

# THE CONVERGENCE OF PARTICLE-IN-CELL SCHEMES FOR COSMOLOGICAL DARK MATTER SIMULATIONS

ANDREW MYERS, PHILLIP COLELLA, AND BRIAN VAN STRAALLEN  
 Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA, 94720, USA  
*Draft version June 19, 2015*

## ABSTRACT

Particle methods are a ubiquitous tool for solving the Vlasov-Poisson equation in comoving coordinates, which is used to model the gravitational evolution of dark matter in an expanding universe. However, these methods are known to produce poor results on idealized test problems, particularly at late times, after the particle trajectories have crossed. To investigate this, we have performed a series of one- and two-dimensional “Zel’dovich Pancake” calculations using the popular Particle-in-Cell (PIC) method. We find that PIC can indeed converge on these problems provided the following modifications are made. The first modification is to regularize the singular initial distribution function by introducing a small but finite artificial velocity dispersion. This process is analogous to artificial viscosity in compressible gas dynamics, and, as with artificial viscosity, the amount of regularization can be tailored so that its effect outside of a well-defined region - in this case, the high-density caustics - is small. The second modification is the introduction of a particle remapping procedure that periodically re-expresses the dark matter distribution function using a new set of particles. We describe a remapping algorithm that is third-order accurate and adaptive in phase space. This procedure prevents the accumulation of numerical errors in integrating the particle trajectories from growing large enough to significantly degrade the solution. Once both of these changes are made, PIC converges at second order on the Zel’dovich Pancake problem, even at late times, after many caustics have formed. Furthermore, the resulting scheme does not suffer from the unphysical, small-scale “clumping” phenomenon known to occur on the Pancake problem when the perturbation wave vector is not aligned with one of the Cartesian coordinate axes.

*Subject headings:* cosmology: theory — methods: numerical — dark matter

## 1. INTRODUCTION

Cosmological dark matter (DM) simulations follow the evolution of a self-gravitating, collisionless fluid in a coordinate system that expands along with the universe. This fluid can either be treated in isolation or coupled to a second baryonic fluid that is subject to its own set of physical processes. Such simulations cover a wide range of length scales, from cosmic-scale ( $\sim 10$  Gpc) calculations that cover a representative volume of the observable universe (e.g. Alimi et al. 2012), down to galactic scale ( $\sim 10$  kpc) calculations that probe the substructure of the DM fluid within Milky-Way sized ( $\sim 10^{12} M_{\odot}$ ) halos (e.g. Kuhlen et al. 2008). The results of these simulations are used to guide, interpret, or calibrate virtually all current and planned observational efforts to study the nature of dark matter and dark energy, including direct and indirect dark matter detection, galaxy redshift surveys, and weak lensing studies (Kuhlen et al. 2012).

The evolution of the DM fluid is governed by the Vlasov-Poisson equation in comoving coordinates. While it is possible to solve this set of equations using grid-based methods in phase space (Yoshikawa et al. 2013), this approach is not common in practice due to the computational expense of working in high-dimensional spaces. Typically, particle methods are used instead. These methods are based on a Lagrangian description of the Vlasov-Poisson system, in which the initial dark matter distribution function is sampled by a finite set of particles. The particle positions and velocities are then

advanced in time using the appropriate equations of motion derived from the Vlasov-Poisson equation, combined with some sort of scheme for computing the forces at the particle positions. This force calculation can be carried out in a variety of ways. In PIC methods, for example, an Eulerian representation of the density is constructed from the particle positions by deposition and the Poisson equation for the gravitational potential is solved on the resulting mesh points. An example of a cosmological DM code that uses pure PIC for its force solve is MC<sup>2</sup> (Heitmann et al. 2005); other popular choices are tree codes, e.g. pdgrav2 (Stadel 2001) and 2HOT (Warren 2013), adaptive PIC codes, e.g. ART (Kravtsov et al. 1997), RAMSES (Teyssier 2002) and Nyx (Almgren et al. 2013), or combinations of the two, e.g. GADGET2 (Springel 2005) and HACC (Habib et al. 2012). While the details of the force solves differ, all of these codes share the underlying particle discretization of the Vlasov-Poisson system. Such calculations have now been run with hundreds of millions (Alimi et al. 2012) to over a trillion (Ishiyama et al. 2012; Skillman et al. 2014) particles.

The accuracy of cosmological observations has advanced tremendously over the past few decades, and with the advent of next-generation instruments like the Large Synoptic Survey Telescope <sup>1</sup> in sight, the prospect of measuring the basic cosmological parameters of the universe to better than 1% accuracy has become conceivable (e.g. Heitmann et al. 2009, 2010). However, interpreting these observations will require theoretical predictions of

similar or better accuracy, and it is well-known that standard particle techniques produce poor results on some idealized test problems. A particularly striking example is the off-axis plane wave collapse problem of Melott et al. (1997). Melott et al. set up a plane wave perturbation that was arbitrarily aligned relative to the Cartesian coordinate axes and traced its evolution into the non-linear regime using several different particle-based numerical schemes. All of the schemes failed to converge to the correct solution when sparse particle counts were used. Instead of respecting the symmetry of the problem setup, non-physical clumping was observed in directions perpendicular to the axis of the perturbation. This result has since been reproduced by several researchers (Heitmann et al. 2005; Hahn et al. 2013).

The inability of cosmological dark matter simulations to converge on this problem is troubling. It is not clear to what extent these problems manifest themselves in more realistic calculations. Code comparisons involving more realistic initial conditions generally show encouraging agreement on physical observables for dark-matter only problems (e.g. Heitmann et al. 2005; Heitmann et al. 2008; Kim et al. 2014). However, all of these codes solve the Vlasov-Poisson system using particle methods, differing mainly in the way the force solve is carried out. Thus, these comparisons cannot rule out the hypothesis that common assumptions are giving rise to common errors. A related issue is the small-scale artificial fragmentation observed in particle simulations of warm dark matter (Wang & White 2007). This fragmentation may be related to the “clumping” effect observed in the off-axis pancake problem, but this needs to be demonstrated explicitly. Additionally, as pointed out by Hahn et al. (2013), we cannot be sure that cold dark matter simulations do not suffer from the same problems, but are simply hidden by the presence of initial power on small scales.

In light of this, a particle method that can be rigorously shown to converge - even on a simple test problem like the Zel’dovich Pancake - is desirable. The problem of improving the accuracy of particle methods on cosmology calculations has been considered by Hahn et al. (2013), who presented an improved PIC scheme that exploits the continuity of the infinitesimally thin DM sheet in phase space. We take an alternative approach, based on making the cosmology problem look like more like its cousin in electrostatic plasma physics, where PIC schemes can be shown to converge and a rigorous convergence theory exists (Wang et al. 2011, hereafter WMC2011). This theory suggests a number of modifications that improve the accuracy of PIC on the Zel’dovich Pancake problem. **The first modification is to remove the singularity in the initial data. In plasma applications, the initial conditions are typically warm, meaning that the initial distribution function has some finite velocity dispersion around a mean value. DM calculations, on the other hand, almost always assume perfectly cold initial conditions, in which the initial distribution function is a  $\delta$  function in velocity space. This is done for sound physical reasons: given that the dark matter velocity dispersion scales inversely with the expansion factor of the universe, the present-day dispersion of any plausible DM candidate would indeed be tiny**

compared to the velocities that arise during collapse (Hahn et al. 2013). However, singular initial data gives rise to resolution-dependent maximum densities in dark matter caustics, which complicates code verification and leads to slow convergence in the gravitational field and potential.

A natural approach to dealing with this singularity is to *regularize* the initial conditions by introducing a finite, artificial velocity dispersion to the initial DM distribution function. A finite initial distribution function leads to bounded density peaks that are resolvable by a finite number of computational elements. The artificial velocity dispersion should be small, so that its effect far from the density maxima is negligible, and so that it can be resolved by a relatively small number of additional particles. An instructive analogy is to the use of an artificial viscosity in shock-capturing schemes for compressible gas dynamics. For Reynolds numbers characteristic of supersonic flows in astrophysics, the molecular mean free path of the gas is tiny compared to the other length scales of interest, and thus the effects of viscosity should be negligible. However, neglecting the viscous terms entirely leads to numerical instabilities that can contaminate the solution far from any shocks. A common solution is to introduce an artificially large numerical viscosity that smooths out the solution near a shock front over a few grid cells. Away from the shock, the integrity of the solution is preserved. Likewise, in structure formation, the present-day microscopic velocity dispersion of dark matter particles is negligibly tiny compared to the bulk velocities that arise from gravitational collapse. However, ignoring the velocity dispersion completely leads to resolution-dependent results at the caustic locations. Using an artificially large initial velocity dispersion smooths out the dark matter caustics over a length scale that is numerically resolvable, preserving the qualitative features of the dark matter density distribution without affecting the solution outside of the caustics.

The second modification suggested by the error analysis of WMC2011 is to periodically *remap* the DM particles in phase space. This addresses a downside of particle methods, which is that numerical errors made in integrating the particle trajectories tend to compound with each successive time step. In electrostatic PIC, this phenomenon manifests itself as an exponentially growing term in the stability error for the electric field (WMC2011, Equation 3.12). This term eventually overwhelms the solution as the simulation evolves. To mitigate this, the distribution function must be periodically re-expressed using a new set of particles before integration errors have a chance to accumulate. This procedure is crucial for obtaining quality solutions to plasma problems for long time integrations. Note that, as currently constructed, this procedure requires the above regularization procedure, in that we must be able to generate a well-defined distribution function in phase space corresponding to a given particle distribution.

In this paper, we show that these modifications improve the accuracy of PIC on the Zel’dovich Pancake test

problem. The structure of the paper is as follows. We begin by reviewing the cosmological Vlasov-Poisson system and the standard PIC algorithm for solving it in Section 2. In Section 3, we perform a series of convergence studies that test PIC's ability to track the gravitational collapse of a single, plane-wave perturbation to late times. We perform two versions of this test, first with only one spatial dimension, and second in two spatial dimensions, with the axis of the perturbation misaligned with the Cartesian coordinate axes. We confirm that standard PIC with singular initial data converges slowly at late times in 1D, and that it suffers from unphysical clumping on the 2D, oblique problem, regardless of the number of particles per Poisson cell employed. Next, in Section 4, we consider the same numerical method, using initial data that has been regularized via the introduction of a finite artificial velocity dispersion. We show that, given finite initial conditions, the standard PIC technique does converge at 2nd order in 1D. We also show that, in the limit that the artificial velocity dispersion goes to zero, we recover the solution to the problem with perfectly cold initial conditions. However, even with regularization, we still find evidence of significant artificial clumping on the 2D, oblique problem. Finally, in Section 5, we introduce particle remapping. We perform this remapping on an adaptive set of grids that automatically tracks the evolution of the dark matter distribution function in phase space as the universe expands. Our results using both regularization and remapping are shown in Section 6. We show that, when both techniques are employed, 2nd order convergence results are still obtained in 1D, and the solution is not affected by the artificial clumping present in other cases. We summarize our conclusions in Section 7.

## 2. EQUATIONS AND ALGORITHMS

### 2.1. The Vlasov-Poisson system

We work with a non-dimensional form of the Vlasov-Poisson equation in comoving coordinates:

$$\frac{\partial f}{\partial t} = -\frac{\mathbf{v}}{a} \cdot \frac{\partial f}{\partial \mathbf{x}} + \left[ \left( \frac{\dot{a}}{a} \right) \mathbf{v} + \frac{1}{a} \nabla \phi \right] \cdot \frac{\partial f}{\partial \mathbf{v}}. \quad (1)$$

Here,  $f(\mathbf{x}, \mathbf{v})$  is the dark matter distribution function in  $2D$ -dimensional phase space  $(\mathbf{x}, \mathbf{v}) \in \mathbb{R}^D \times \mathbb{R}^D$ , where  $D$  is the number of spatial dimensions under consideration.  $\mathbf{x}$  and  $\mathbf{v}$  are the usual comoving position and peculiar velocity coordinates (Peebles 1993), related to the proper position coordinate  $\mathbf{r}$  by

$$\begin{aligned} \mathbf{x} &= a(t)^{-1} \mathbf{r} \\ \mathbf{v} &= a(t) \dot{\mathbf{x}}. \end{aligned} \quad (2)$$

The expansion of the universe is described by the time-dependent scale factor  $a(t)$ . In general, the time evolution of  $a(t)$  is determined by the assumed cosmological parameters. In this paper, we specialize to a flat, critical-density, matter-only universe, so that

$$a(t) = \left( \frac{3t}{2} \right)^{2/3}. \quad (3)$$

The gravitational potential  $\phi$  obeys the Poisson equation:

tion:

$$\nabla^2 \phi = \frac{3}{2a} (\rho - 1), \quad (4)$$

where  $\rho$  is the comoving dark matter density. We complete the specification of the problem by adopting periodic boundary conditions on the domain  $\mathbf{x} \in (0, 1)^D$  and supplying appropriate initial conditions,  $f(\mathbf{x}, \mathbf{v}, t = t_{\text{ini}})$ .

### 2.2. Particle Discretization

In particle methods for numerically solving Equation 1, the initial distribution function is discretized by a set of Lagrangian particles,  $\mathbb{P}$ . Given  $f(\mathbf{x}, \mathbf{v}, t_{\text{ini}})$ , we must choose a set of particles with initial positions  $\mathbf{x}_p^i$ , velocities  $\mathbf{v}_p^i$ , and masses  $m_p$  such that

$$f(\mathbf{x}, \mathbf{v}, t_{\text{ini}}) \approx \sum_{p \in \mathbb{P}} m_p \delta(\mathbf{x} - \mathbf{x}_p^i) \delta(\mathbf{v} - \mathbf{v}_p^i). \quad (5)$$

From these initial coordinates, the particles evolve according to the appropriate equations of motion. The trajectory  $(\mathbf{x}_p(t), \mathbf{v}_p(t))$  of each particle can be obtained by substituting Equation 5 into Equation 1. The result is the following system of ODEs:

$$\begin{aligned} \frac{dm_p}{dt} &= 0 \\ \frac{d\mathbf{x}_p}{dt} &= \frac{1}{a} \mathbf{v}_p \\ \frac{d\mathbf{v}_p}{dt} &= -\frac{\dot{a}}{a} \mathbf{v}_p + \frac{1}{a} \mathbf{g}_p, \end{aligned} \quad (6)$$

where  $\mathbf{g}_p$  is the acceleration on particle  $p$  induced by the dark matter distribution. It is important to note that these particles have nothing to do with the actual dark matter particles whose statistics are described by the Vlasov-Poisson equation; they are simply a set of interpolating points from which the distribution function can be recovered.

### 2.3. The PIC Update

Once the initial particles have been generated, we advance their positions and velocities using a standard Particle-in-Cell technique (Hockney & Eastwood 1981). We write the particle comoving positions and peculiar velocities at time  $t^n$  as  $\mathbf{x}_p^n$  and  $\mathbf{v}_p^n$ , where  $n$  is the time step number. We also assume that we know the gravitational field at the particle positions  $\mathbf{g}_p^n$  at the same time index, either by performing an initial Poisson solve or by storing the result from the previous time step. Given that information, we advance the particles to time  $t^{n+1}$  as follows:

**Particle Kick.** For our time integration scheme, we use the second-order and symplectic Kick-Drift-Kick sequence from Miniati & Colella (2007). Note that this scheme is self-starting in the sense that it does not require any information from time steps prior to  $n$  to complete the update - an important feature given our use of remapping below. We begin by updating the particle velocities to the half-time index  $n + 1/2$ :

$$\mathbf{v}_p^{n+1/2} = \frac{a^n}{a^{n+1/2}} \mathbf{v}_p^n + \frac{1}{a^{n+1/2}} \mathbf{g}_p^n \frac{\Delta t}{2}. \quad (7)$$

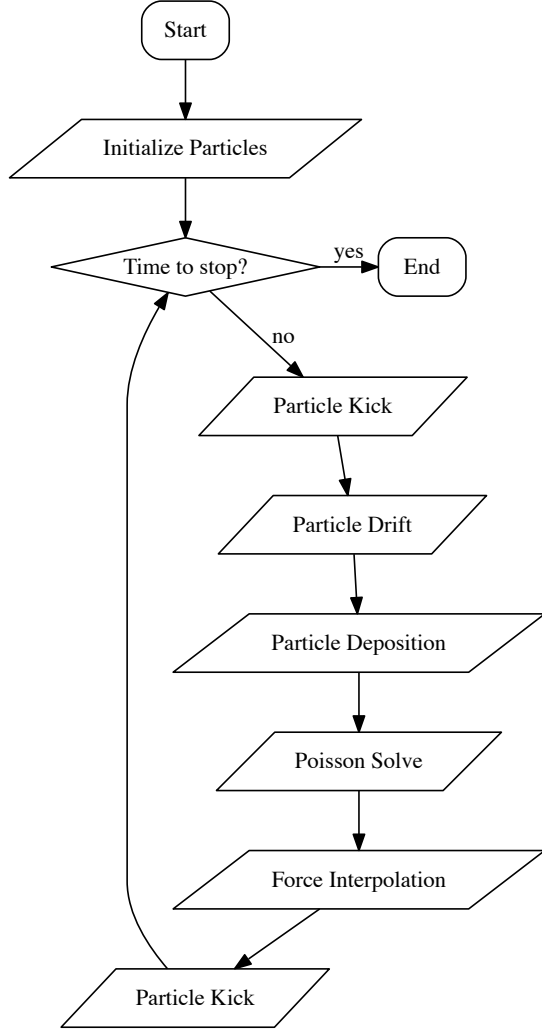


FIG. 1.— The overall control flow of a PIC calculation

Here,  $\Delta t$  is the time step, and  $a^n$  and  $a^{n+1/2}$  are the expansion factors computed at times  $t^n$  and  $t^{n+1/2}$ .

**Particle Drift.** Next, the particle positions are advanced using the half-time velocity:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{1}{a^{n+1/2}} \mathbf{v}_p^{n+1/2} \Delta t. \quad (8)$$

The final update to the particle velocities cannot be completed until we compute the new forces at time index  $n+1$ .

**Particle Deposition.** To do so, we compute the density at the mesh points  $\mathbf{x}_i = (\mathbf{i} + 1/2) \Delta \mathbf{x}$ , where  $\mathbf{i} \in \mathbb{Z}^D$  are the cell indices:

$$\rho_i^{n+1} = \sum_p \left( \frac{m_p}{V_i} \right) \mathbf{W}_{\text{CIC}} \left( \frac{\mathbf{x}_i - \mathbf{x}_p^{n+1}}{\Delta \mathbf{x}} \right). \quad (9)$$

Here,  $V_i = \Delta x^D$  is the volume of cell  $i$  and  $\mathbf{W}_{\text{CIC}}(\mathbf{x})$  is the  $D$ -dimensional cloud-in-cell (CIC) interpolating

function:

$$\mathbf{W}_{\text{CIC}}(\mathbf{x}) = \prod_{d=1}^D W_{\text{CIC}}(x_d), \quad (10)$$

$$W_{\text{CIC}}(x) = \begin{cases} 1 - |x|, & 0 < |x| < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Note that in general, we do not use the same mesh spacing for the particle discretization and the Poisson mesh, i.e.  $\Delta x \neq h_x$ .

**Poisson Solve.** The next step is to solve the Poisson equation for the gravitational potential at the same grid points on which the density is defined. We discretize the Laplacian operator using the standard  $2D+1$  point centered difference approximation:

$$-\sum_{d=1}^D \frac{\phi_{i+e^d}^{n+1} - 2\phi_i^{n+1} + \phi_{i-e^d}^{n+1}}{\Delta x^2} = \frac{3}{2a^{n+1}} (\rho_i^{n+1} - 1). \quad (12)$$

To solve the resulting linear system, we use the geometric multigrid package in Chombo (Adams et al. 2015), using Gauss-Seidel with Red-Black ordering as the smoother. We set the solver tolerance to  $10^{-9}$  - small enough its precise value does not affect our convergence results below. After iterating to convergence, the gravitational field  $\mathbf{g} = -\nabla \phi$  can be computed at the same grid points as

$$\mathbf{g}_i^{n+1} = -\frac{\phi_{i+e^d}^{n+1} - \phi_{i-e^d}^{n+1}}{2\Delta x}. \quad (13)$$

**Force Interpolation.** Next, we interpolate the field back to the particle positions using the same interpolating function as in the deposition step:

$$\mathbf{g}_p^{n+1} = \sum_j \mathbf{g}_j V_j \mathbf{W}_{\text{CIC}} \left( \frac{\mathbf{x}_j - \mathbf{x}_p^{n+1}}{\Delta \mathbf{x}} \right). \quad (14)$$

**Particle Kick.** Finally, we complete the update to the particle positions as:

$$\mathbf{v}_p^{n+1} = \frac{a^{n+1/2}}{a^{n+1}} \mathbf{v}_p^{n+1/2} + \frac{1}{a^{n+1}} \mathbf{g}_p^{n+1} \frac{\Delta t}{2}. \quad (15)$$

#### 2.4. Time stepping strategy

In this paper, the size of the time step  $\Delta t$  is controlled by two factors. First, we limit the factor by which the background can expand in a single time step. The time step associated with the expansion factor is:

$$\Delta t_{\text{exp}} = C_{\text{exp}} \left( \frac{a}{\dot{a}} \right), \quad (16)$$

where  $C_{\text{exp}}$  is a tunable parameter.

Second, we enforce a Courant-Friedrichs-Lewy (CFL)-type constraint, which limits the distance that a particle can travel in a single time step to be some fraction of  $\Delta x$ . The time step associated with this constraint is:

$$\Delta t_{\text{part}} = C_{\text{part}} \frac{\Delta x}{\max(|\mathbf{v}_p|)}. \quad (17)$$

In this paper, we take  $C_{\text{part}} = 0.5$ . We use the smaller of these two time steps to advance the simulation:

$$\Delta t = \min(\Delta t_{\text{exp}}, \Delta t_{\text{part}}). \quad (18)$$

This approach to time stepping is quite similar to that in, e.g., Almgren et al. (2013). In the runs presented in this paper, the expansion of the background typically controls  $\Delta t$  at early times, while the particle CFL constraint controls it at late times.

Given an initial set of particles, the above equations describe how to advance the system to an arbitrary later time. The overall control flow of the PIC procedure is illustrated in Figure 2.3. We have implemented the above algorithm, as well as the remapping procedure described in Section 5, using the Chombo<sup>2</sup> software framework for partial differential equations, using standard message passing with block-structured domain decomposition for parallelization. This code, as well as our data analysis scripts, are available online<sup>3</sup>.

### 3. THE ZEL'DOVICH PANCAKE

We now study the convergence of the PIC method for plane-wave initial conditions. The gravitational collapse of a plane-wave perturbation in an expanding background, sometimes called the ‘‘Zel’dovich Pancake’’ problem because of the flattened structures it produces, is a common test case for cosmological codes (e.g. Bryan et al. 1995; Kravtsov et al. 1997; Miniati & Colella 2007; Hopkins 2014) for several reasons. At early times (before any matter parcels cross) an exact solution exists in 1D, making the problem valuable for code validation. After the first shell crossing, the problem still provides a valuable test on a code’s ability to deal with strong density contrasts and poorly resolved features. Finally, while the problem is clearly idealized, it still physically relevant, as the collapse of a single Fourier mode forms the basis for more complex structure formation calculations.

#### 3.1. Initial conditions

The initial distribution function is taken to be an infinitesimally thin sheet in phase space:

$$f(\mathbf{x}, \mathbf{v}, t_{\text{ini}}) = \rho_z(\mathbf{x}, t_{\text{ini}}) \delta(\mathbf{v} - \mathbf{v}_z(\mathbf{x}, t_{\text{ini}})). \quad (19)$$

The initial density  $\rho_z(\mathbf{x}, t_{\text{ini}})$  and velocity  $\mathbf{v}_z(\mathbf{x}, t_{\text{ini}})$  are then computed using the Zel’dovich approximation. For example, if the matter parcels of the unperturbed DM fluid are labelled by their comoving Lagrangian coordinates  $\mathbf{q}$ , and we apply a sinusoidal perturbation of the form

$$\mathbf{S}(\mathbf{q}) = \frac{5A}{2} \sin(\mathbf{k} \cdot \mathbf{q}) \hat{\mathbf{k}}, \quad (20)$$

where  $A$  is the amplitude and  $\mathbf{k}$  the wavenumber, then (in our adopted Einstein-de Sitter cosmology) the comoving position and peculiar velocity of the fluid at time  $t$  are

$$\mathbf{x}(t) = \mathbf{q} + aA \sin(\mathbf{k} \cdot \mathbf{q}) \hat{\mathbf{k}} \quad (21)$$

$$\mathbf{v}(t) = a\dot{a}A \sin(\mathbf{k} \cdot \mathbf{q}) \hat{\mathbf{k}}. \quad (22)$$

The corresponding density is

$$\rho(\mathbf{q}, t) = \frac{1}{1 + aA|\mathbf{k}| \cos(\mathbf{k} \cdot \mathbf{q})}, \quad (23)$$

which can be converted to Eulerian coordinates  $\mathbf{x}$  using Equation 21.

<sup>2</sup> <https://commons.lbl.gov/display/chombo>

<sup>3</sup> <https://bitbucket.org/atmyers/cosmologicalpic>

TABLE 1  
SUMMARY OF PARAMETERS  
FOR THE SINGULAR, 1D RUNS

$N_{\text{cells}}$	$n_x$	$C_{\text{exp}}$
256	128	$1.0 \times 10^{-2}$
512	128	$5.0 \times 10^{-3}$
1024	128	$2.5 \times 10^{-3}$
2048	128	$1.25 \times 10^{-2}$

The above equations are valid until the first caustic forms, which happens at the expansion factor

$$a_{\text{caustic}}(t) = \frac{1}{A|\mathbf{k}|}. \quad (24)$$

At that point, the velocity becomes multi-valued and the density diverges at the positions corresponding to  $\cos(\mathbf{k} \cdot \mathbf{q}) = 1$ . Prior to that time, we can use the above equation to set initial conditions. In particular, we represent the above initial conditions with a collection of  $N_p$  equal-mass particles labelled by the Lagrangian coordinates

$$\mathbf{q}_p = (\mathbf{i} + 1/2) h_x, \quad (25)$$

where  $h_x = 1/N_p$  is the initial particle spacing (which is not necessarily the same as the mesh spacing for the Poisson solve,  $\Delta x$ ),  $\mathbf{i} \in \mathbb{Z}^D$ , and  $m_p = 1/N_p$  is the particle mass. We start the integration at time  $t_{\text{ini}}$ , corresponding to expansion factor  $a_{\text{ini}}$ . At that time, the Eulerian positions and velocities of the particles are

$$\mathbf{x}_p^i = \mathbf{q}_p + a_{\text{ini}} A \sin(\mathbf{k} \cdot \mathbf{q}_p) \hat{\mathbf{k}} \quad (26)$$

$$\mathbf{v}_p^i = a_{\text{ini}} \dot{a}_{\text{ini}} A \sin(\mathbf{k} \cdot \mathbf{q}_p) \hat{\mathbf{k}}. \quad (27)$$

Note that after applying a perturbation, neither the particle positions nor the velocities are laid out on a Cartesian grid. This procedure, in which the Zel’dovich approximation is used to construct initial particle positions and velocities and the initial distribution function is perfectly cold, is standard in cosmological dark matter simulations (e.g. Hahn & Abel 2011).

#### 3.2. 1D results

We begin by performing a resolution study on a 1D problem setup. This is the easiest test for the method to pass, since some numerical instabilities may only be possible in multi-dimensional problems. However, this test will help us verify that we have implemented the above scheme correctly, and it may also provide clues about the behavior of the higher dimensional case.

For these runs, we set the perturbation wavenumber to  $2\pi$ , the fundamental mode of the computational box. We initialize the problem at  $a_{\text{ini}} = 1/200$  and choose the perturbation amplitude so that shell crossing occurs at  $a_{\text{caustic}} = 0.1$ . We then evolve the simulations until  $a_{\text{stop}} = 1$ , using the scheme described in Section 2.3. The simulation results are dumped out every  $\Delta a = 0.01$ , i.e. at  $a = 0.01, 0.02, \dots, 0.99, 1.00$ . We fix the number of particles per Poisson cell,  $n_x$ , where  $n_x = N_p/N_{\text{cells}}$ , to 128, and vary the number of Poisson cells in the problem domain  $N_{\text{cells}} = 1/\Delta x$  over the range  $\{256, 512,$

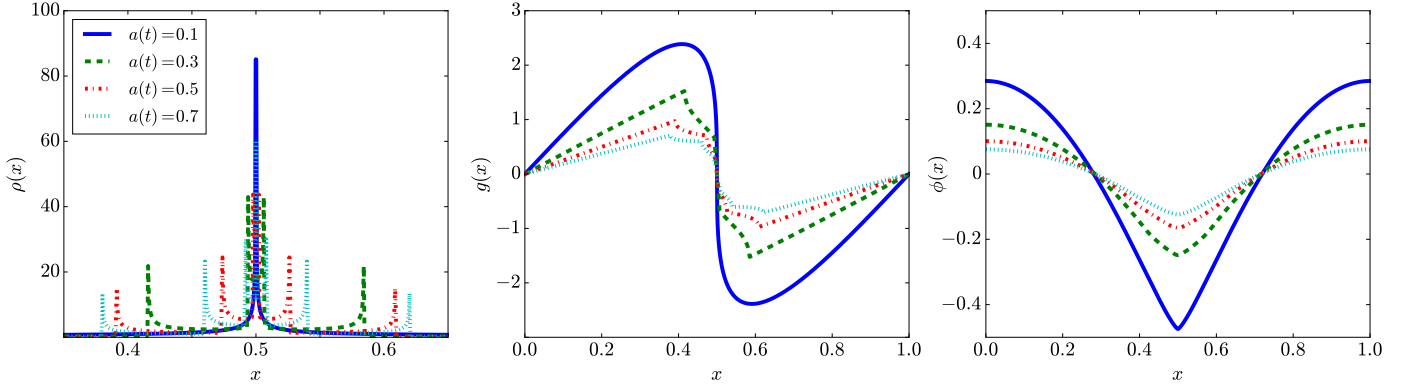


FIG. 2.— The time evolution of the solution to the 1D, singular problem. The left panel shows the density, the middle panel the gravitational field, and the right panel the potential. The plotted curves are from the run with  $N_{\text{cells}} = 2048$ . For the gravitational field and the potential, the  $N_{\text{cells}} = 1024$  run would be indistinguishable from the plotted solutions at the scales shown. The solutions for the density, on the other hand, are not converged; see Figure 3. The different lines correspond to the results at different expansion factors: solid blue line -  $a(t) = 0.1$ ; dashed green line -  $a(t) = 0.3$ ; dashed-dotted red line -  $a(t) = 0.5$ ; dotted cyan line -  $a(t) = 0.7$ .

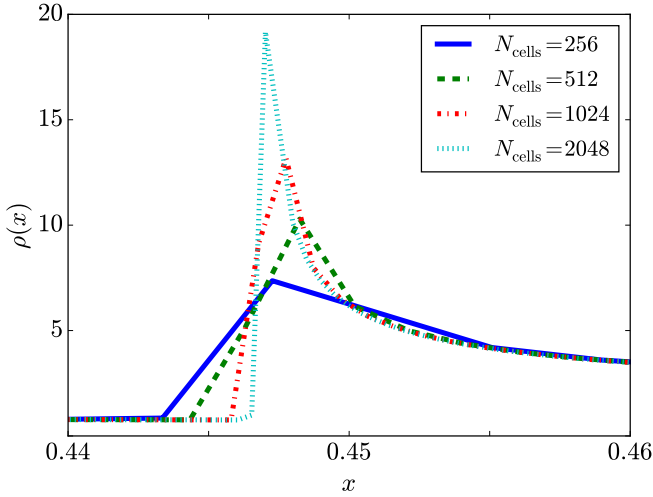


FIG. 3.— A zoomed-in view of one of the outer density caustics in the 1D, singular problem, taken at  $a(t) = 1$ . The different colors correspond to different resolutions: solid blue line -  $N_{\text{cells}} = 256$ ; dashed green line -  $N_{\text{cells}} = 512$ ; dashed-dotted red line -  $N_{\text{cells}} = 1024$ ; dotted cyan line -  $N_{\text{cells}} = 2048$ .

1024, 2048}. Every time we decrease the mesh spacing, we also decrease the parameter  $C_{\text{exp}}$  by a factor of 2. These parameters are summarized in Table 1. Note that our use of 128 particles per cell is something of a best-case scenario; most calculations in cosmology use significantly fewer particles. The resulting solutions are shown at selected times in Figure 2. As expected, at  $a(t) = 0.1$  the particle trajectories cross each other and a single dark matter caustic forms. As the particles continue to evolve in phase space, secondary and tertiary caustics form. Each caustic is marked by a peak in the density profile and cusp in the gravitational field. The density peaks in the dark matter caustics are clearly not converged, as is to be expected with singular initial data (see Figure 3). Visually, however, the gravitational field and the potential appear to approach well-defined solutions.

To verify this, we compute the convergence rates for the density, gravitational field, and potential. While an analytic solution to the Zel’dovich Pancake problem exists prior to  $a_{\text{caustic}}$ , we are also interested in the con-

vergence behavior at late times, and it is convenient to have an error metric that applies to both  $a < a_{\text{caustic}}$  and  $a > a_{\text{caustic}}$ . Additionally, as we will later want to apply our metric to regularized and remapped runs, we cannot compare particle quantities directly. Instead, we must focus on the grid-defined quantities used in the PIC update:  $\mathbf{g}_i$ ,  $\rho_i$ , and  $\phi_i$ . We therefore use Richardson extrapolation to compute our error estimates. If  $Q^h$  is one of these grid quantities, and  $Q^{2h}$  is the same quantity computed with *all* the discrete elements ( $\Delta x$ ,  $h_x$ ,  $\Delta t$ ,  $C_{\text{exp}}$ ) coarsened by a factor of 2, then we define the relative solution error as

$$e^h = |Q^h - Q^{2h}|. \quad (28)$$

To compare solutions with different numbers of grid points, we average the finer solution down to the resolution of the coarse solution. The order  $q$  of the method is then

$$q = \log_2 \left( \frac{\|e^{2h}\|}{\|e^h\|} \right), \quad (29)$$

where  $\|x\|$  is one of the  $L_1$ ,  $L_2$ , or  $L_\infty$  norms of  $x$ . In what follows, we use the symbols  $q_\rho$ ,  $q_g$ , and  $q_\phi$  to denote the order of convergence in the density, gravitational field, and potential, respectively.

The resulting convergence rates are displayed as a function of time in Figure 4. Formally, our adopted PIC scheme should be second-order accurate in space, and it indeed achieves this up until first shell crossing, at least for the gravitational field and the potential. After  $a_{\text{caustic}}$ , however, the convergence rates for  $\rho_i$  and  $\mathbf{g}_i$  drop dramatically, in all of the norms we consider. The solution for  $\rho_i$  does not converge at all in the max norm, while the solution for  $\mathbf{g}_i$  converges only slowly.

In a sense, it is not surprising that we see poor convergence in these metrics after  $a_{\text{caustic}}$ . With perfectly cold initial conditions, there is nothing to limit the peak densities in the dark matter caustics that form after  $a_{\text{caustic}}$  other than the finite numerical resolution employed. Indeed, we see in Figure 4 that the  $L_\infty$ -norm of  $\rho$  actually diverges with  $q_\rho < 0$  after  $a_{\text{caustic}}$ . This resolution dependence in turn contaminates the results for the gravitational field and potential. While this poor convergence in 1D is limited to the caustic positions, it points towards more significant problems in 2D or 3D calculations,

TABLE 2  
SUMMARY OF PARAMETERS  
FOR THE SINGULAR, 2D RUNS

$N_{\text{cells}}$	$n_x$	$C_{\text{exp}}$
$128^2$	$1/2^2$	$2.0 \times 10^{-2}$
$128^2$	$1^2$	$2.0 \times 10^{-2}$
$128^2$	$16^2$	$2.0 \times 10^{-2}$
$256^2$	$1/2^2$	$1.0 \times 10^{-2}$
$256^2$	$1^2$	$1.0 \times 10^{-2}$
$256^2$	$16^2$	$1.0 \times 10^{-2}$

where the symmetry constraints of the initial conditions are not automatically met.

### 3.3. 2D Results

In this section, we examine a 2D analog of the non axis-aligned plane wave collapse problem considered by Melott et al. (1997). This setup, in which the perturbation wave vector is not aligned with any of the coordinate axes, is known to produce unphysical clumping along directions in which the density should be constant given the symmetry of the initial conditions. In this section, we confirm that this effect is also present in 2D using our standard PIC method. We set the perturbation wavenumber  $\mathbf{k} = (k_x, k_y)$  to be  $(2, 5)$  in units of the fundamental mode. Otherwise, the initial conditions are the same as in our 1D resolution study:  $a_{\text{ini}} = 1/200$ ,  $a_{\text{caustic}} = 0.1$ , and  $a_{\text{stop}} = 1$ . Throughout the rest of this paper, we refer to these initial conditions as the “oblique” pancake.

We perform six runs in total, as summarized in Table 2. We vary the number of cells in the Poisson mesh over  $\{128^2, 256^2\}$ , and run the problem with  $n_x = 1/2^2$ , 1, and  $16^2$  particles per Poisson cell. The particles are initially arranged within the Poisson cells in the usual way. In the run with  $1/2^2$  particles per cell, we coarsen the Poisson mesh by a factor of two and place particles at centers of each cell in the resulting coarsened mesh. In the run with  $16^2$  particles per cell, we refine the mesh, instead.

The resulting particle  $x$  and  $y$  positions at  $a = 1$  are shown in Figure 5. We only display the high resolution results in Figure 5; the low resolution results are qualitatively similar. By the symmetry of the problem, the particle density should be constant along directions perpendicular to the axis of the perturbation. However, all of the runs show signs of small-scale fragmentation that do not obey this symmetry requirement, and thus cannot be valid solutions to Vlasov-Poisson for the given initial conditions. This effect is not limited to runs with sparse particle counts, and is even present in the run with the  $16^2$  particles per Poisson cell - a much larger number than typically used in realistic dark matter simulations. We conclude that simply using very large particle counts with the basic PIC scheme is not sufficient to prevent these errors.

This clumping effect is present in the grid-deposited density field, as well. In Figure 6, we construct a 2D density field by depositing the particles onto a  $256^2$  mesh using cloud-in-cell deposition (Equation 9). This Figure was constructed using the run with  $16^2$  particles per cell; the other runs produce an even stronger effect. Figure 7

shows the corresponding density field from the  $16^2$  particle per cell, high resolution run, where we generate the density by deposition onto a  $512^2$  mesh. Both figures shows signs of unphysical fragmentation along the dark matter caustics.

It is possible that this artificial fragmentation is related to the divergence of the density at the caustic locations in 1D observed in Section 3.2. To test this hypothesis, we introduce a regularization procedure that limits the maximum density at the caustic positions. The results of this test are described in the next section.

## 4. THE REGULARIZED PANCAKE

### 4.1. Initial Conditions

The above initial conditions are singular in the sense that the initial distribution function is a delta function in velocity space, centered at  $\mathbf{v} = \mathbf{v}_z$ . With delta function initial conditions, the solution to the Zel’dovich pancake problem at late times contains singularities in the density, and corresponding discontinuities in the derivative of the gravitational field, at the positions of the dark matter caustics. When solving this problem numerically with PIC or other particle methods, however, the peak densities obtained in the numerical solution will be determined by the finite amount of resolution employed. In effect, the mass deposition kernel, which spreads out the mass in a given particle over a few grid cells, provides a regularization that removes these singularities; however, it does so in a relatively uncontrolled way that is explicitly resolution-dependent. As a consequence, with perfectly cold initial data, it is not possible to separate numerical errors, which should improve with mesh refinement, from parameters of the numerical model, which should have well-defined effects on the solution once a converged answer is found.

Instead of letting the mesh spacing provide the regularization, we investigate an alternative regularization procedure that removes the singularities in the solution by introducing a finite initial velocity dispersion,  $\sigma_i$ , to equation 19. We choose a Gaussian form for the velocity profile, so that the initial distribution becomes:

$$f(\mathbf{x}, \mathbf{v}, t_{\text{ini}}) = \left( \frac{1}{2\pi\sigma_i^2} \right)^{D/2} \exp \left[ -\frac{(\mathbf{v} - \mathbf{v}_z(\mathbf{x}, t_{\text{ini}}))^2}{2\sigma_i^2} \right] \times \frac{\langle \rho \rangle}{1 + aA|\mathbf{k}| \cos(\mathbf{k} \cdot \mathbf{q})}. \quad (30)$$

This process makes the initial conditions more closely resemble those used in electrostatic PIC calculations, for which the expected convergence rates can be demonstrated. In the limit  $\sigma_i \rightarrow 0$ , Equation 19 is recovered. Using these modified initial conditions, we then evolve the regularized version of the Zel’dovich Pancake problem using the same PIC method as before. To generate the initial particles, we sample Equation 30 with a set of particles that are initially laid

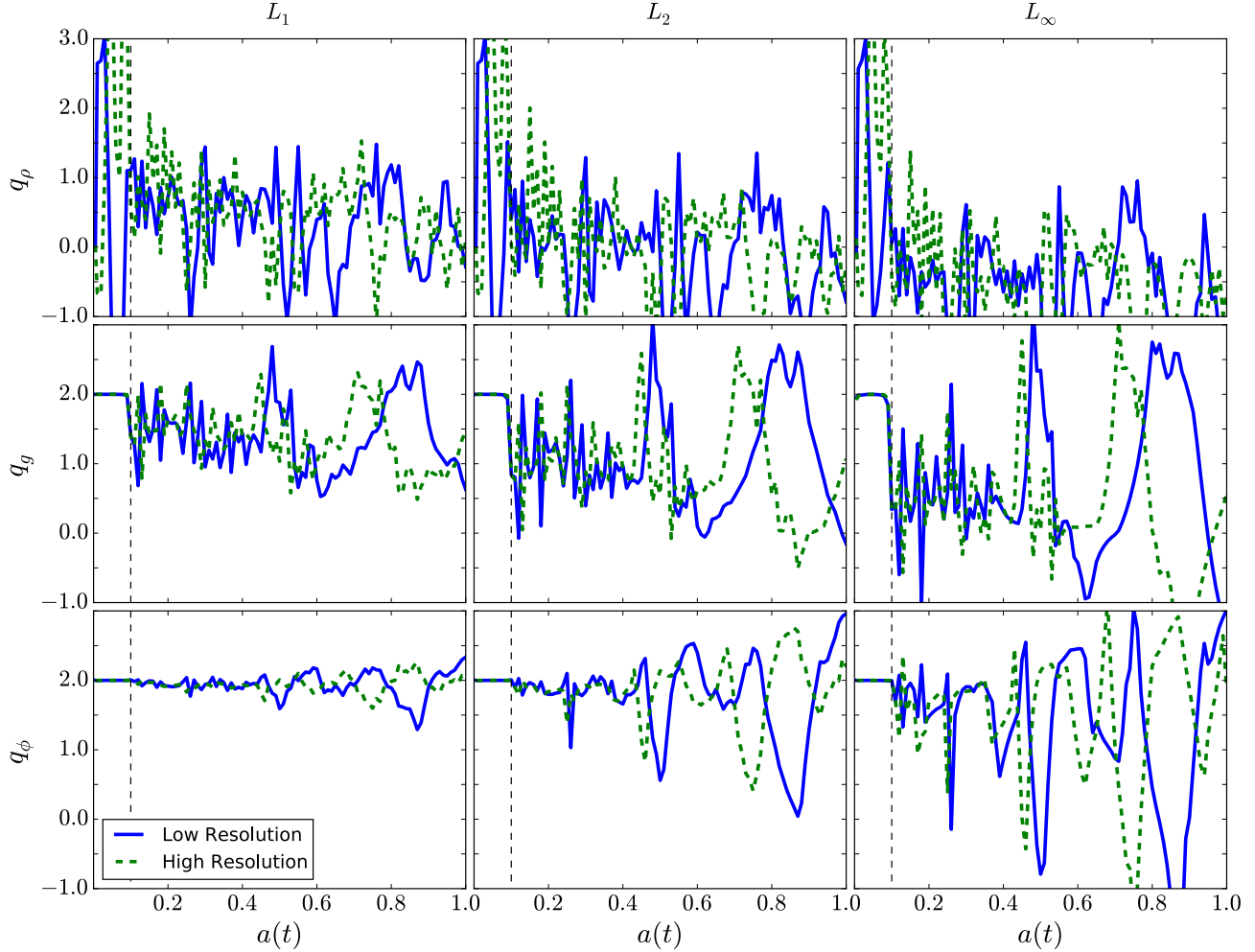


FIG. 4.— Convergence results for the singular version of the 1D pancake problem. The plotted curves show the Richardson-estimated order versus the expansion factor  $a(t)$ . For the solid blue line, the order was estimated using the runs with  $N_{\text{cells}} = 256, 512$ , and  $1024$ , while the dashed green line shows the same quantity for resolutions  $N_{\text{cells}} = 512, 1024$ , and  $2048$ . The top row shows the rates computed for the density, the middle row the gravitational field, and the bottom row the potential. Each column shows the rate using a different error norm; from left to right, we show the  $L_1$ ,  $L_2$ , and  $L_\infty$  norms, respectively. The expansion factor at which the first caustic forms is indicated with a dashed vertical line.

out on a cell-centered, Cartesian grid in phase space. We label this grid  $\Omega_0$ . The computational domain stretches from 0 to 1 in each physical dimension and  $-V$  to  $V$  in each velocity dimension. The domain bounds in velocity space are chosen so that  $f(\mathbf{x}, \mathbf{v})$  is negligibly small outside of the problem domain. The number of points in  $\Omega_0$  in the position and velocity dimensions are  $N_x$  and  $N_v$ , respectively. The locations of the cell centers are thus  $\mathbf{x}_i = (\mathbf{i} + 1/2)\mathbf{h}_x$  and  $\mathbf{v}_j = (\mathbf{j} + 1/2)\mathbf{h}_v - V$ , where  $(\mathbf{i}, \mathbf{j}) \in (\mathbb{Z}^D, \mathbb{Z}^D)$ , and  $(h_x, h_v) = (1/N_x, 2V/N_v)$  are the cell spacings in position and velocity space. We place one particle in each cell  $(\mathbf{i}, \mathbf{j})$ . For a given initial distribution function,  $f(\mathbf{x}, \mathbf{v}, t_{\text{ini}})$ , the initial discretization is completed by assigning each particle  $p \in \mathbb{P}$  a mass  $m_p$ :

$$m_p = f(\mathbf{x}_p^i, \mathbf{v}_p^i, t_{\text{ini}}) h_x^D h_v^D. \quad (31)$$

where  $\mathbf{x}_p^i$  and  $\mathbf{v}_p^i$  are the initial position and velocity of particle  $p$ . For computational efficiency, we discard particles with masses less than  $10^{-12}$ ; experimentation reveals that tightening this value does not meaningfully af-

fect our convergence results. We illustrate the difference between our “regularized” and “singular” initial particle layouts in Figure 8. We also show, in Figure 9, a sample spectrum of the particle masses obtained through this procedure. Specifically, these were the actual particle masses used in the  $\sigma_i = 1$ ,  $N_{\text{cells}} = 256$  calculation described in Table 3. Note that there is long tail of low-mass particles that contribute little to the overall distribution function - this is a consequence of our selecting a relatively low particle mass threshold and a relatively large domain in velocity space. It is likely that, by relaxing these parameters, significantly fewer particles can be used without degrading the solution.

As an aside, although our regularized initial conditions are “warm” in a sense, they should not be confused with “Warm Dark Matter”. Our initial velocity dispersion is artificially large and introduced for numerical purposes only. It has nothing to do with the physical velocity dispersion of a hypothetical DM candidate. Simulations



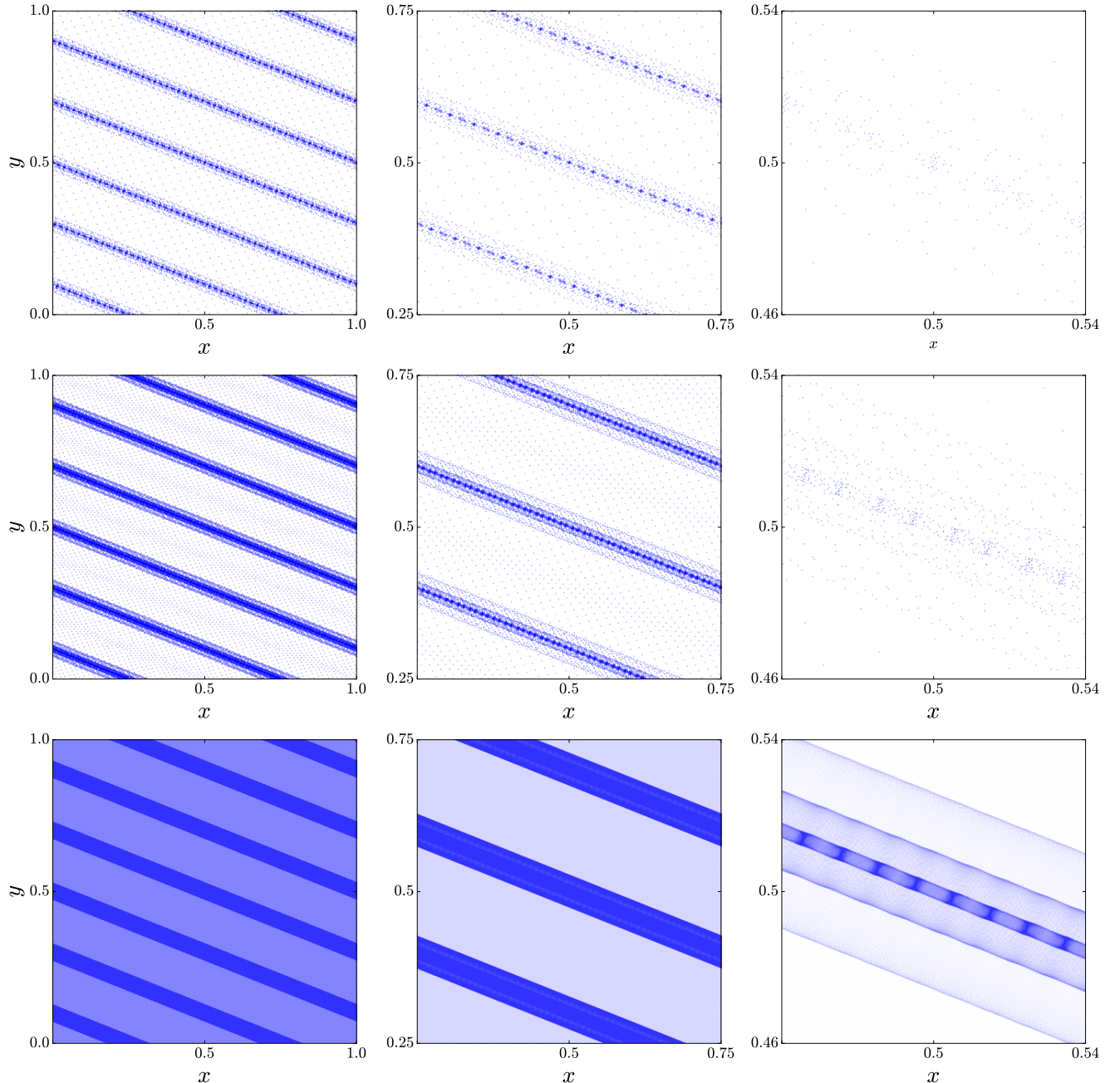


FIG. 5.— Particle  $x$  and  $y$  positions at  $a(t) = 1.0$  on the singular version of the oblique pancake problem. All runs use a  $256^2$  Poisson mesh. The top row shows a run with a sparse particle count of 1 particle every 4 Poisson cells. The middle row uses 1 particle per cell, and the bottom row uses a very high particle count of 256 particles per cell. For all three runs, the leftmost panels show the entire problem domain, and the view zooms in as you move to the right. All runs show signs of small-scale fragmentation that is inconsistent with the symmetry of the problem.

of structure formation in Warm DM universes in fact use “singular” initial conditions by our definition, but they suppress power in fluctuations below some free-streaming scale associated with the rest mass of the assumed dark matter particle.

#### 4.2. 1D Results

To begin, we fix the regularization parameter at  $\sigma_i = 1$ . We otherwise repeat the calculation in Section 6, using the same values for  $a_{\text{ini}}$ ,  $a_{\text{caustic}}$ ,  $a_{\text{stop}}$ ,  $k$ , and  $C_{\text{exp}}$ . For our lowest resolution run, we use  $N_{\text{cells}} = 256$ ,  $N_x = 512$ , and  $N_v = 512$ . We also perform runs with the resolution

increased to  $N_{\text{cells}} = 512, 1024, 2048$ , and  $4096$ , and the other elements of the discretization refined accordingly. That is, when we refine the Poisson mesh, we also refine  $N_x$ ,  $N_v$ , and decrease the time step parameter  $C_{\text{exp}}$ , as before. These run parameters are summarized in Table 3. The resulting solutions are shown in Figure 11. The effect of the (in this case quite large) artificial velocity dispersion to smooth out the density peaks over some comoving length scale, so that the peak densities in the caustics now have well-defined maximum values. As shown in Figure 10, the density caustics now appear

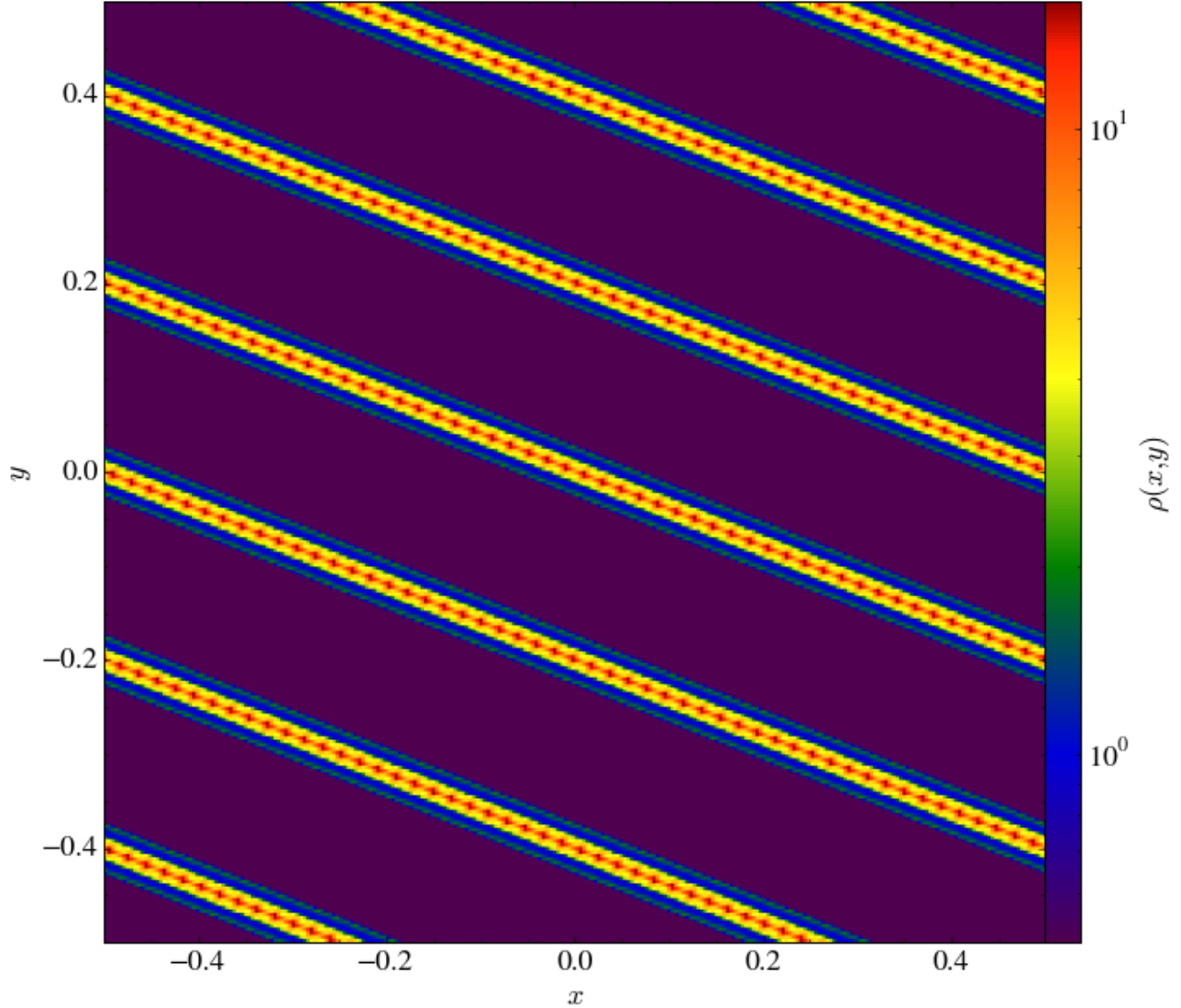


FIG. 6.— The density field at  $a(t) = 1$  from the low resolution, singular version of the oblique pancake problem. The density has been computed by deposition onto a  $256^2$  mesh.

TABLE 3  
SUMMARY OF PARAMETERS FOR THE REGULARIZED, 1D  
PANCAKE RUNS

$\sigma_i$ <sup>a</sup>	$N_{\text{cells}}$	$N_x$ <sup>b</sup>	$N_v$ <sup>c</sup>	$C_{\text{exp}}$
$\{1, 1/2, \dots, 1/32\}$	64	128	128	$4.0 \times 10^{-2}$
$\{1, 1/2, \dots, 1/32\}$	128	256	256	$2.0 \times 10^{-2}$
$\{1, 1/2, \dots, 1/32\}$	256	512	512	$1.0 \times 10^{-2}$
$\{1, 1/2, \dots, 1/32\}$	512	1024	1024	$5.0 \times 10^{-3}$
$\{1, 1/2, \dots, 1/32\}$	1024	2048	2048	$2.5 \times 10^{-3}$
$\{1, 1/2, \dots, 1/32\}$	2048	4096	4096	$1.25 \times 10^{-2}$

<sup>a</sup> The notation  $\{1, 1/2, \dots, 1/32\}$  means we have varied the parameter  $\sigma_i$  over the stated range.

<sup>b</sup> The number of cells in the initial particle mesh in the  $x$  direction.

<sup>c</sup> The number of cells in the initial particle mesh in the  $v$  direction.

to be converging with increased mesh resolution.

We also examine the convergence rates for the density, gravitational field, and potential on the regularized version of the problem. The rates are computed as in Section 3.2 and shown in Figure 12. Overall, the convergence behavior is much better than on the singular version of the problem. The density order is quite noisy at early times, but the rates for the potential and gravitational field are close to 2 for all the times we consider. The convergence rate for the density is slower than 2 at late times, but the density is not divergent as it was on the singular problem.

#### 4.3. The Double Limit

In Section 4.2, we showed that for a given value of  $\sigma_i$ , our PIC method converges at late times on the 1D, regularized pancake problem. Of course, in dark matter simulations, we are really interested in the case where  $\sigma_i = 0$ . Fortunately, we can exploit the fact that we are able to obtain converged solutions to the regular-

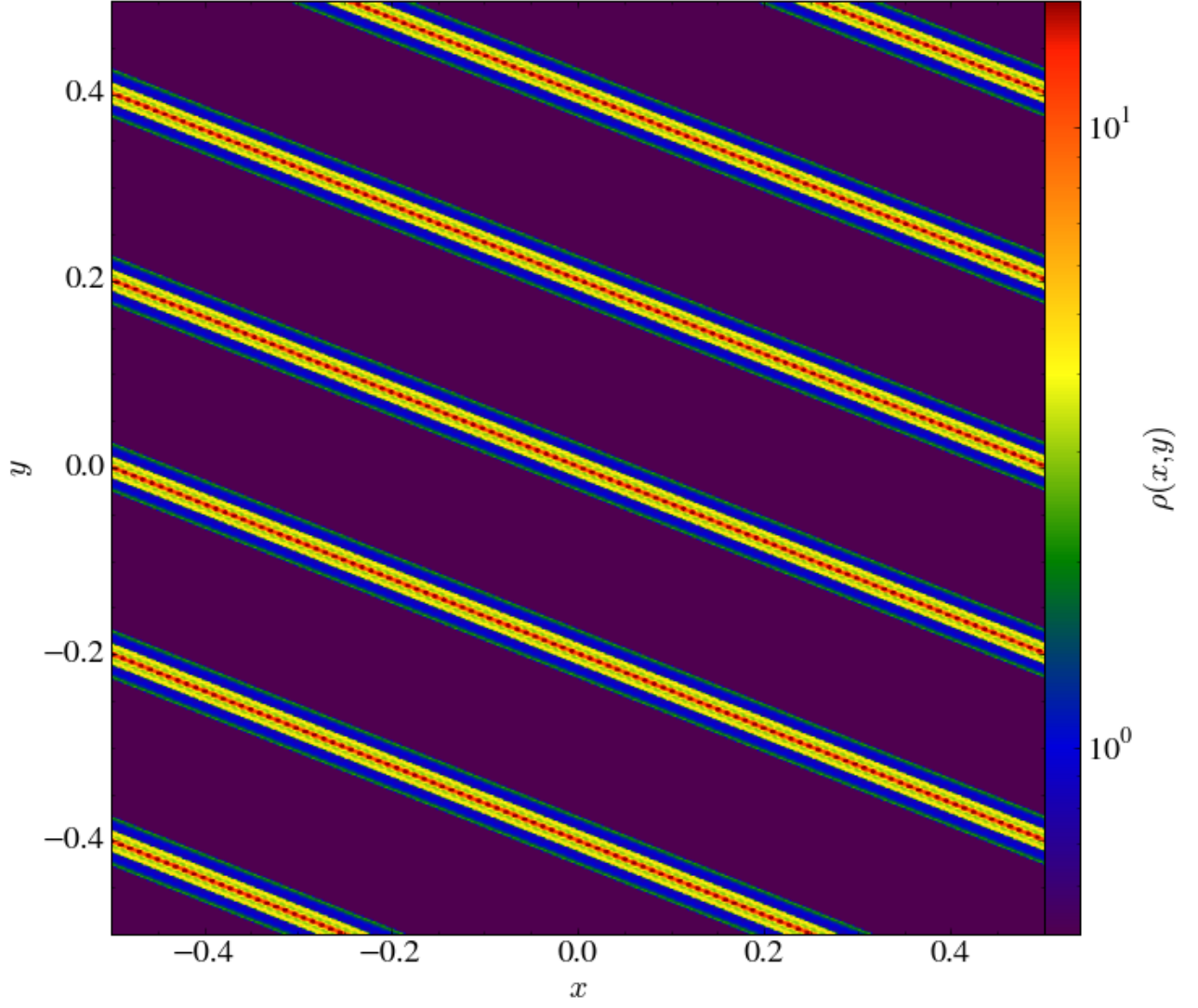


FIG. 7.— The density field at  $a(t) = 1$  from the high resolution, singular version of the oblique pancake problem, using the same color scale as Figure 6. The density has been computed by deposition onto a  $512^2$  mesh.

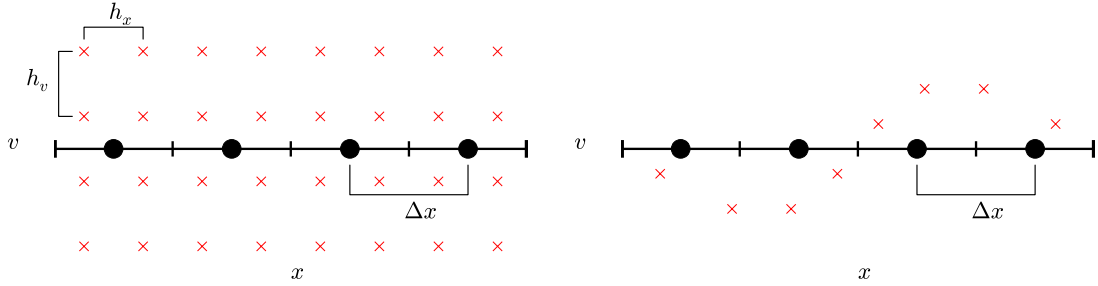


FIG. 8.— A schematic demonstration of the difference between regularized and singular initial conditions. Left panel - the initial phase-space discretization for our regularized runs, for  $\Delta x/h_x = 2$ . The red crosses show the initial particle locations, and the black dots mark the centers of the Poisson mesh. The particle masses are computed by sampling the initial distribution function. Right - the initial particle positions for our singular runs, also for  $\Delta x/h_x = 2$ . The particles all have equal masses, and have been displaced from their initial positions using the Zel'dovich approximation.

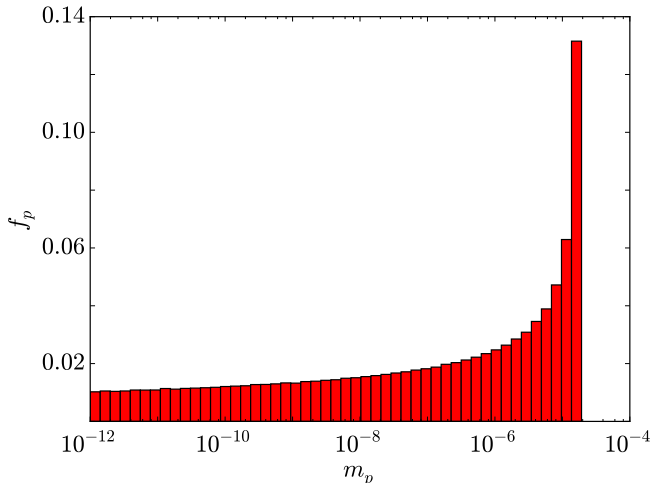


FIG. 9.— A histogram of the particle masses obtained through the sampling procedure described in Section 4.1. We have used 50 logarithmically-spaced mass bins, and show  $f_p$ , the fraction of the total number of particles in each bin. To generate this plot, we have used  $\sigma_i = 1.0$ ,  $V = 6.0$ , and taken  $N_x = N_v = 512$ . All other parameters are the same as in Section 3.2.

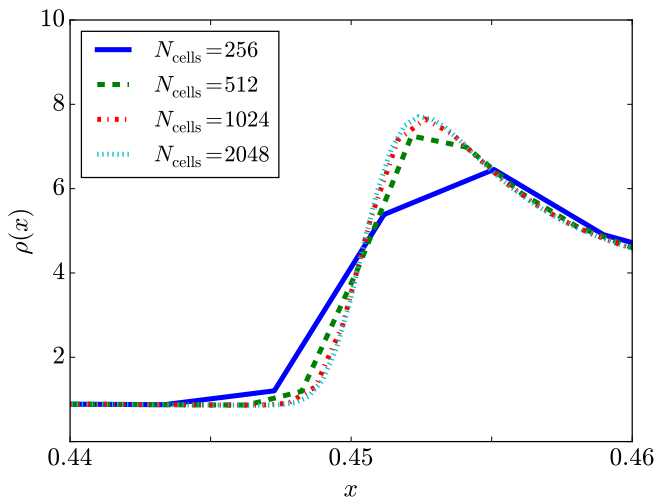


FIG. 10.— A zoomed-in view of one of the outer density caustics from the 1D, regularized problem with  $\sigma_i = 1$ , taken at  $a(t) = 1$ . The different colors correspond to different resolutions: solid blue line -  $N_{\text{cells}} = 256$ ; dashed green line -  $N_{\text{cells}} = 512$ ; dashed-dotted red line -  $N_{\text{cells}} = 1024$ ; dotted cyan line -  $N_{\text{cells}} = 2048$ .

ized problems to also obtain a solution to the singular problem by considering the limit as  $\sigma_i \rightarrow 0$ . **This procedure gives us a way of separating numerical errors, which should improve with mesh refinement if our method converges, from regularization error, i.e. the fact that our caustics are artificially smoothed out by the artificial initial velocity dispersion.** In this section, we illustrate this process on our 1D problem setup. We vary the quantity  $\sigma_i$  over the range  $\{1, 1/2, 1/4, 1/8, 1/16, 1/32\}$ . For each value of  $\sigma_i$ , we increase the mesh resolution until we obtain a converged solution (see Table 3 for a full list of the run parameters). Next, we consider the limit of the converged solutions to the regularized problems as  $\sigma_i \rightarrow 0$ . This concept of a “double limit” - letting both the mesh spacing and  $\sigma_i$  go to zero - was inspired by Krasny (1986),

who used a similar procedure to remove singularities in vortex methods for incompressible fluid flows. The results are shown in Figure 13. These figures show that the regularized solutions approach the singular solution as  $\sigma_i \rightarrow 0$ . In particular, for  $\sigma_i < 1/4$ , the outer caustic positions match precisely. The main difference between the singular and regularized runs is that the peak caustic densities are limited in the regularized runs, and the structures in the inner caustics are smoothed out over some comoving length scale. Outside of the caustics, the effect of  $\sigma_i$  on the solutions is small. This finding suggests an alternative approach to perfectly cold initial conditions in DM simulations. **Instead of using singular initial conditions, which make code validation difficult, one instead decides on a length scale below which one will not believe the answer, and then chooses the corresponding initial artificial velocity dispersion. Given that velocity dispersion, one can rigorously demonstrate convergence at the desired order of accuracy.**

To make the concept of the double limit more quantitative, we define the following error metric, analogous to the Richardson extrapolated error introduced in Section 3.2:

$$e^h(\sigma) = \|g_i^h(\sigma) - g_i^h(\sigma/2)\|. \quad (32)$$

That is, for each value of the mesh spacing  $h$ , we compute the norm of the difference between the solution with the regularization parameter  $\sigma_i = \sigma$  and the one with  $\sigma_i = \sigma/2$ . We use the  $L_2$  norm to compute the error metric, and focus on the solution for the gravitational field. The results for  $e^h(\sigma)$  for all the runs in Table 3 are shown in Figure 14, for the same expansion factors shown in Figure 2. There are two conclusions to be drawn from this plot. First, it shows that the quantity  $e^h(\sigma)$  approaches a well-defined value for each pair of velocity dispersions. By  $N_{\text{cells}} = 2048$ ,  $e^h(\sigma)$  has leveled off for all of the values of  $\sigma_i$  considered. Second, it confirms that the regularized solutions themselves converge as  $\sigma_i \rightarrow 0$ .

There are several conclusions that can be drawn from this exercise. First, it shows that our method of regularizing the initial conditions through an artificial Gaussian velocity dispersion does not affect the dark matter structure outside the caustic positions, provided  $\sigma_i$  is chosen small enough. Second, it gives us reason to believe that the solution to the singular problem obtained via the standard PIC method is in fact the correct answer to the 1D pancake problem after  $a_{\text{caustic}}$ . Because the density field diverges in the perfectly cold limit, the standard method of validating the solution through examining the convergence rate is not available. By considering the limit of a series of regularized problems, we can have confidence that the solution obtained to the singular problem is correct. Third, since we are not able to solve the oblique version of this problem directly using cold initial conditions, regularization may provide a way to obtain approximate solutions to the singular problem in more than one spatial dimension. We test this hypothesis in the next section.

#### 4.4. 2D Results

We now compute the solution to a regularized version of the 2-dimensional problem in Section 3.3. We do two runs in this section, which are summarized in Table 4.

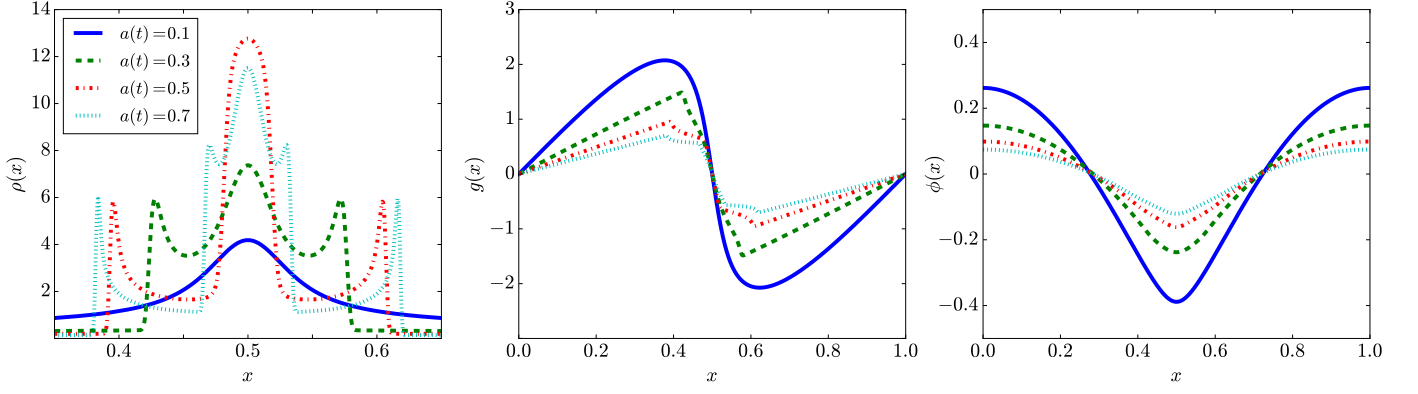


FIG. 11.— The time evolution of the solution to the  $\sigma_i = 1$  version of the 1D, regularized problem. The left panel shows the density, the middle panel the gravitational field, and the right panel the potential. The plotted curves are from the run with  $N_{\text{cells}} = 2048$ ; the  $N_{\text{cells}} = 1024$  run would be indistinguishable from the plotted solutions at the scales shown. The different lines correspond to the results at different expansion factors: solid blue line -  $a(t) = 0.1$ ; dashed green line -  $a(t) = 0.3$ ; dashed-dotted red line -  $a(t) = 0.5$ ; dotted cyan line -  $a(t) = 0.7$ .

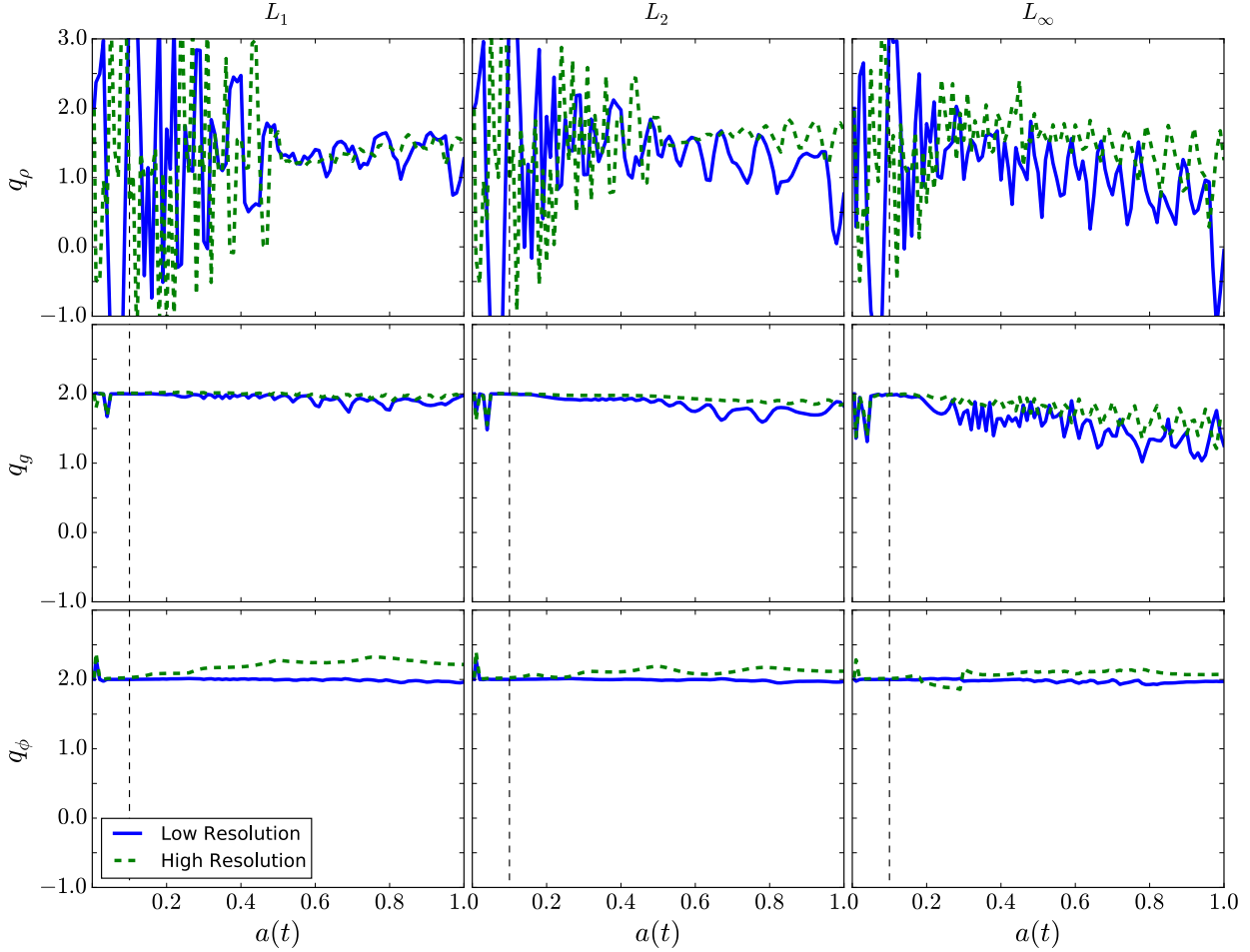


FIG. 12.— The same as Figure 4, but for the  $\sigma_i = 1$  version of the regularized problem. The blue line compares the result for the  $N_{\text{cells}} = 256, 512$ , and  $1024$ , while the green line shows the same quantity for resolutions  $N_{\text{cells}} = 512, 1024$ , and  $2048$ .



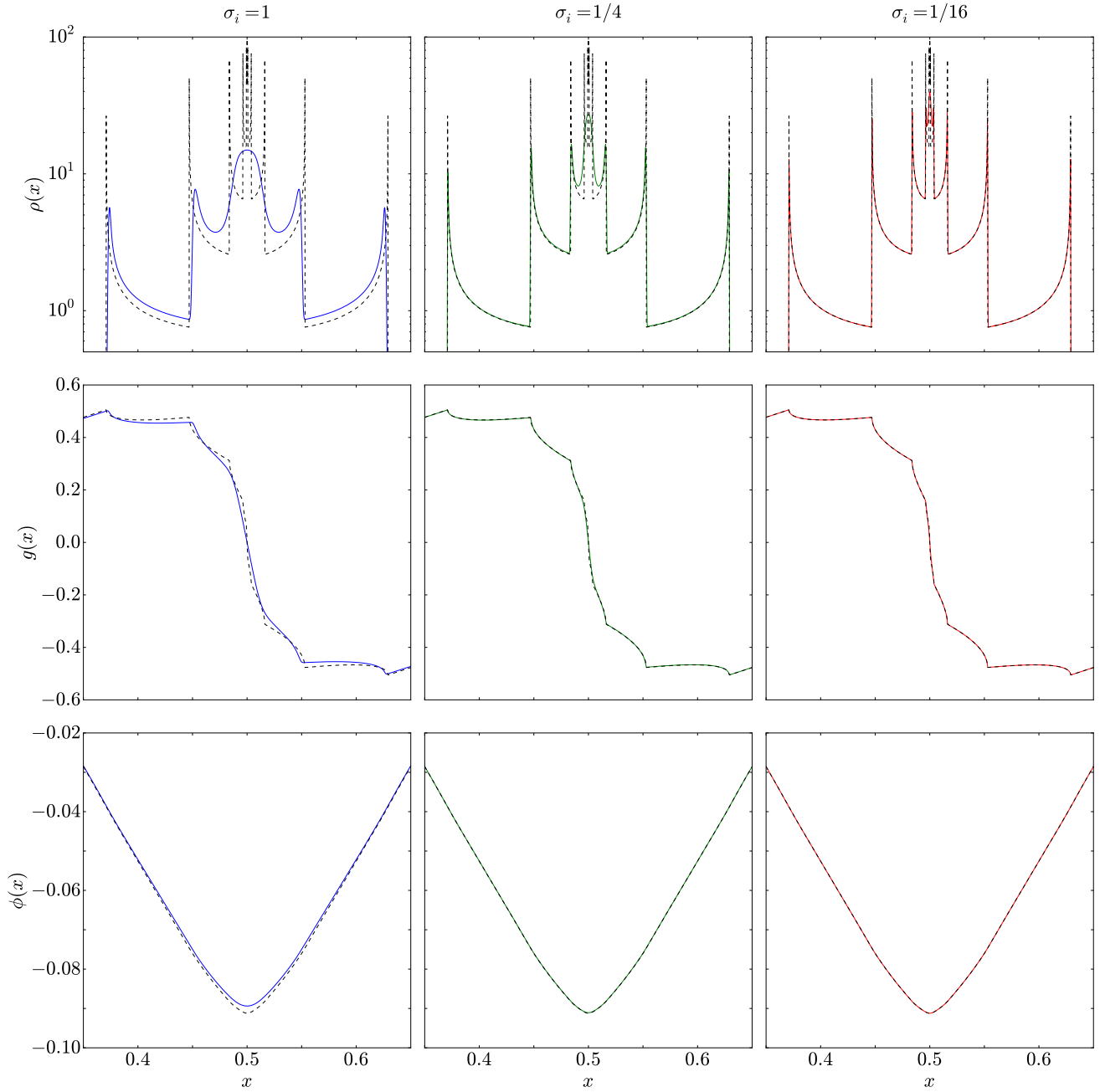


FIG. 13.— The effect of letting  $\sigma_i \rightarrow 0$  on the obtained solutions. The solid lines show the converged density (top row), gravitational field (middle row), and potential (bottom row) from our regularized runs at  $a(t) = 1$ . The black dotted lines are taken from a very high ( $N_{\text{cells}} = 16,384$ ) resolution singular calculation. The regularized results are taken from the runs with  $N_{\text{cells}} = 4096$  - the highest resolution available. The solutions are sufficiently converged that the  $N_{\text{cells}} = 2048$  solutions would be indistinguishable from the plotted curves on this plot. The left column (blue) shows  $\sigma_i = 1$ , the middle column (green) shows  $\sigma_i = 1/4$ , and the right column (red) shows  $\sigma = 1/16$ .

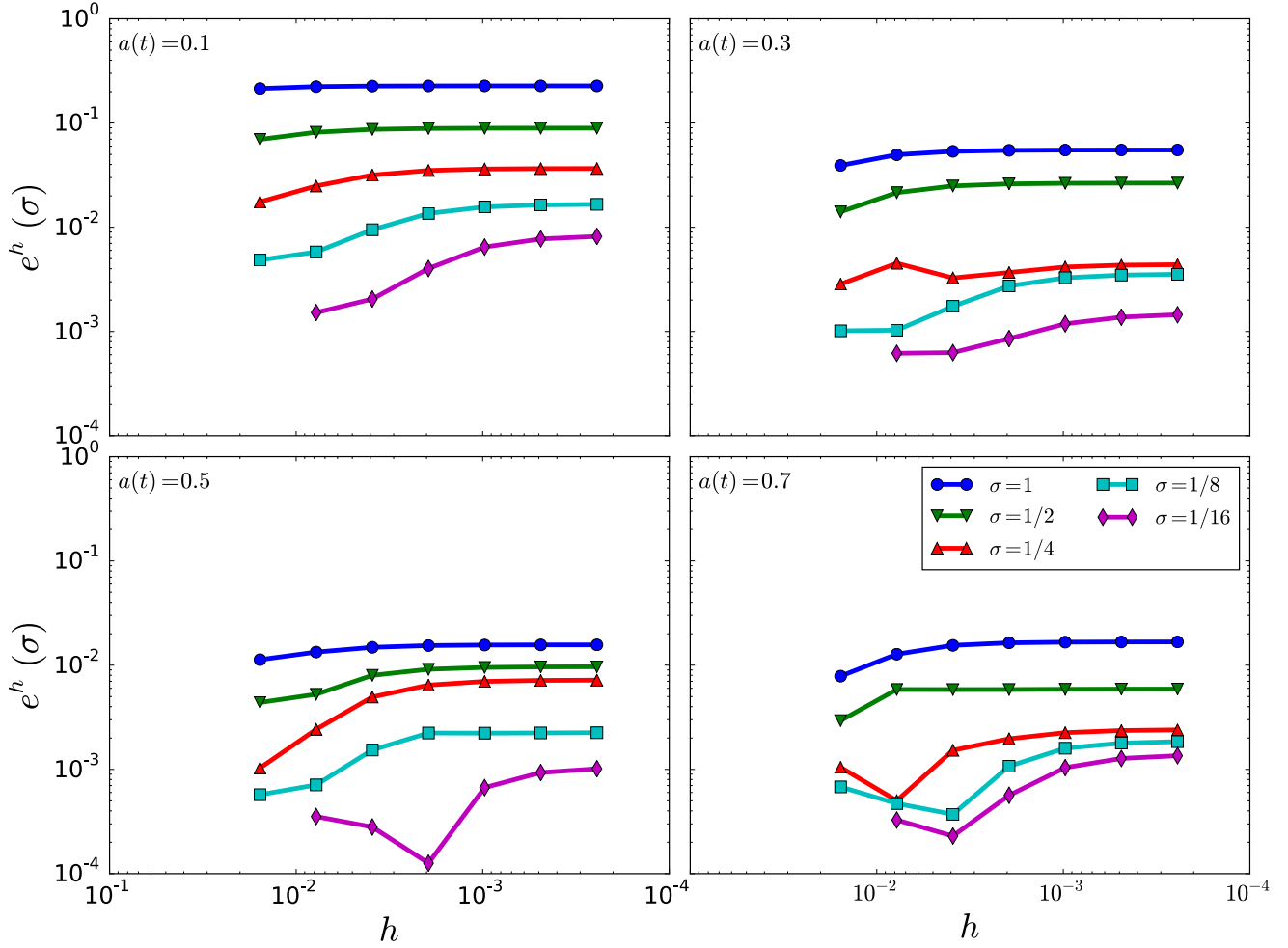


FIG. 14.— Our error metric  $e^h(\sigma)$  computed for all runs in Table 3. The blue curve shows  $e^h(\sigma = 1)$  - that is, the  $L_2$  norm of the change in the solution for the gravitational field when  $\sigma$  changes from 1.0 to 0.5, as a function of  $h$ . The green curve corresponds to  $\sigma = 0.5$ , the red curve  $\sigma = 0.25$ , cyan is  $\sigma = 0.125$ , and magenta is 0.0625. The values in this plot were computed at  $a(t) = 0.1$ , (top left),  $a(t) = 0.3$ , (top right),  $a(t) = 0.5$ , (bottom left), and  $a(t) = 0.7$ , (bottom right).

TABLE 4  
SUMMARY OF PARAMETERS FOR THE  
REGULARIZED, 2D PANCAKE RUNS

$\sigma_i$	$N_{\text{cells}}$	$N_x^a$	$N_v^b$	$C_{\text{exp}}$
1/16	$128^2$	256	128	$2.0 \times 10^{-2}$
1/16	$256^2$	512	128	$1.0 \times 10^{-2}$

Both runs have set  $\sigma_i = 1/16$ , and construct  $\Omega_0$  such that  $N_x = 256$ , and  $N_v = 128$ . In this case, the total number of cells in the phase space domain on which the particles are initialized is  $N_x^2 \times N_v^2$ . The number of cells in the Poisson mesh is fixed at  $N_{\text{cells}} = 128^2$ . All of the other parameters are the same as in Section 3.3. The resulting particle positions for the higher resolution are shown in Figure 15. We also display deposited fields, as in Section 3.3, for both the low (Figure 16) and high (Figure 17) resolutions runs.

Unfortunately, despite the improved convergence rates in 1D, the regularized problem suffers from the same clumping problem as the singular problem for the oblique

pancake setup. However, there is an additional refinement we can introduce to the basic PIC procedure. In the next section, we introduce particle remapping, and show that, in concert with the regularization of the initial conditions described in this section, it can improve the performance of PIC on the oblique problem.

## 5. REMAPPING

In the error analysis of electrostatic PIC in WMC2011, the stability error for the electric field contains a term that grows exponentially with time. Given long enough integration times, this error term can become large enough to significantly degrade the quality of the numerical solution. A common strategy for controlling this errors is to periodically restart the problem with a new set of particles, before the accumulated errors in the particle trajectories become too large. Such “regridding” or “remapping” techniques have been applied successfully to particle schemes for fluid dynamics, such as vortex methods and smoothed particle hydrodynamics (SPH) (Vadlamani et al. 2004; Koumoutsakos 1997; Cottet & Koumoutsakos 2000; Chaniotis et al. 2002), and to PIC

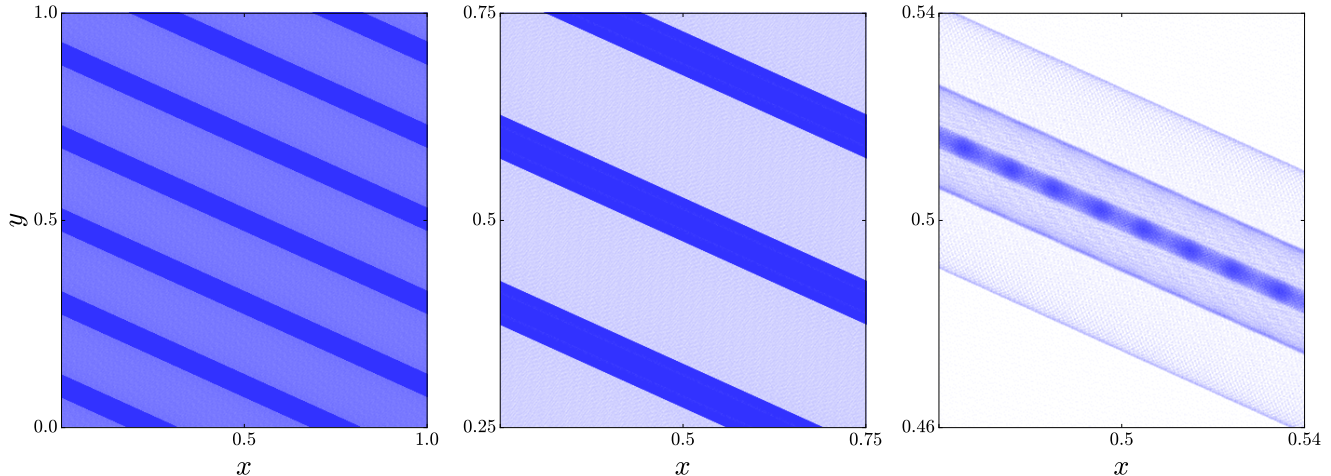


FIG. 15.— Particle  $x$  and  $y$  positions at  $a(t) = 1$  for the  $\sigma_i = 1/16$  version of the regularized oblique pancake problem. The scales shown are the same as in Figure 5

in the context of plasma physics (Denavit 1972; Chen et al. 2008; Wang et al. 2011, 2012).

In the case of PIC, the basic idea is to interpolate the distribution function back onto a regular mesh in phase space, initialize a new set of particles, and restart the calculation. In principle, the remapping mesh can uniform. However, repeatedly re-sampling the distribution function on a uniform mesh leads to a large number of relatively weak particles in the tails of the distribution function - a poor use of the available resolution elements. Furthermore, in the cosmological context, remapping serves an additional role. We have regularized the initial conditions by introducing a finite initial velocity dispersion,  $\sigma_i$ . However, as the scale factor  $a(t)$  increases,  $\sigma(t)$  will scale as  $\sigma_i/a(t)$ . Thus, if we employ the above procedure on a uniform mesh, the artificial velocity spread in the distribution function will be resolved more and more poorly as time advances, undoing the natural adaptivity of Lagrangian schemes.

To remedy these issues, we remap on an adaptive hierarchy of meshes that automatically adjusts the velocity-space resolution on the finest level so that the distribution is resolved by approximately the same number of particles throughout the simulation. The AMR hierarchy is defined as a set of cell-centered, Cartesian meshes in phase space. We label the meshes  $\Omega_\ell$ , where  $0 \leq \ell \leq \ell_{\max}$  are the level numbers. The coarsest mesh,  $\Omega_0$ , covers the entire phase-space domain and is the same as the mesh used for generating the initial particles. The finer meshes are unions cell-centered rectangles and in general cover only a portion of the problem domain. The mesh spacing of level  $\ell$  is related to the mesh spacing on the next coarsest level by  $h_\ell = h_{\ell-1}/r_{\text{ref}}$ , where  $r_{\text{ref}}$  is the AMR refinement ratio. The valid region of an AMR level is defined as the region not covered by finer levels, and the composite grid is defined as the union of the valid regions of each level.

We describe the remapping algorithm in detail below, first using a single, uniform mesh to represent  $f(\mathbf{x}, \mathbf{v})$ , second using a fixed hierarchy of levels, and finally using the full, adaptive hierarchy. Note that, as presently constructed, the remapping procedure requires the regularization of the initial conditions - there must be a well-defined distribution function to generate the parti-

cles from.

### 5.1. Remapping on a Uniform Mesh

As an illustration, consider remapping a set of particles on  $\Omega_0$  only. With only one level, our remapping algorithm is identical to WMC2011. The remapping step proceeds as follows:

**Deposition.** From the particle data, we represent  $f(\mathbf{x}, \mathbf{v})$  on  $\Omega_0$  by deposition. The position and velocity coordinates of cell centers on  $\Omega_0$  are the same as those used for initialization,  $\mathbf{x}_i = (\mathbf{i} + 1/2)h_x$  and  $\mathbf{v}_j = (\mathbf{j} + 1/2)h_v - V$ , with  $(\mathbf{i}, \mathbf{j}) \in (\mathbb{Z}^D, \mathbb{Z}^D)$ . The values of  $f$  on the mesh points are thus:

$$f_{ij} = \sum_p \left( \frac{m_p}{\Gamma} \right) \mathbf{W}_4 \left( \frac{\mathbf{x}_i - \mathbf{x}_p}{h_x} \right) \mathbf{W}_4 \left( \frac{\mathbf{v}_j - \mathbf{v}_p}{h_v} \right), \quad (33)$$

where  $\Gamma = h_x^D h_v^D$  is the phase-space cell volume and  $\mathbf{W}_4(\mathbf{x})$  is the  $2D$ -dimensional version of the third-order accurate interpolating function from Monaghan (1985):

$$\mathbf{W}_4(\mathbf{x}) = \prod_{d=1}^{2D} W_4(x_d), \quad (34)$$

$$W_4(x) = \begin{cases} 1 - \frac{5}{2}|x|^2 + \frac{3}{2}|x|^3, & 0 \leq |x| \leq 1, \\ \frac{1}{2}(2 - |x|^2)(1 - |x|), & 1 \leq |x| \leq 2, \\ 0 & \text{otherwise.} \end{cases} \quad (35)$$

High-order interpolation is necessary because one order of accuracy is lost during the remap step (WMC2011). Thus, for the overall scheme to have second-order accuracy, a third-order or higher interpolating function must be used.

**Positivity Preservation.** A consequence of high-order interpolation is that  $f$  is not guaranteed to be positive for all cells in  $\Omega_0$ . To account for this, a correction,  $\Delta f$ , must be applied to the distribution function after it is deposited onto the mesh. We accomplish this using a mass transfer procedure that redistributes matter to cells with  $f < 0$  from neighboring cells in proportion to each neighbor's current value of  $f$ . We write the undershoot in cell  $(\mathbf{i}, \mathbf{j})$  as  $\delta f_{ij}$ :

$$\delta f_{ij} = \min(0, f_{ij}), \quad (36)$$



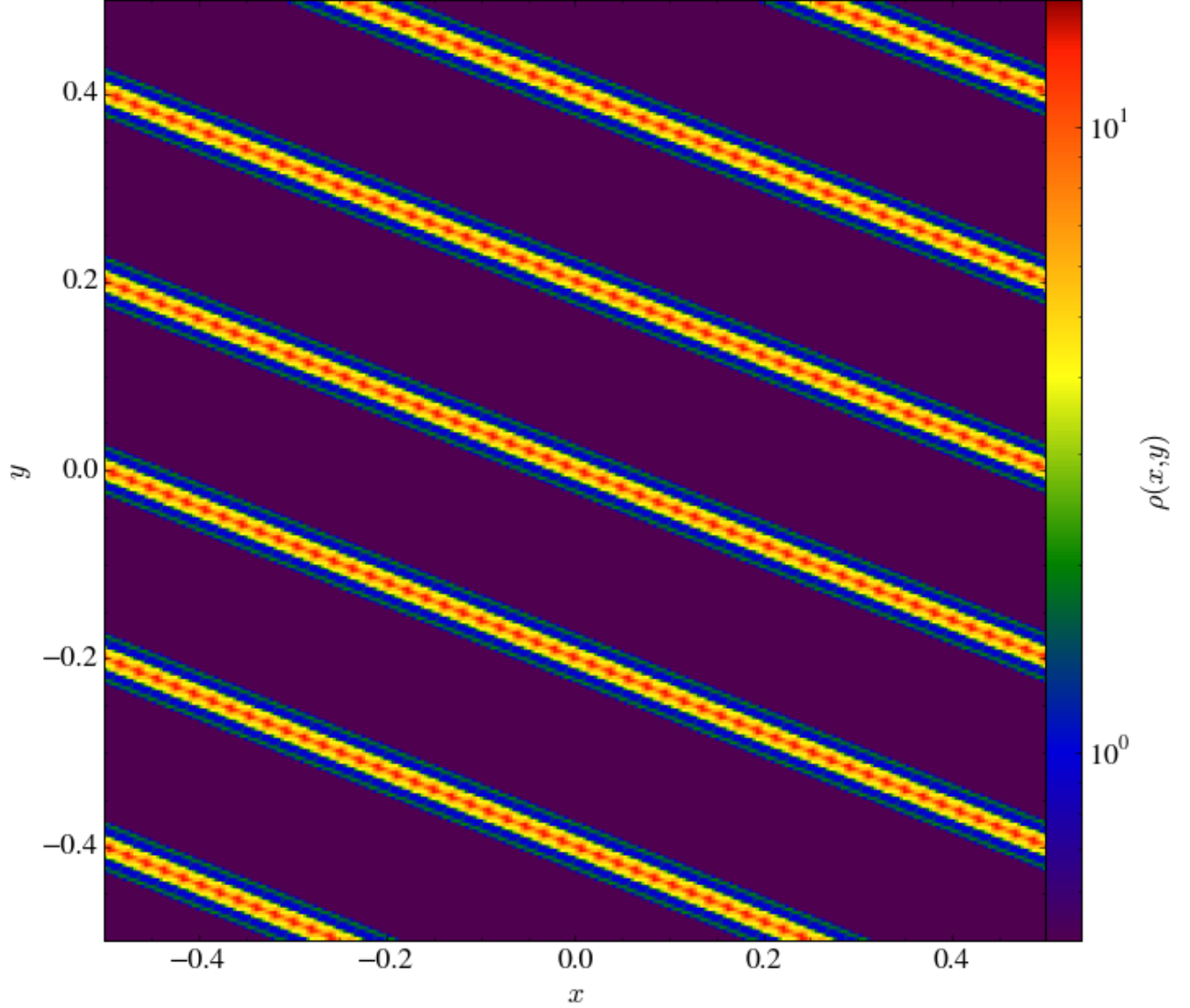


FIG. 16.— The solution for the density at  $a = 1$  from the low-resolution calculation of the oblique pancake problem using regularized initial conditions with  $\sigma_i = 1/16$ . The color scale is the same as in Figure 6. Particle remapping was not turned on for this run. For details of the parameter choices we adopted, see text

and initialize  $\Delta f = 0$ . For every cell with negative  $\delta f$ , we compute the capacity of each neighboring cell  $\mathcal{C}_{i+m,j+n}$  as

$$\mathcal{C}_{i+m,j+n} = \max(0, f_{i+m,j+n}), \quad (37)$$

where the notation  $(i + m, j + n)$  refers to any neighboring cell in the redistribution zone. We perform this redistribution over a window of  $\mathcal{N}$  cells in each direction. The value of  $\mathcal{N}$  must be chosen so that it matches the number of cells involved in the particle interpolation stencil: in this case,  $\mathcal{N} = 4$ . The total capacity is

$$\mathcal{C}_{\text{tot}} = \sum_{\substack{\text{neighbors} \\ m' \neq 0, n' \neq 0}} \mathcal{C}_{i+m',j+n'}. \quad (38)$$

The correction to  $f$  within the redistribution window is incremented as

$$\Delta f_{i+m,j+n} += \frac{\mathcal{C}_{i+m,j+n}}{\mathcal{C}_{\text{tot}}} \delta f_{ij}. \quad (39)$$

This procedure is repeated for each non-positive cell in  $\Omega_0$ , and the correction is then applied to  $f$ . Note that there is no guarantee that  $f$  will be positive after just one pass of this algorithm. In general, we must iterate until  $f$  is positive everywhere. The full positivity preservation algorithm looks like:

```

while  $\max(f) > 0$  do
  Set  $\Delta f = 0$ 
  for all  $(i, j) \in \Omega_0$  do

```

```

    if  $f_{ij} < 0$  then

```

```

        for all neighbors  $(i + m, j + n)$  do

```

$$\Delta f_{i+m,j+n} += \frac{\mathcal{C}_{i+m,j+n}}{\mathcal{C}_{\text{tot}}} \delta f_{ij}. \quad (40)$$

```

        end for
    end if
  end for

```

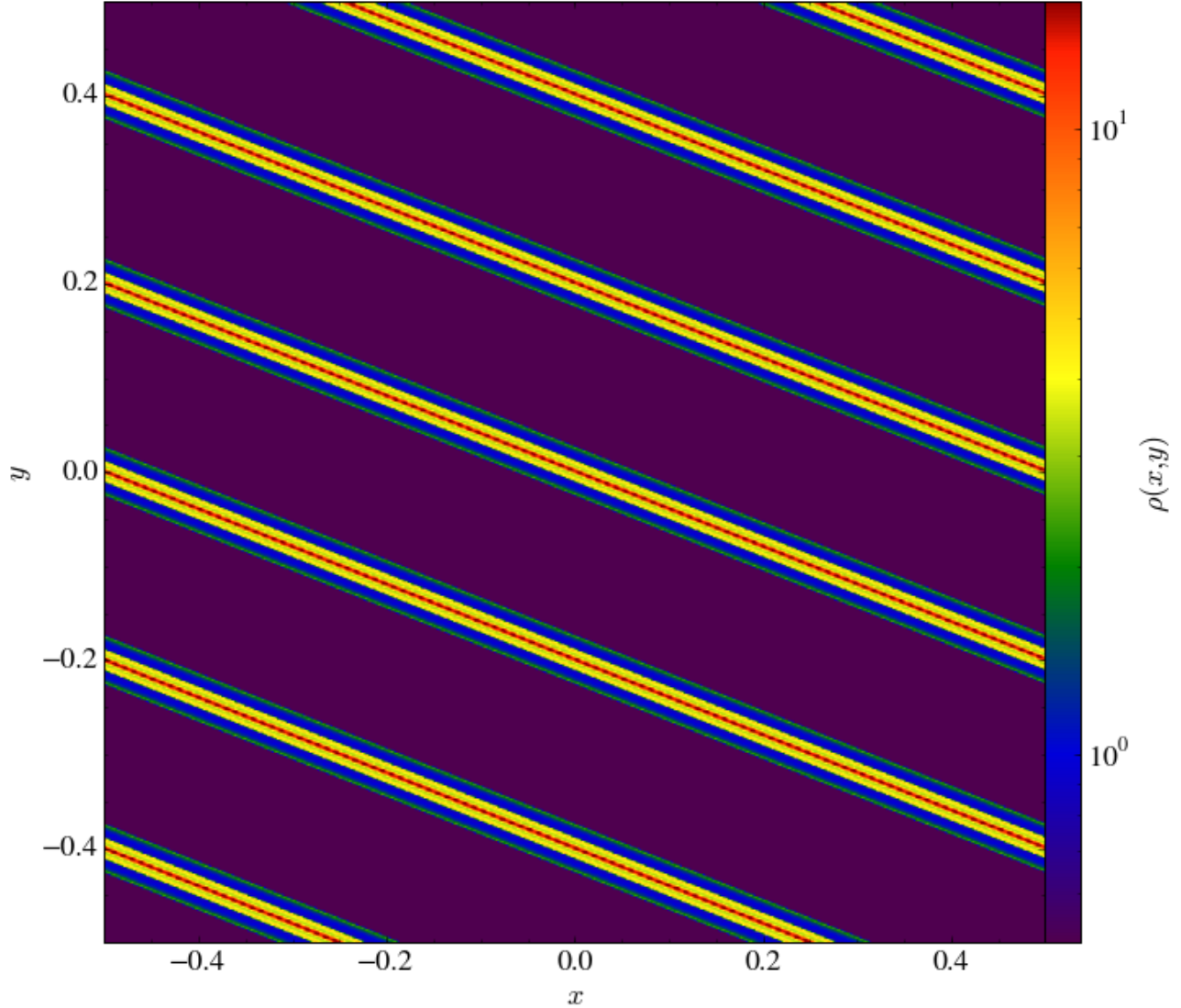


FIG. 17.— The solution for the density at  $a = 1$  from the high-resolution calculation of the oblique pancake problem using regularized initial conditions with  $\sigma_i = 1/16$ . The color scale is the same as in Figure 6. Particle remapping was not turned on for this run. For details of the parameter choices we adopted, see text.

**end if**  
**end for**

$$f^{\text{new}} = f^{\text{old}} + \Delta f. \quad (41)$$

**end while**

In practice, we find that 2 or 3 iterations is usually sufficient to ensure that the positivity of the distribution function is preserved.

**Particle generation.** Finally, we generate a new set of particles by sampling the remapped distribution at every cell  $(i, j)$  in  $\Omega_0$  and initializing a particle with mass

$$m_p = f_{ij} h_x^D h_v^D. \quad (42)$$

As in the problem initialization, we discard particles with masses less than  $10^{-12}$ .

## 5.2. Remapping on a Fixed Hierarchy

Next, consider the case with a hierarchy of levels,  $\{\Omega_0, \Omega_1, \dots, \Omega_{\ell_{\max}-1}\}$ . We impose the constraint that these levels be properly nested by a buffer of  $n_{\text{buff}} = 4$  fine-level cells; this choice is determined by the above interpolation stencil. For now, we assume that the levels are fixed in time and known in advance; we consider the full AMR case in the next section. We must now compute  $f$  on the composite grid. To do so, we first partition the particles  $\mathbb{P}$  into sets  $\{\mathbb{P}_0, \mathbb{P}_1, \dots, \mathbb{P}_{\ell_{\max}-1}\}$  according to what level they will be deposited on. We perform this partition by associating each particle with the finest level that contains its entire interpolation stencil; particles that lie too close to a coarse/fine boundary for their entire clouds to be contained on the fine level go on the next coarser level.

Remapping on a hierarchy of meshes raises the additional issue of what mesh spacing to use in the deposition step: the spacing of the mesh that the particles are being interpolated *to*, or the spacing of the mesh the par-

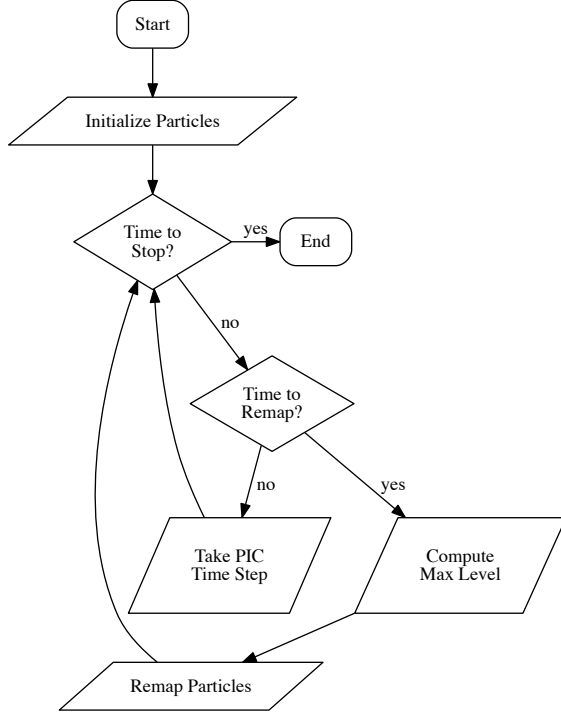


FIG. 18.— The overall control flow of our PIC algorithm, with remapping.

ticles were generated *from*. It is common for AMR PIC schemes to use the spacing of the target mesh for density deposition. However, we find that this technique leads to spurious oscillatory features in the distribution function when used in our remapping algorithm (see Figure 19). Instead, we find it necessary for each particle to remember the spacing of the mesh on which it was generated. We label these spacings  $\epsilon_x^p$  and  $\epsilon_v^p$ .

With that caveat, the remapping step proceeds as follows:

**Deposition.** We deposit the particles in each set onto the appropriate level. For example, on  $\Omega_0$ :

$$f_{ij}^0 = \sum_{p \in \mathbb{P}_0} \left( \frac{m_p}{\Gamma} \right) \mathbf{W}_4 \left( \frac{\mathbf{x}_i - \mathbf{x}_p}{\delta x^0} \right) \mathbf{W}_4 \left( \frac{\mathbf{v}_j - \mathbf{v}_p}{\delta v^0} \right),$$

for  $(i, j) \in \Omega_0$ ,

(43)

where the expression is evaluated for *every* cell  $(i, j)$  in  $\Omega_0$ , not just the valid cells. The quantities  $\delta x^\ell$  and  $\delta v^\ell$  are:

$$\begin{aligned} \delta x^\ell &= \max(h_x r_{\text{ref}}^\ell, \epsilon_x^p) \\ \delta v^\ell &= \max(h_v r_{\text{ref}}^\ell, \epsilon_v^p). \end{aligned}$$
(44)

That is, we use whichever is larger, the mesh spacing of the target mesh or the mesh spacing with which the particles were generated. We find that this interpolation scheme results in a smooth representation of the underlying distribution function even when adaptive refinement is employed.

The corresponding expression for level 1 is:

$$f_{ij}^1 = \sum_{p \in \mathbb{P}_1} \left( \frac{m_p}{\Gamma} \right) \mathbf{W}_4 \left( \frac{\mathbf{x}_i - \mathbf{x}_p}{\delta x^1} \right) \mathbf{W}_4 \left( \frac{\mathbf{v}_j - \mathbf{v}_p}{\delta v^1} \right) + f_{i^c j^c}^0,$$

for  $(i, j) \in \Omega_1$ .

(45)

The second term in this expression accounts for the particles in  $\mathbb{P}_0$  that deposited some of their mass on the region covered by  $\Omega_1$ . The indices  $i^c, j^c$  are *coarsened* versions of  $(i, j)$ :

$$\begin{aligned} i^c &= \left( \left\lfloor \frac{i_0}{r_{\text{ref}}} \right\rfloor, \dots, \left\lfloor \frac{i_{D-1}}{r_{\text{ref}}} \right\rfloor \right), \\ j^c &= \left( \left\lfloor \frac{j_0}{r_{\text{ref}}} \right\rfloor, \dots, \left\lfloor \frac{j_{D-1}}{r_{\text{ref}}} \right\rfloor \right). \end{aligned}$$
(46)

Note that by adopting a 4-cell proper nesting requirement, we avoid the case where this coarse/fine correction can involve more than 2 levels. This process continues up to  $\ell_{\text{max}}$ . In general:

$$f_{ij}^\ell = \sum_{p \in \mathbb{P}_\ell} \left( \frac{m_p}{\Gamma} \right) \mathbf{W}_4 \left( \frac{\mathbf{x}_i - \mathbf{x}_p}{\delta x^\ell} \right) \mathbf{W}_4 \left( \frac{\mathbf{v}_j - \mathbf{v}_p}{\delta v^\ell} \right) + f_{i^c j^c}^{\ell-1},$$

for  $(i, j) \in \Omega_\ell$ .

(47)

The result of applying this procedure over levels  $\{\Omega_0, \dots, \Omega_{\ell_{\text{max}}-1}\}$  is a representation of  $f$  that fully conserves matter over the composite grid.

**Positivity Preservation.** The positivity preservation step is similar to the above. The difference is that when applying positivity preservation algorithm to level  $\ell$ , we must sometimes draw mass from cells that lie on level  $\ell + 1$  and  $\ell - 1$ . For convenience, we accomplish by augmenting each level with a layer of  $\mathbb{N}$  ghost cells. When we apply the correction to  $f^\ell$ , we also apply it to the subsets of levels  $\ell + 1$  and  $\ell - 1$  that are covered by the level  $\ell$  ghosts. To coarsen the correction, we use simple arithmetic averaging, and to refine it, we use piecewise-constant interpolation.

**Particle Generation.** The particle generation step proceeds as in the single-level case, except that we generate one particle for every *valid* cell in the hierarchy. That is, we do:

**for**  $\ell = 0, \dots, \ell_{\text{top}} - 1$  **do**

**for all**  $(i, j) \in \Omega_{\ell, \text{valid}}$  **do**

Initialize particle with mass  $m_p = f_{ij}^\ell h_x^D h_v^D$ .

**end for**

**end for**

### 5.3. Remapping with AMR

In practice, the AMR hierarchy is not known at the beginning of a remapping step. Instead, the AMR level structure must be self-consistently built up from the particle distribution and a suitable set of refinement criteria. In this paper, we use the condition that a cell is tagged for refinement whenever the value of  $f_{ij}^\ell$  exceeds the predetermined threshold  $f_{\text{thresh}} = 0.1$ . Finer levels are then generated in such a way that they cover these tags, and

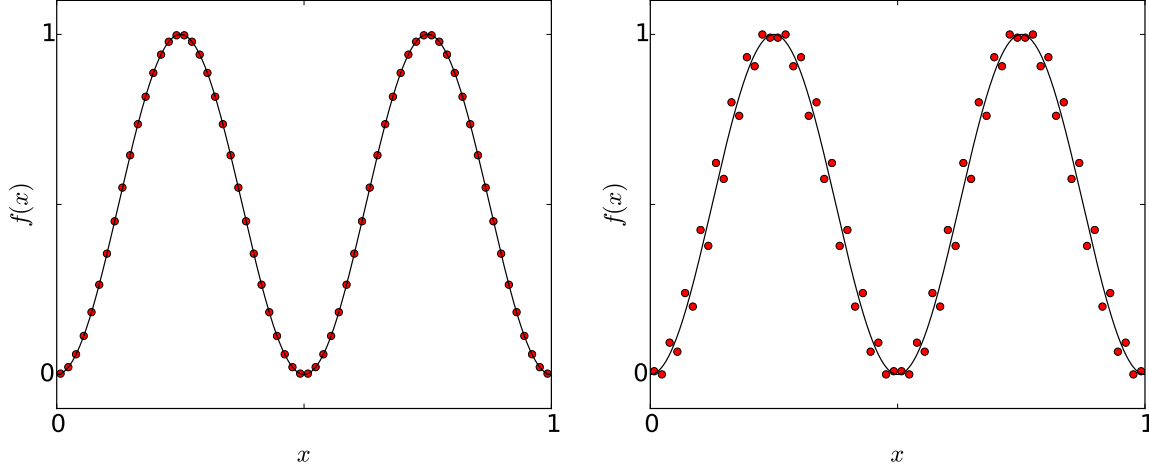


FIG. 19.— A 1D demonstration of our method for depositing particle quantities on refined meshes. In both panels, an initial sinusoidal function  $f(x)$  (the black curve) has been sampled by a set of 32 particles, such that  $h_x = 1/32$ . These particles were then deposited onto a 64 cell mesh ( $\Delta x = 1/64$ ). This mesh was used to generate a new set of particles. In the left panel, the coarser spacing  $h_x$  was used for deposition, while in the right panel, the fine cell spacing  $\Delta x$  was used instead. The coarse spacing results in a much smoother representation of the underlying function.

are consistent with our proper nesting requirement. The other input to the process is the maximum number of allowed levels,  $\ell_{\max}$ . We then build up the AMR representation of  $f$  using the standard bootstrapping procedure:

```

for  $\ell_{\text{top}} = 0, \dots, \ell_{\max} - 1$  do
  Partition particles into  $\{\mathbb{P}_0, \dots, \mathbb{P}_{\ell_{\text{top}}}\}$ 
  Deposit particles on  $\{\Omega_0, \dots, \Omega_{\ell_{\text{top}}}\}$ 
  if  $\ell_{\text{top}} < \ell_{\max} - 1$  then
    Regrid:
    for  $\ell = 0, \dots, \ell_{\text{top}}$  do
      Tag cells on  $\Omega_\ell$  where  $f_\ell > f_{\text{thresh}}$ 
    end for
    for  $\ell = 1, \dots, \ell_{\text{top}} + 1$  do
      Generate new  $\Omega_\ell$ 
    end for
  end if
end for

```

The positivity preservation and particle generation steps are the same as for the fixed hierarchy case. The result of this procedure is an AMR representation of  $f$ . See Figure 20 for an illustration in the 1 + 1D case. Note that, since the particle remap is a purely local process, it is not necessary (nor would it be advisable in 3 + 3D) to store the full phase space distribution at once. However, in the present work we confine our attention to low-dimensional problems, so we store the full  $f$  for simplicity and for illustrative purposes.

#### 5.4. Refinement Criteria

The goal of our use of AMR is to resolve the artificial velocity dispersion  $\sigma_i$  by a roughly constant number of particles during the remap step, even as  $\sigma_i$  decreases with the inverse of the expansion factor. We achieve this by fixing the velocity-space domain boundaries and the cell spacing of the base grid and adding additional levels of refinement to the remapping mesh as the universe expands. We compute the number of AMR levels by requiring that  $\sigma(a)$  is always resolved by at least  $N_\sigma$  cells. If the cell spacing of the base grid in velocity space is  $h_v$ , then we perform the remapping on an AMR hierarchy

TABLE 5  
SUMMARY OF PARAMETERS FOR THE REGULARIZED AND REMAPPED 1D PANCAKE RUNS

$\sigma_i$	$N_{\text{cells}}$	$N_\sigma$	$\Delta a_{\text{remap}}$	$N_x$	$N_v$	$C_{\text{exp}}$
1.0	256	2	0.01	128	128	$1.0 \times 10^{-2}$
1.0	512	4	0.01	256	256	$5.0 \times 10^{-3}$
1.0	1024	8	0.01	512	512	$2.5 \times 10^{-3}$
1.0	2048	16	0.01	1024	1024	$1.25 \times 10^{-3}$

with

$$\ell_{\max} = \left\lceil \frac{\log(N_\sigma h_v / \sigma_i)}{\log(r_{\text{ref}})} \right\rceil \quad (48)$$

levels of refinement, where  $r_{\text{ref}}$  is the AMR refinement ratio.

#### 5.5. Summary

We show the updated control flow of PIC with remapping in Figure 18. After initialization, we begin PIC time integration using the time step in section 2.4. We continue until we reach either the final expansion factor,  $a_{\text{stop}}$ , or the expansion factor associated with the first remap. The remapping can be applied either in fixed interval in  $a$  or every fixed number of time steps. For each remap step, we compute the finest allowed level  $\ell_{\max}$  using Equation 48, and perform the remap on  $\{\Omega_0, \dots, \Omega_{\ell_{\max}}\}$ . This process continues until  $a_{\text{stop}}$  is reached.

### 6. THE REGULARIZED, REMAPPED PANCAKE

#### 6.1. 1D Results

In this section, we examine the performance of the PIC scheme with remapping on the 1D pancake problem with regularized initial conditions (see Section 4.1). Basic PIC converged on this problem (provided the initial conditions were regularized) so we must verify that we can also obtain second order convergence with remapping enabled. For this run, we choose  $\sigma_i = 1$ , and we arrange our remapping meshes so that this velocity dispersion is always resolved by  $N_\sigma = 8$  cells. The initial phase-space grid  $\Omega_0$  is  $N_x \times N_v$ , where  $N_x = 1/h_x = 512$ ,

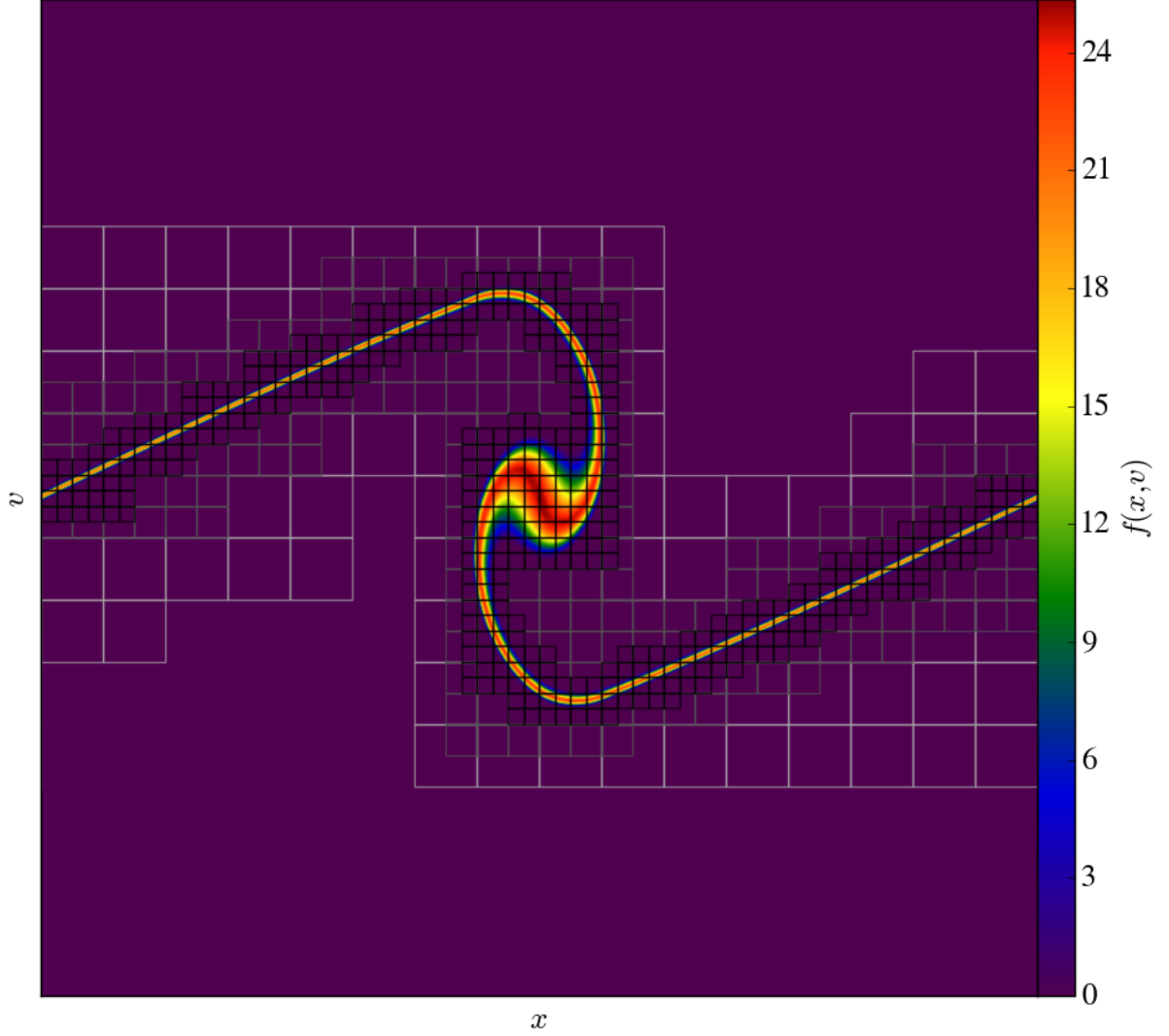


FIG. 20.— An example of a  $1 + 1$ D distribution function represented on an adaptive hierarchy of grids. The level 0 grids have been omitted for clarity, while the level 1, 2, and 3 grids have been over-plotted in white, gray, and black, respectively.

TABLE 6  
FINAL ENERGY ERROR AS A FUNCTION OF RESOLUTION

$N_{\text{cells}}$	Singular <sup>a</sup>	Regularized <sup>a</sup>	Remapped <sup>a</sup>
256	$6.3 \times 10^{-4}$	$3.2 \times 10^{-4}$	$1.2 \times 10^{-3}$
512	$1.5 \times 10^{-4}$	$8.2 \times 10^{-5}$	$1.5 \times 10^{-4}$
1024	$4.2 \times 10^{-5}$	$2.0 \times 10^{-5}$	$7.1 \times 10^{-6}$
2048	$1.2 \times 10^{-5}$	$6.1 \times 10^{-6}$	$3.3 \times 10^{-6}$

<sup>a</sup> The displayed quantity is the energy conservation error  $\epsilon$ , evaluated at  $a(t) = 1$ . For the regularized and remapped columns, we have used the runs with  $\sigma_i = 1$ .

$N_v = 1/h_v = 512$ , and the Poisson mesh is  $N_{\text{cells}} = 256$  cells across. By  $a(t) = 1$ , the mesh spacing in velocity space has been refined by a factor of  $2^5$  on the finest level. We apply the remapping every  $\Delta a_{\text{remap}} = 0.01$  (i.e. at  $a = 0.01, 0.02$ , etc...) and dump out  $f(x, v)$ ,  $\rho(x)$ ,  $g(x)$ , and  $\phi(x)$  after every remap. We show the resulting so-

lution at representative times in Figures 21 through 24. In addition to the density, gravitational field, and potential, we have also displayed the full  $1 + 1$ D phase-space distribution function at the same expansion factors. As expected, most of the phase space cells are empty as thus do not need to be represented by a particle. Visually, the regularized and remapped solutions appear quite similar to the regularization-only solutions, with the effect of  $\sigma_i$  being to smooth out the density at the caustic locations so that rigorous convergence becomes possible.

To investigate the convergence rate, we run an analogous set of problems where we vary all of  $\Delta_x$ ,  $h_x$ ,  $h_v$ , and  $\Delta t$  over the same values as in Section 4.2. Additionally, we also increase  $N_\sigma$  by a factor of two as the resolution increases. The resulting convergence rates are plotted in Figure 25. The convergence rates at late times are much improved compared to the singular runs. Additionally, the convergence rates for the density are improved beyond the regularization-only case. The noisy



convergence rates seen in Section 4.2 at early times for the density are gone. The reason for this behavior is that the remapping procedure constrains the density field to be sampled by several particles per Poisson cell everywhere, which results in a smoother representation of the density. Additionally, the convergence rates for the density are better at late times, as well, with rates close to 2 being obtained in all the norms. The faster-than-second-order convergence rates in the potential are a consequence of the positivity preservation procedure - they are not present if this part of the algorithm is disabled.

It is important to note that, while the above remapping scheme conserves the total  $f$  exactly (modulo our choice to discard particles with masses less than  $10^{-12}$ ), it is not guaranteed to conserve total energy, due to discretization error in performing the remapping on a mesh with finite  $h_x$  and  $h_v$ . However, if enough resolution is used, these errors can be made negligibly small. To demonstrate this, we evaluate the overall energy conservation error in our runs as a function of time both with and without remapping. We define the error in energy conservation as follows. If our code conserved energy exactly, then the particles would obey the Layzer-Irvine equation (e.g., Peebles 1993):

$$\frac{d}{dt} [a(T + U)] = -\dot{a}T, \quad (49)$$

where  $T = \frac{1}{2} \sum_p m_i v_i^2$  is the total kinetic energy of the particles and  $U = \frac{1}{2} \sum_p m_i \phi_i$  is their total gravitational potential energy. We use this equation to define the energy conservation error as  $\epsilon$ :

$$\epsilon = \frac{a(T + U) - a_0(T_0 + U_0) + \int_{a_0}^a T da}{a_0 U_0 - aU}. \quad (50)$$

This quantity would be zero in the case of perfect energy conservation. In Figure 22, we compare  $\epsilon$  as a function of time for the singular pancake run, the  $\sigma_i = 1$  regularized run, and the  $\sigma_i = 1$  regularized and remapped run. We use the highest resolution data available ( $N_{\text{cells}} = 2048$ ). Because of our use of a variable time step, neither the singular nor the regularized runs conserves energy exactly, despite our use of a symplectic integrator. However, the overall error in the energy conservation is small - at the end of all three runs, we have conserved energy to better than 0.002 %. Interestingly, the energy error is actually largest at the end of the singular run, which suffers a spike in the energy error at first caustic formation.

The energy error is practically the same between the regularized and remapped runs - an indication that, at this resolution, the energy error is dominated by something other than discretization error during the remap. For our lower resolution runs, this is not the case. In Table 6, we compare the energy error  $a(t) = 1$  between the singular run, the regularized run with  $\sigma_i = 1$ , and the regularized and remapped run with  $\sigma_i = 1$  for four different resolutions. We see that on

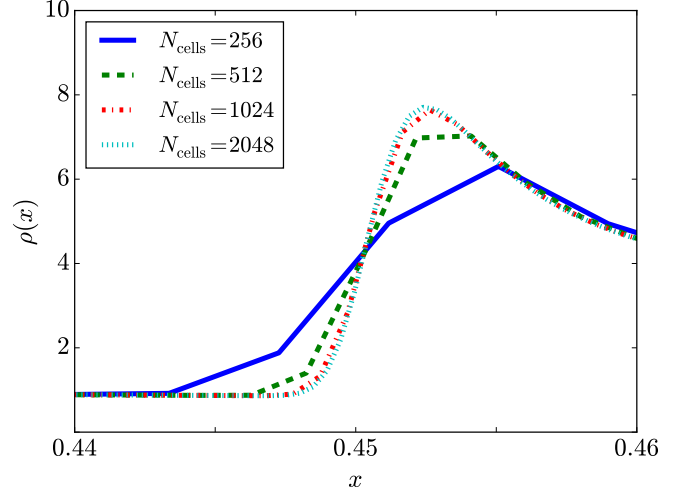


FIG. 21.— A zoomed-in view of one of the outer density caustics from the 1D, regularized and remapped problem with  $\sigma_i = 1$ , taken at  $a(t) = 1$ . The different colors correspond to different resolutions: solid blue line -  $N_{\text{cells}} = 256$ ; dashed green line -  $N_{\text{cells}} = 512$ ; dashed-dotted red line -  $N_{\text{cells}} = 1024$ ; dotted cyan line -  $N_{\text{cells}} = 2048$ .

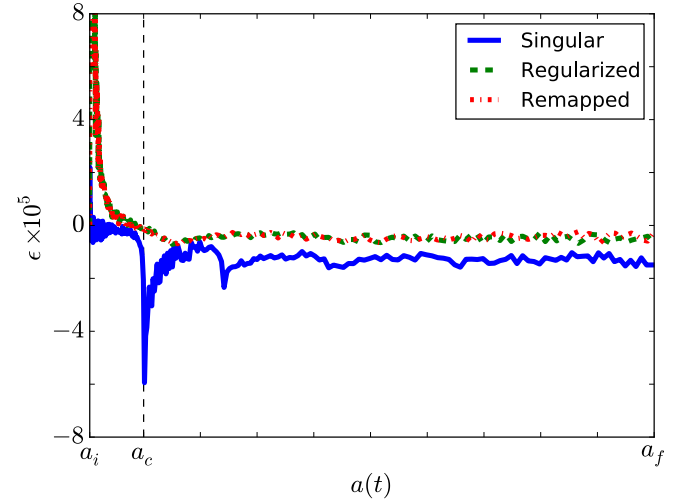


FIG. 22.— The energy conservation error  $\epsilon$  (see text for definition) as a function of  $a(t)$  for the singular run (blue solid line), the regularized run (green dashed line), and the regularized and remapped run (red dashed-dotted line) for the 1D pancake problem. In this figure, we have used the highest resolution data available, for which  $N_{\text{cells}} = 2048$ . The variation in the energy error with respect to mesh refinement is summarized in Table 6. The vertical dashed line marks the expansion factor at which the first caustic forms.

the  $N_{\text{cells}} = 256$  run, the energy error is actually the largest ( $\approx 0.1$  %) in the run with remapping; at this resolution, the discretization error in the remap step is clearly significant. By  $N_{\text{cells}} = 512$ ,  $\epsilon$  in the remapped run has become comparable to that in the singular run, and at still higher resolution, it becomes comparable to the error in the regularized run.

## 6.2. 2D Results

Finally, we repeat the oblique pancake problem with both regularization and remapping turned on. The run parameters are summarized in Table 7. We set  $\sigma_i = 1/16$

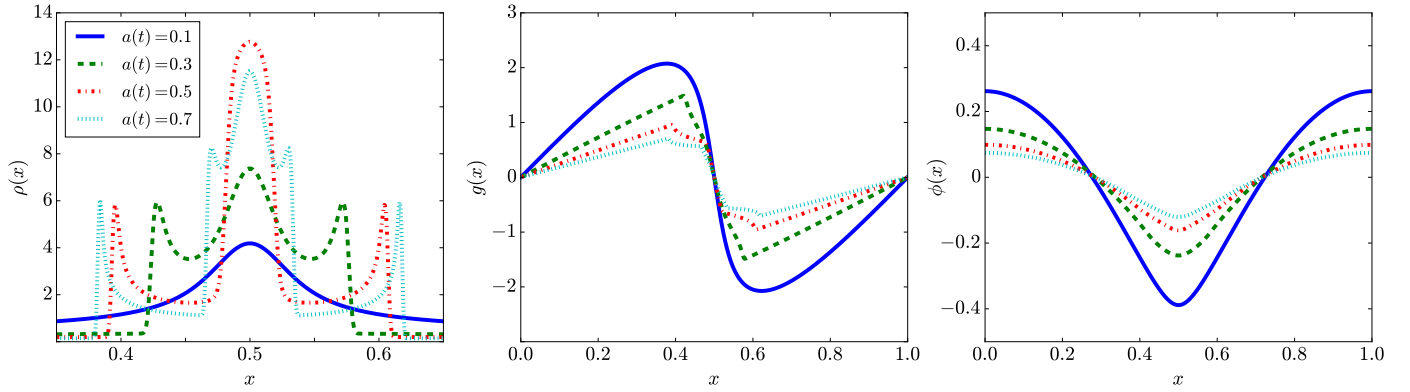


FIG. 23.— The time evolution of the solution to the  $\sigma_i = 1$  version of the 1D, regularized and remapped problem. The left panel shows the density, the middle panel the gravitational field, and the right panel the potential. The plotted curves are from the run with  $N_{\text{cells}} = 2048$ ; the  $N_{\text{cells}} = 1024$  run would be indistinguishable from the plotted solutions at the scales shown. The different lines correspond to the results at different expansion factors: solid blue line -  $a(t) = 0.1$ ; dashed green line -  $a(t) = 0.3$ ; dashed-dotted red line -  $a(t) = 0.5$ ; dotted cyan line -  $a(t) = 0.7$ .

and vary  $N_{\text{cells}}$  over  $\{128, 256\}$ . We use velocity-space mesh refinement during the remap phase so that  $\sigma_i$  is always resolved by at least  $N_\sigma = 1$  cells, up until a maximum refinement level of 2 is reached. We remap in linear increments in the expansion factor, with  $\Delta a_{\text{remap}} = 0.01$ .

The resulting particle positions are displayed in Figure 26. Unlike Figures 5 and 15, which show the particles positions at  $a = 1$  for the singular and regularized but not remapped runs, Figure 26 shows no sign of particle clumping along the caustic axes. Instead, a consequence of the remapping procedure is that the particle spacings are forced to be relatively constant throughout the populated regions of phase space. The information about the density is contained in the particle masses, instead. Figures 28 and 29 shows the grid-deposited density at  $a = 1$  for  $128^2$  and  $256^2$  run, respectively, where we have constructed the 2D density field by depositing the particles onto a  $256^2$  and  $512^2$  mesh using cloud-in-cell deposition. The unphysical clumping visible in Figures 6 and 16 is no longer visible in these plots.

It is important to note that this improvement is not simply a matter of using more particles per Poisson cell. Our regularized, remapped calculation of the oblique pancake (Section 6.2) uses approximately 380 particles per Poisson cell on average (the precise number varies with time), almost identical to the number used in the regularized but not remapped run ( $\approx 385$ ), and similar to the 256 used in the corresponding singular calculation. Simply sampling singular initial conditions with more particles in physical space does not lead to a convergent method - our results suggest that the initial conditions must be regularized *and* the extra particles must be arranged in the correct way in phase space.

However, while using many particles per cell is not *sufficient* to prevent artificial fragmentation, our results do suggest that it may be *necessary* in our scheme to carry around more particles per Poisson cell than is currently standard practice. In 3D, even using a relatively modest number of particles in each velocity space direction (say, 4, comparable to our low resolution runs in Section 6), would require  $4^3 = 64$  particles where one would exist in the comparable singular problem. Clearly, optimizing both the basic PIC scheme and our remapping code for such high particle-per-cell counts will be necessary before

TABLE 7  
SUMMARY OF PARAMETERS FOR THE REGULARIZED AND REMAPPED 2D PANCAKE RUNS

$\sigma_i$	$N_{\text{cells}}$	$N_\sigma$	$\Delta a_{\text{remap}}$	$N_x$	$N_v$	$C_{\text{exp}}$
1/16	$128^2$	1	0.01	256	128	$2.0 \times 10^{-2}$
1/16	$256^2$	1	0.01	512	128	$1.0 \times 10^{-2}$

our approach can be used for large-scale problems.

We examine our oblique solutions more quantitatively in Figure 27. Here, we plot the max norm of the quantity  $g_\perp/g_\parallel$ , where  $g_\perp$  is the component of the gravitational acceleration perpendicular to  $\mathbf{k}$  and  $g_\parallel$  is the corresponding component parallel to  $\mathbf{k}$ . To make a clean comparison between runs, we use the gravitational force as computed on the mesh points by the PIC algorithm in computing this quantity (Equation 13). In the exact solution, this quantity should be zero, but due to unavoidable discretization errors in the representation of the initial conditions, this will not be precisely true. A stable numerical scheme, however, will limit the ability of the asymmetries to grow with time. In all three runs, the degree of asymmetry in the force increases dramatically after  $a_{\text{caustic}}$ . However, we find that in our remapped run, this error is significantly reduced compared to the other two. This is true for both our low resolution and our high resolution calculations.

## 7. CONCLUSIONS AND CAVEATS

**In this paper, we have investigated the convergence of PIC schemes on a variety of “Zel’dovich Pancake” setups. Our conclusions are as follows. In one spatial dimension, the perfectly cold problem does not converge at the expected order of accuracy, due to singularities at the positions of the dark matter caustics. This is true regardless of the number of particles per Poisson cell employed. Once these singularities are removed through the introduction of a finite initial artificial velocity dispersion, PIC can converge at the expected order on the 1D problem. Additionally, by taking the limit as  $\sigma_i \rightarrow 0$ , we have shown that we recover the solution obtained in the perfectly cold limit. This gives us confidence that, in 1D,**

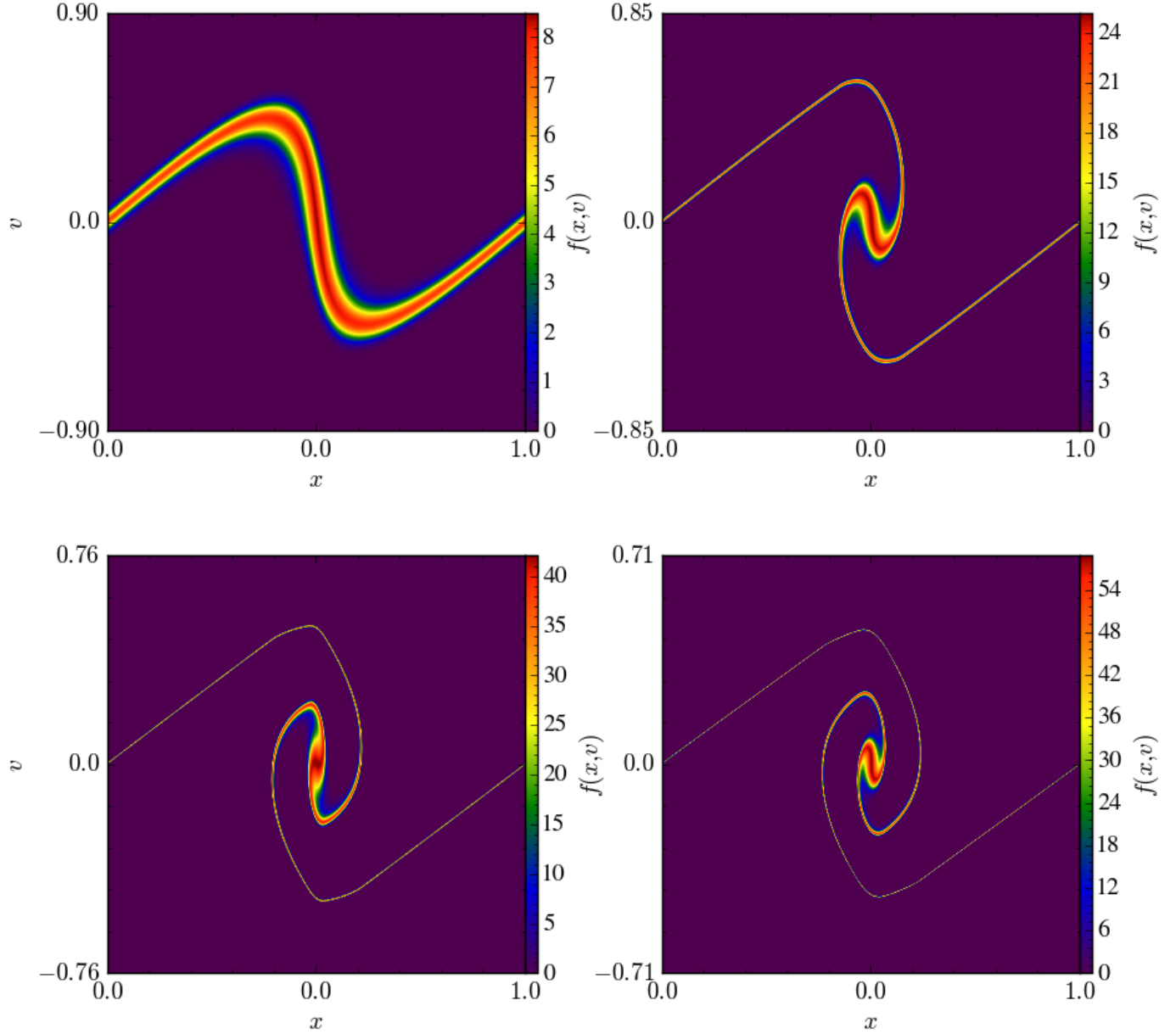


FIG. 24.— Time evolution of the distribution function for the  $\sigma_i = 1$  version of the regularized and remapped problem. Top Left -  $f(x, v)$  at the expansion factor  $a(t) = 0.1$ . Top Right - same, at  $a(t) = 0.3$ . Bottom Left - same, at  $a(t) = 0.5$ . Bottom Right - same, at  $a(t) = 0.7$ . The bounds of the images in velocity space have been adjusted slightly in each plot.

the results obtained with singular initial data are in fact valid solutions to the Vlasov-Poisson equation in the zero-temperature limit.

On the 2D oblique pancake problem, however, the basic PIC method with cold initial data does not converge to a valid solution of the Vlasov-Poisson equation. Instead, the solutions exhibit small-scale fragmentation that does not improve with mesh refinement. This problem is not solved by increasing the number of particles per Poisson cell, nor is it solved by using the artificially warm initial data. However, the particle remapping algorithm we describe does improve the accuracy of PIC on this problem, reducing the off-axis component of the gravitational acceleration significantly when applied 100 times over between

$a(t) = 1/200$  and  $a(t) = 1$ .

Given that the limiting case of the regularized solutions agree with the solution obtained using cold initial data in 1D, and given that regularization does not, by itself, prevent artificial fragmentation in 2D, a possible conclusion is that the artificial velocity dispersion is not necessary in and of itself, but is primarily important in that allows the use of the plasma-style remapping algorithm. If this is the case, it may be possible to construct a remapping algorithm that does not require regularization, by e.g. exploiting the continuity of the distribution function in phase space as in Hahn et al. (2013). However, whether designing such an algorithm is possible is an open question.



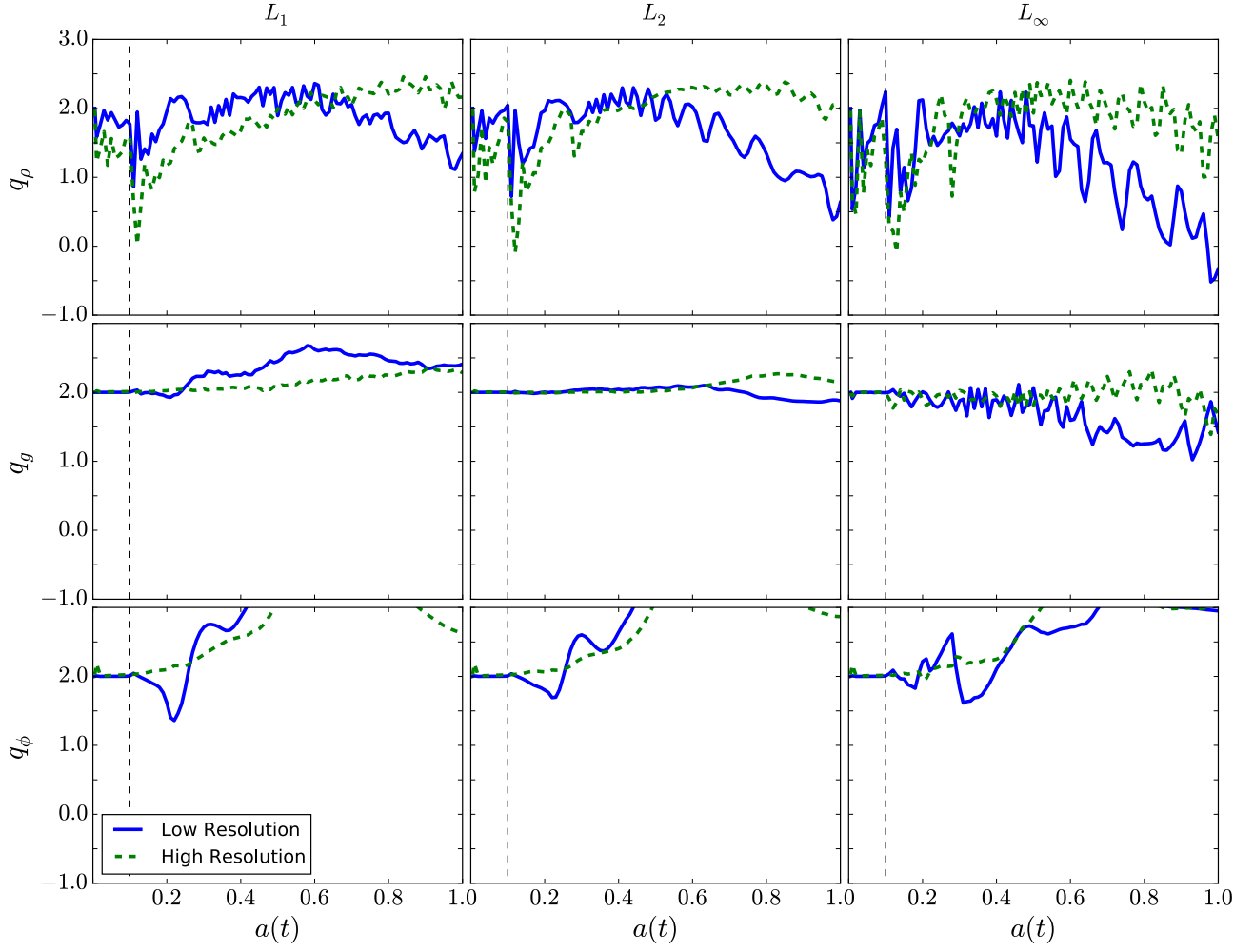


FIG. 25.— The same as Figure 4, but for the  $\sigma_i = 1$  version of the regularized, remapped problem. The blue line compares the result for the  $N_\sigma = 2, 4$ , and 8 runs, while the green line is for  $N_\sigma = 4, 8$ , and 16.

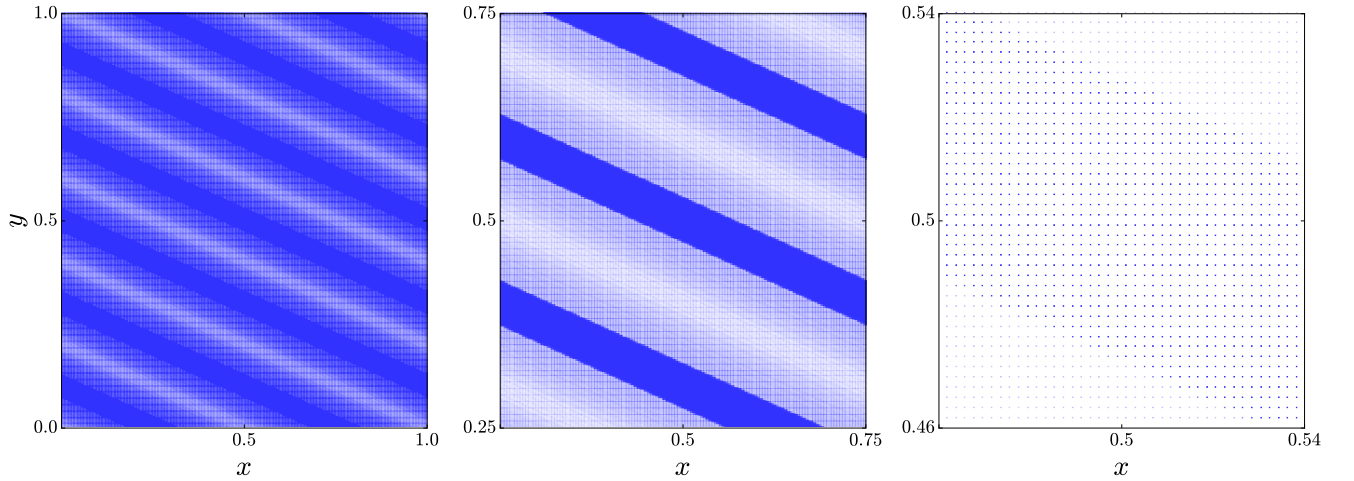


FIG. 26.— Particle  $x$  and  $y$  positions at  $a(t) = 1$  for the  $\sigma_i = 1/16$  version of the regularized and remapped oblique pancake problem. The scales shown are the same as in Figure 5

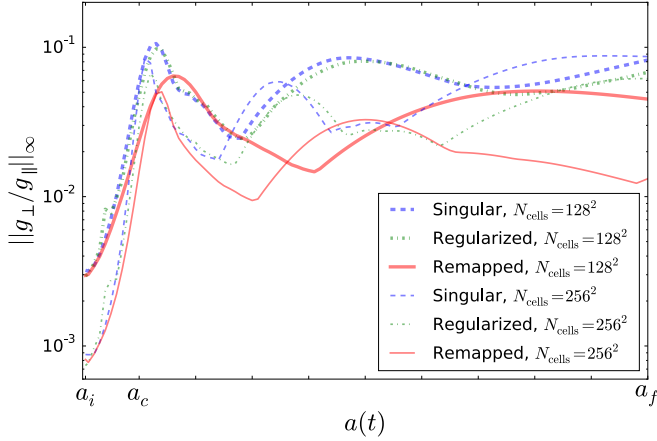


FIG. 27.— A measure of the asymmetry in the gravitational force as a function of time for the singular (dashed blue lines), regularized without remapping (dash-dotted green lines) and regularized with remapping (solid red lines) versions of the oblique pancake problem. Low resolution results are shown using thick lines, and high resolution results are shown using thin lines. The quantity on the y-axis is the maximum off-axis (that is, perpendicular to the perturbation axis) force as computed on the mesh in units of the maximum on-axis (along the perturbation) force.

There are a number of caveats to our work that bear mentioning. The first is that we have restricted our attention to various configurations of the Zel'dovich Pancake problem. We have not yet applied our technique to other, more realistic test problems or to a full structure formation calculation. Second, we have restricted

ourselves to working in one- and two-dimensional spaces only. Work towards creating a three-dimensional version of our remapping algorithm is in progress. Finally, while our scheme does use adaptivity in velocity space, we have not yet introduced AMR in the spatial directions, either for solving the Poisson equation or for generating the particle positions during remapping. Such a modification would clearly be advantageous, in that it would concentrate both the particles and the resolution of the force grid towards the regions where high resolution is most needed. However, this was not necessary for the relatively simple problems considered in this paper.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Program and performed under the auspices of the U.S. Department of Energy by Lawrence Berkeley National Laboratory under Contract DE-AC02-05CH11231. The code used to run the simulations in this paper was built using Chombo's ParticleTools library, which was originally developed by Francesco Miniati for the CHARM code. AM wishes to thank Anshu Dubey, Daniel Graves, and Daniel Martin for sharing their time and expertise with Chombo, and the anonymous referee for a thoughtful report that improved the quality of the paper. This work relied heavily on yt<sup>4</sup> (Turk et al. 2011) and the core scientific Python packages, including IPython<sup>5</sup> (Pérez & Granger 2007), NumPy<sup>6</sup> (van der Walt et al. 2011), and Matplotlib<sup>7</sup> (Hunter 2007), for data analysis and plotting.

## REFERENCES

- Adams, M., Colella, P., Graves, D. T., et al. 2015, Chombo Software Package for AMR Applications - Design Document, Tech. rep., Lawrence Berkeley National Laboratory LBNL-6616E
- Alimi, J.-M., Bouillot, V., Rasera, Y., et al. 2012, ArXiv e-prints, arXiv:1206.2838
- Alimi, J.-M., Bouillot, V., Rasera, Y., et al. 2012, ArXiv e-prints, 1206.2838
- Almgren, A. S., Bell, J. B., Lijewski, M. J., Lukić, Z., & Van Andel, E. 2013, ApJ, 765, 39
- Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R., & Ostriker, J. P. 1995, Computer Physics Communications, 89, 149
- Chaniotis, A. K., Poulikakos, D., & Koumoutsakos, P. 2002, Journal of Computational Physics, 182, 67
- Chen, Y., Parker, S. E., Rewoldt, G., et al. 2008, Physics of Plasmas, 15, 055905
- Cottet, G.-H., & Koumoutsakos, P. D. 2000, Vortex Methods: Theory and Practice (Cambridge University Press)
- Denavit, J. 1972, Journal of Computational Physics, 9, 75
- Habib, S., Morozov, V., Finkel, H., et al. 2012, ArXiv e-prints, 1211.4864
- Hahn, O., & Abel, T. 2011, Monthly Notices of the Royal Astronomical Society, 415, 2101
- Hahn, O., Abel, T., & Kaehler, R. 2013, Monthly Notices of the Royal Astronomical Society, 434, 1171
- Heitmann, K., Higdon, D., White, M., et al. 2009, The Astrophysical Journal, 705, 156
- Heitmann, K., Ricker, P. M., Warren, M. S., & Habib, S. 2005, The Astrophysical Journal Supplement Series, 160, 28
- Heitmann, K., White, M., Wagner, C., Habib, S., & Higdon, D. 2010, The Astrophysical Journal, 715, 104
- Heitmann, K., Lukić, Z., Fasel, P., et al. 2008, Computational Science and Discovery, 1, 015003
- Hockney, R. W., & Eastwood, J. W. 1981, Computer Simulation Using Particles
- Hopkins, P. F. 2014, ArXiv e-prints, arXiv:1409.7395
- Hunter, J. 2007, Computing in Science and Engineering, 9, 90
- Ishiyama, T., Nitadori, K., & Makino, J. 2012, ArXiv e-prints, arXiv:1211.4406
- Kim, J.-h., Abel, T., Agertz, O., et al. 2014, The Astrophysical Journal Supplement Series, 210, 14
- Koumoutsakos, P. 1997, Journal of Computational Physics, 138, 821
- Krasny, R. 1986, Journal of Computational Physics, 65, 292
- Kravtsov, A. V., Klypin, A. A., & Khokhlov, A. M. 1997, The Astrophysical Journal Supplement Series, 111, 73
- Kuhlen, M., Diemand, J., & Madau, P. 2008, The Astrophysical Journal, 686, 262
- Kuhlen, M., Vogelsberger, M., & Angulo, R. 2012, Physics of the Dark Universe, 1, 50
- Melott, A. L., Shandarin, S. F., Splinter, R. J., & Suto, Y. 1997, The Astrophysical Journal Letters, 479, L79
- Miniati, F., & Colella, P. 2007, Journal of Computational Physics, 227, 400
- Monaghan, J. J. 1985, Computer Physics Reports, 3, 71
- Peebles, P. J. E. 1993, Principles of Physical Cosmology
- Pérez, F., & Granger, B. E. 2007, Computing in Science and Engineering, 9, 21
- Skillman, S. W., Warren, M. S., Turk, M. J., et al. 2014, ArXiv e-prints, 1407.2600
- Springel, V. 2005, Monthly Notices of the Royal Astronomical Society, 364, 1105
- Stadel, J. G. 2001, Ph.D. Thesis, 21
- Teyssier, R. 2002, Astronomy and Astrophysics, 385, 337
- Turk, M. J., Smith, B. D., Oishi, J. S., et al. 2011, The Astrophysical Journal Supplement Series, 192, 9
- Vadlamani, S., Parker, S. E., Chen, Y., & Kim, C. 2004, Computer Physics Communications, 164, 209

<sup>4</sup> <http://yt-project.org>

<sup>5</sup> <http://ipython.org>

<sup>6</sup> <http://www.numpy.org>

<sup>7</sup> <http://matplotlib.org>

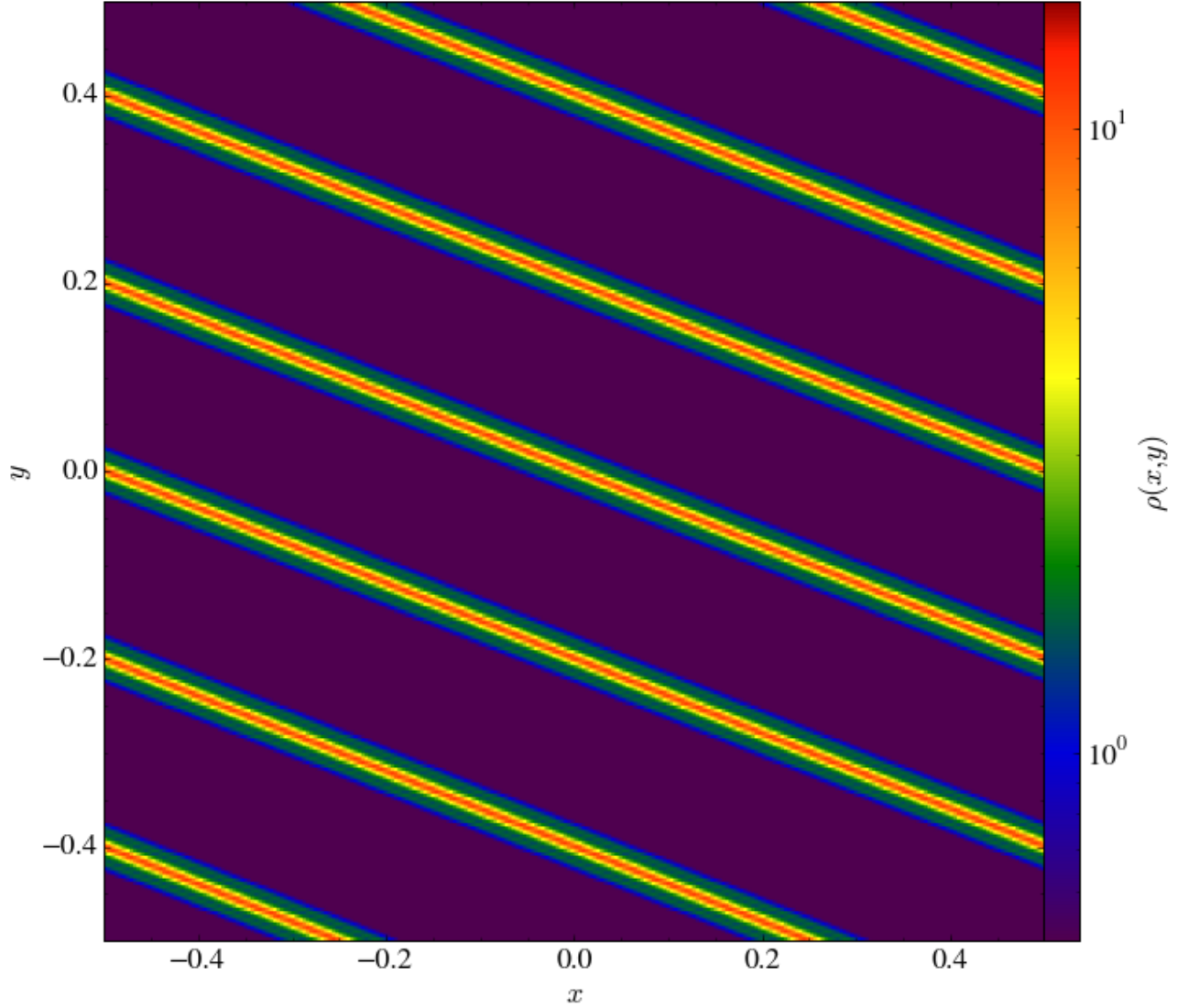


FIG. 28.— The solution for the density at  $a = 1$  from the low-resolution calculation of the oblique pancake problem using both regularized initial conditions and remapping. The color scale is the same as in Figure 6. For details of the parameter choices we adopted, see text.

van der Walt, S., Colbert, S., & Varoquaux, G. 2011, *Computing in Science and Engineering*, 13, 22  
Wang, B., Miller, G., & Colella, P. 2011, *SIAM Journal on Scientific Computing*, 33, 3509  
—, 2012, *SIAM Journal on Scientific Computing*, 34, B909  
Wang, J., & White, S. D. M. 2007, *Monthly Notices of the Royal Astronomical Society*, 380, 93

Warren, M. S. 2013, *ArXiv e-prints*, 1310, 4502  
Yoshikawa, K., Yoshida, N., & Umemura, M. 2013, *The Astrophysical Journal*, 762, 116

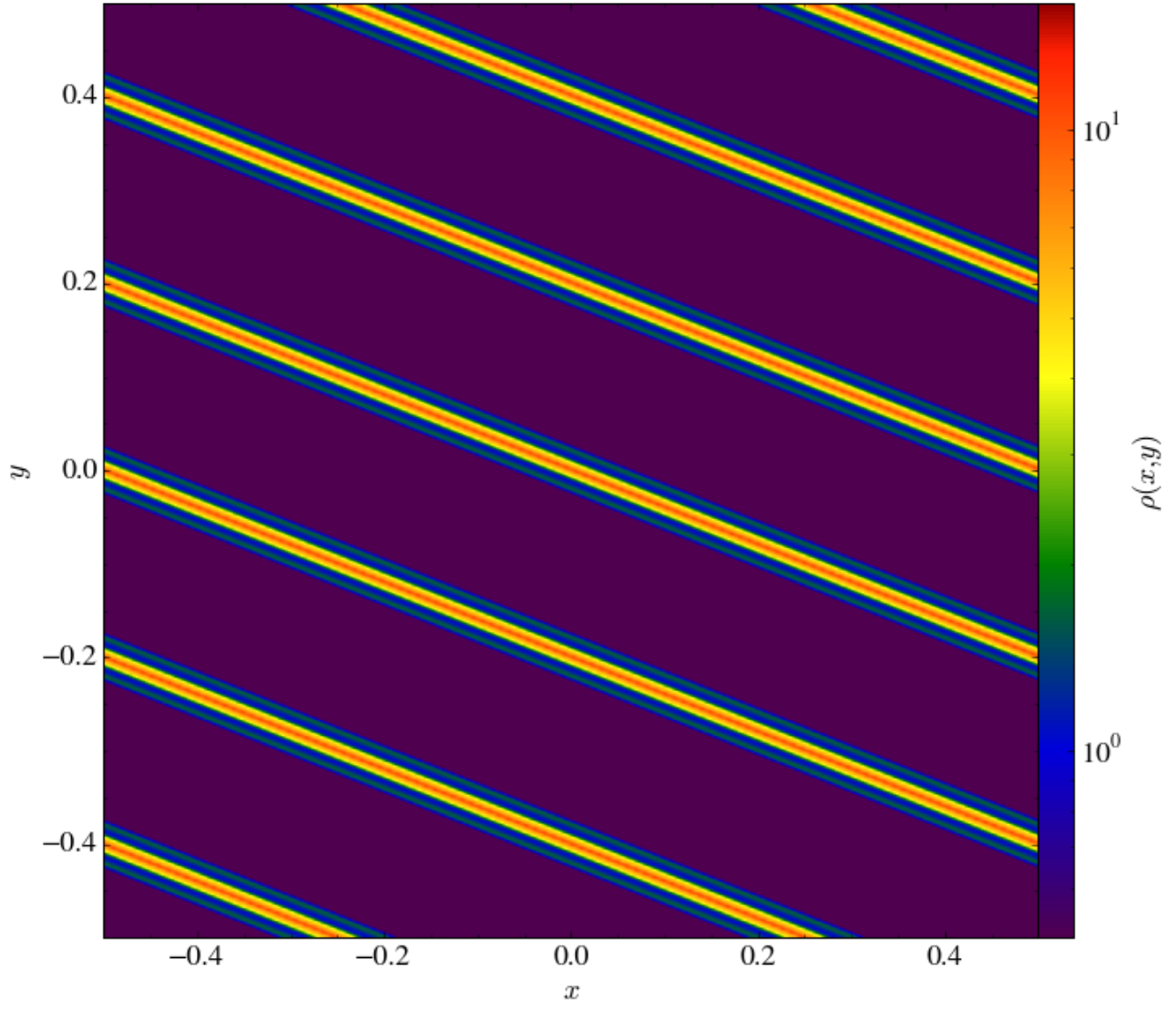


FIG. 29.— The solution for the density at  $a = 1$  from the high-resolution calculation of the oblique pancake problem using both regularized initial conditions and remapping. The color scale is the same as in Figure 6. For details of the parameter choices we adopted, see text.