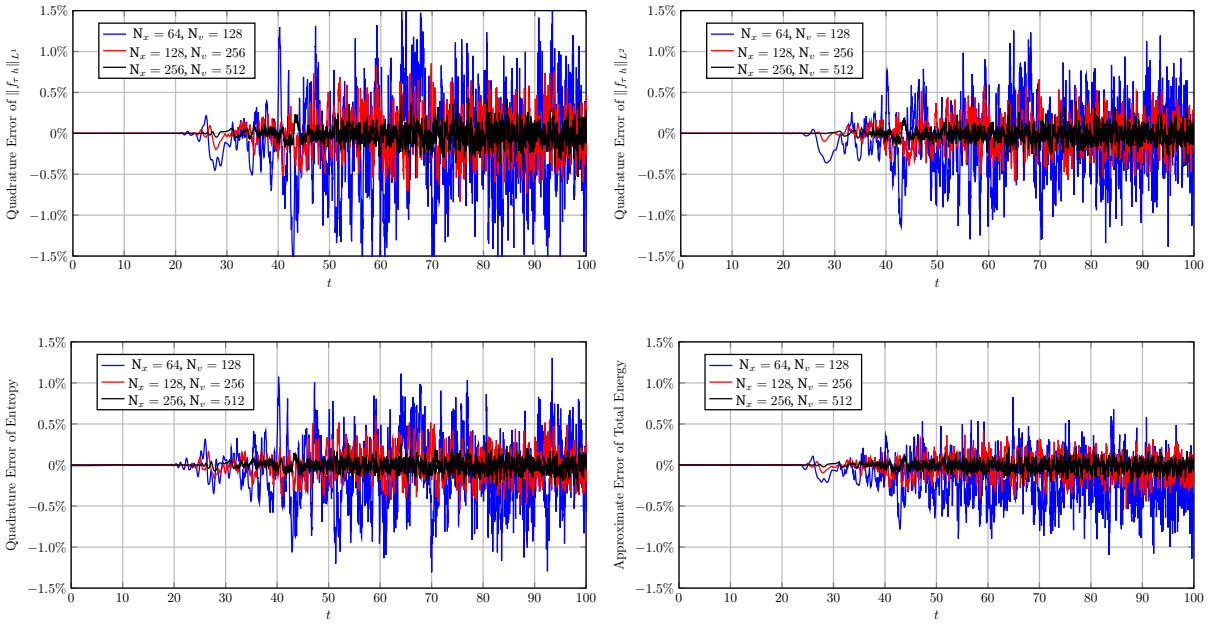
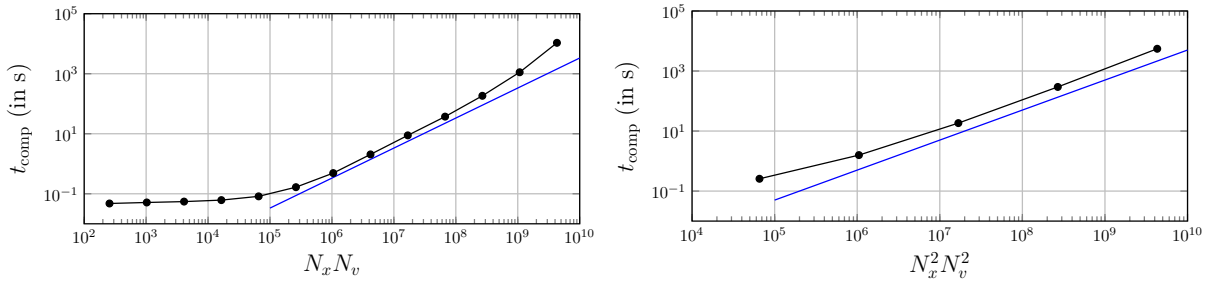


图 13：计算 L^1-范数、 L^2-范数以及以下对象的动理学熵时的求积误差：$f_{\tau,\ h}$ 在 $d = 1$ 上使用 $\tau = \frac{1}{32}$进行双流不稳定性基准测试。 这些量严格守恒，但其数值只能通过数值求积近似得到。将这些值与时间 $t = 0$处的近似值进行比较。总能量不可能严格守恒，也无法精确计算。不过，其近似误差与严格守恒量的求积误差处于同一数量级，且未观察到漂移。



(a) $d = 1$ (b) $d = 2$

图14： $n = 480$ 个时间步长以及 $N_x = 2^k$、 $N_v = 2^{k+2}$、 $k = 3$、4、5、 ...的计算时间。左： $d = 1$，单精度，Claix 集群上的一块图形处理器。右： $d = 2$，双精度，配备两块图形处理器的一个节点。复杂度随求积点数量 $N_x^d N_v^d$线性增长。对于$d = 1$ ，缓存效应在 10^9 个求积点时开始显现。

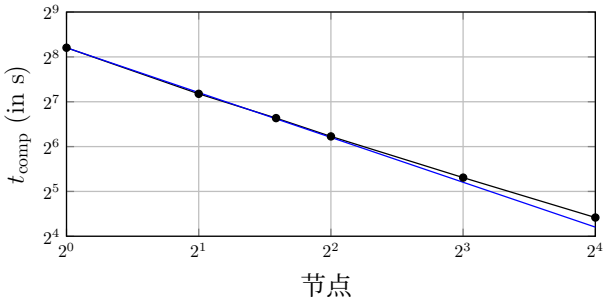


图15：对于 $d = 2$, $N_x = 64$, $N_v = 256$, $n = 480$ 个时间步长，采用双精度，并使用 Claix 集群每个节点上的两块图形处理器时，强扩展性随节点数的变化。我们观察到近乎理想的强扩展性；使用 16 个节点时，并行效率仅略有下降。

既适用于现代超大规模并行高性能计算硬件，也适用于工作站。我们相信，这些特性最终将使得在最重要的情形 $d = 3$ 下，对 Vlasov–Poisson 方程进行分辨率足够高的模拟成为可能，即使是在“Tier 2”数据中心也能实现。

该格式还可以进行多种扩展。这种方法可直接应用于多组分系统，也可扩展至完整的麦克斯韦方程组。

在这项工作中，为求简便，我们采用了完全均匀的离散化。不过，我们很容易设想采用自适应格式：$\varphi_{\tau,\ h}$ 可以采用自适应时空离散进行存储，也可以使用自适应数值求积和自适应时间步进格式。然而，与通常情况一样，自适应格式会显著增加实现有效并行化的难度，并需要复杂的负载均衡和同步机制。

非齐次右端项 $g(t,\ x,\ v)$ 也可以很容易地纳入 Vlasov方程 (1) 的考虑。在这种情况下，引理 2.3 中的解公式变为：

$$f(t,x,v) = f_0\big(\Phi_t^0(x,v)\big) + \int_0^t g\big(s, \Phi_t^s(x,v)\big)\,\mathrm{d}s. \tag{33}$$

最后一个积分可通过数值流 $\Psi_{\tau,\ h}$ 并使用梯形法则进行高效近似。这样也可以将碰撞算子纳入数值模拟。

致谢

Matthias Kirchhart 在亚琛工业大学任职期间开始了这项工作，并获得了德国研究基金会（DFG）的资助，项目编号 432219818，项目名称为“不可压缩流的涡方法”。这项工作的构思、分析和内容均在此期间完成。他目前就职于英特尔德国有限公司，该公司慷慨地允许他完成本文。

Paul Wilhelm 获得了德国国家高性能计算组织（NHR）的奖学金。

我们谨此感谢 Jan Eifert 对代码实现的帮助。没有这些资助和支持，这项工作便不可能完成。

参考文献

[1] E. Hairer、G. Wanner 和 C. Lubich。《几何数值积分：常微分方程的保结构算法*ff*。第 2 版。施普林格计算数学丛书 31。Springer，2006 年。 ISBN：3540306633。[2] L. N. Trefethen 和 J. A. C. Weideman。《指数收敛梯形法则》。载于：*SIAM* 评论 58.3（2014 年 1 月），第 385–458 页。 ISSN：0036–1445。 DOI： 10.1137/130932132。[3] G.-H. Cottet 和 P.-A. Raviart。《用于一维 Vlasov–Poisson 方程的粒子方法》。载于：*SIAM* 数值分析杂志 21.1（1984 年 6 月），第 52–76 页。 ISSN：0036–1429。 DOI： 10.1137/0721003。

[4] P. Raviart。《粒子方法分析》。载于：《流体动力学数值方法》（1985 年），第 243–324 页。doi：10.1007/BFb0074532.[5] R. Hockney 和 J. Eastwood。《使用粒子进行计算机模拟》。CRC Press，2021 年 3 月。isbn：978-1-4398-2205-0。[6] J Denavit。《在相空间中进行周期性平滑的等离子体数值模拟》。载于：《计算物理学杂志》9.1（1972 年），第 75–98 页。issn：0021-9991。doi：10.1016/0021-9991(72)90037-X.[7] B. Wang、G. H. Miller 和 P.Colella。《一种用于动力学等离子体的自适应相空间重映射粒子网格法》。载于：SIAM J. Sci. Comput. 33.6（2011 年 12 月），第 3509–3537 页。doi：10.1137/100811805。[8] G. Russo 和 J. A. Strain。《用于二维 Euler 方程的快速三角剖分涡旋方法》。载于：《计算物理学杂志》111.2（1994 年），第 291–323 页。issn：0021-9991。doi：10.1006/jcph.1994.1065.[9] R. P. Wilhelm 和 M. Kirchhart。《用于 Vlasov–Poisson 方程的插值粒子方法》。载于：《计算物理学杂志》473（2023 年），第 111720 页。issn：0021-9991。doi：10.1016/j.jcp.2022.111720。[10] A. Myers、P. Colella 和 B. Straalen。《一种用于 Vlasov–Poisson 方程的四阶粒子网格法及相空间重映射》。载于：SIAM Journal on Scientific Computing 39.9（2016 年 2 月），B467–B485。issn：1095–7197。doi：10.1137/16M105962X.[11] G.-H. Cottet、J.-M. Etancelin、F. Perignon 和 C. Picard。《用于输运方程的高阶半拉格朗日粒子方法：数值分析与实现问题》。载于：ESAIM: Mathematical Modelling and Numerical Analysis 48.4（2014 年），1029–1060。doi：10.1051/m2an/2014009.[12] N. Crouseilles、M. Mehrenberger 和 E. Sonnendr¨ucker。《用于 Vlasov 方程的守恒半拉格朗日格式》。载于：《计算物理学杂志》229.6（2010 年），第 1927–1953 页。issn：0021-9991。doi：10.1016/j.jcp.2009.11.007.[13] J. A. Rossmanith 和 D. C. Seal。《一种保持正性的高阶半拉格朗日间断 Galerkin 格式，用于 Vlasov–Poisson 方程》。载于：《计算物理学杂志》230.16（2011 年），第 6203–6232 页。issn：0021–9991。doi：doi.org/10.1016/j.jcp.2011.04.018.[14] N. Besse 和 E. Sonnendr¨ucker。《相空间非结构网格上的 Vlasov 方程半拉格朗日格式》。载于：《计算物理学杂志》 191.2 （ 2003-11 ），第 341–376 页。issn：0021–9991。doi：10.1016/S0021-9991(03)00318-8.[15] G.-H. Cottet。《用于高维 Vlasov–Poisson 系统的半拉格朗日粒子方法》。载于：《计算物理学杂志》365（2018 年），第 362–375 页。issn：0021–9991。doi：10.1016/j.jcp.2018.03.042。[16] F. Filbet 和 E. Sonnendr¨ucker。《欧拉型 Vlasov 求解器比较》。载于：Computer Physics Communications 150.3（2001 年 7 月），第 247–266 页。issn：0010–4655。doi：10.1016/S0010-4655(02)00694-X.[17] F. Filbet、E. Sonnendr¨ucker 和 P. Bertrand。《Vlasov 方程的守恒数值格式》。载于：《计算物理学杂志》172.1（2001 年 9 月），第 166–187 页。issn：0021–9991。doi：10.1006/jcph.2001.6818.[18] T. Arber 和 R. Vann。《基于欧拉网格的 Vlasov 求解器的批判性比较》。载于：《计算物理学杂志》180.1（2002 年），第 339–357 页。issn：0021–9991。doi：10.1006/jcph.2002.7098.[19] K. Kormann。《张量列格式中的半拉格朗日 Vlasov 求解器》。载于：SIAM Journal on Scientific Computing 37.4（2015 年），B613–B632。doi：10.1137/140971270.[20] L. Einkemmer。《四维中半拉格朗日间断 Galerkin 与基于样条的 Vlasov 求解器的性能比较》。载于：《计算物理学杂志》376（2019 年），第 937–951 页。issn：0021-9991。doi：10.1016/j.jcp.2018.10.012。