



ELSEVIER

Journal of Computational and Applied Mathematics 109 (1999) 123–152

JOURNAL OF  
COMPUTATIONAL AND  
APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

# Star formation with 3-D adaptive mesh refinement: the collapse and fragmentation of molecular clouds

Richard I. Klein

*Lawrence Livermore National Laboratory, and Berkeley Department of Astronomy, University of California,  
P.O. Box 808, L-23 Livermore, CA 94550, USA*

Received 27 May 1998; received in revised form 5 March 1999

## Abstract

We describe a powerful methodology for numerical solution of 3-D self-gravitational hydrodynamics problems with unprecedented resolution. This code utilizes the technique of local adaptive mesh refinement (AMR), employing multiple grids at multiple levels of resolution. These grids are automatically and dynamically added and removed as necessary to maintain adequate resolution. This technology allows solution of problems that would be prohibitively expensive with methods using fixed resolution, and it is more versatile and efficient than competing methods of achieving variable resolution. The application of this technique to simulate the collapse and fragmentation of a molecular cloud, a key step in star formation is discussed. The simulation involves many orders of magnitude of variation in length scale as fragments form. In this paper we describe the methodology and present illustrative applications for both isothermal and nonisothermal cloud collapse. We describe the numerical Jeans condition, a new criterion for stability of self-gravitational gas dynamic problems. We find that the uniformly rotating, spherical clouds treated here first collapse to disks in the equatorial plane and then, in the presence of applied perturbations, form *filamentary singularities* that do not fragment while isothermal. As the collapse enters the non-isothermal phase, we show the evolutionary sequence that leads to the formation of a binary system consisting of protostellar cores surrounded by distinct protostellar disks. The scale of the disks, of order 100 AU, is consistent with observations of gaseous disks surrounding single T-Tauri stars and debris disks surrounding systems such as Beta Pictoris. © 1999 Published by Elsevier Science B.V. All rights reserved.

**Keywords:** Gravitation; Hydrodynamics; ISM: clouds; Stars: formation

## 1. Introduction

Self-gravitational hydrodynamics involves the collapse and fragmentation of an unstable cloud into smaller condensations. Fragmentation is essential to understanding the formation of stars and galaxies, yet gaseous flows undergoing fragmentation naturally involve a substantial three-dimensional

*E-mail address:* klein@radhydro.berkeley.edu (R.I. Klein)

variation in length scale. This enormous dynamic range presents a formidable obstacle to obtaining an accurate numerical solution, as the flow must remain well-resolved throughout the evolution. The resolution required after considerable collapse is well in excess of that needed initially. Dynamic range in scale of  $10^4$  or more is not unusual. Fixed-resolution methods can not be used to simulate such a 3-D collapse in a practical amount of time using current computers. Even if fixed-resolution simulation were feasible, use of such an approach would remain grossly inefficient necessitating variable-resolution methods.

Prior to this work, three basic approaches had been developed for employing variable resolution in solution of 3-D self-gravitational problems. The smoothed particle hydrodynamics (SPH) technique is a grid-less, Lagrangian technique in which gravitating particles of fixed mass carry fluid properties (see, e.g., [30]). For resolution yielding computational times comparable to grid-based codes, SPH provides poorer resolution of shocks and poorer resolution of low-density regions [19]. Shocks are an integral feature of calculations of astrophysical hydrodynamics, and this restricts the utility of SPH for such problems [21]. In addition, while the spatial redistribution of particles at each step yields a variable spatial resolution, the mass resolution of SPH codes remains fixed. Recent SPH work in the study of star formation has been carried out by [1]. Another approach to achieve variable resolution is simply to utilize a spherical grid that naturally affords increased resolution in the form of decreased cell volume about the origin. When coupled with a nonuniform radial cell dimension and rezoning, this method is useful for studying collapse that produces structure near the origin. Spherical codes have been used in recent work by Boss and Sigalotti and Klapp [8,33,34]. Finally, a fixed multiple-grid method is utilized by Burkert and Bodenheimer [9]. In this scheme, a series of smaller grids of finer resolution are concentrically positioned about the center of the computational volume. These grids remain fixed throughout the calculation. In typical problems, this method over-resolves the cloud center during the early stages of collapse. This method can not treat the formation of structure outside the preset locations of the finer grids. An early development in the use of adaptive meshes for one-dimensional problems [40] utilized a grid equation that determined where grid refinement was necessary. This scheme allowed a large dynamic range to be achieved in 1-D spherical collapse in the comoving frame of the fluid which had the advantage of being able to investigate highly resolved 1-D radiation hydrodynamic flows, but had the disadvantage that grid points used in highly refined regions were borrowed from other regions of the flow thereby deteriorating accuracy in those regions. A substantial further disadvantage of this approach is the extreme complication of how to generalize it to multi-dimensions.

Our multi-dimensional adaptive mesh refinement (AMR) scheme utilizes grids at different levels of resolution. Linear resolution varies by integral refinement factors — usually 4 — between levels, and a given grid is always fully contained within one at the next coarser level (excluding the coarsest grid). The origin of the method stems from the seminal work of Berger and Oliger [5] and Berger and Colella [4]. Unlike a fixed nested mesh method, however, our AMR method can employ multiple spatially unconnected grids at a given level of refinement. Most importantly, our AMR method dynamically resizes and repositions these grids and inserts new, finer ones within them according to adjustable refinement criteria. Fine grids are automatically removed as flow conditions require less resolution. We have applied our method to great effect, prior to the new addition of self-gravity, to study astrophysical problems including the interaction of supernova blastwaves with interstellar clouds [21–23], X-ray heated coronae and winds from accretion disks [41], and the collision of interstellar clouds [24]. These calculations have been made with unprecedented high resolution.

Star formation ultimately involves the collapse of an interstellar molecular cloud from an initial density of  $\rho \leq 10^{-19} \text{ g cm}^{-3}$  to a final mean stellar density of  $\rho \geq 10^{-1} \text{ g cm}^{-3}$ . The associated length scale  $R$  varies from  $10^{17}$  to  $10^{11} \text{ cm}$ . An initial mass of gas in excess of the Jeans mass  $M_J \propto c_s^3 \rho^{-1/2}$  (where  $c_s$  is the soundspeed) will collapse isothermally until  $\rho \sim 10^{-12} \text{ g cm}^{-3}$  (see, e.g., the nonisothermal calculations of Myhill and Boss [31]). At larger densities the cloud becomes significantly opaque to its own cooling radiation, resulting in a trapping of the heat produced by gravitational compression [35]. The accompanying increase in temperature acts to slow the collapse. However, before the cloud becomes nonisothermal, the Jeans mass scales as  $\rho^{-1/2}$ , opening the possibility of a collapsing cloud fragmenting into yet smaller units. These subunits may subfragment, with the process repeating hierarchically until the gas ceases to be isothermal and thermal pressure stops the collapse.

The fragmentation process may be crucial to establishing key parameters of binary stars: the distributions of mass ratios, periods, and orbital eccentricities. It may also be essential to the formation of larger groups of stars and to the determination of the stellar initial mass function. In this paper we use initial conditions taken from the literature that were designed to model binary formation.

We present the methodology behind our 3-D AMR self-gravity code and describe applications of our work in the collapse and fragmentation of molecular clouds. Our discussion closely follows [37]. The structure of the paper is as follows. In Section 2 we discuss the individual components of the code: the hyperbolic solver, elliptic solver, and AMR framework. Our main refinement criterion, the Jeans condition, is a key component of the methodology for studying isothermal problems and was discussed in detail by Truelove et al. [36] in the context of the collapse of a centrally condensed cloud. We also discuss the use of Richardson extrapolation to constrain the truncation error in an AMR calculation. In Section 3 we present applications to uniform as well as centrally condensed isothermal clouds involving azimuthal  $m=2$  mode perturbations. In each case we set a new benchmark of accuracy by which other solutions with other codes may be measured. Our simulations using AMR to study perturbed isothermal clouds produce either singular or binary filaments, in contrast with the results of previous calculations that used fixed finest resolution and produced multiple quasi-spherical, artificially viscous condensations. Use of ever-finer resolution, guided by the Jeans condition, allows solutions free from artificial perturbations resulting in artificial fragmentation. Our results provide strong confirmation of recent work on isothermal filament formation by Inutsuka and Miyama [15,16]. We present preliminary simulations for nonisothermal collapse and show for the first time the formation of protostellar disks surrounding binary cores.

## 2. Methodology

### 2.1. Self-gravitational hydrodynamics

The basic governing equations of our model are the Euler equations of hydrodynamics in 3-D, including effects of self-gravitation, and Poisson's equation for the gravitational potential:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (1)$$

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla P - \rho \nabla \phi, \quad (2)$$

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot [(\rho E + P)\mathbf{v}] = \rho \mathbf{v} \cdot \nabla \phi, \quad (3)$$

$$\nabla^2 \phi = 4\pi G \rho. \quad (4)$$

Here,  $\rho$  is the mass density,  $\mathbf{v}$  is the velocity vector in 3-D,  $P$  is the thermal pressure,  $\phi$  is the gravitational potential, and  $G$  is the universal constant of gravitation. The total nongravitational energy per unit mass,  $E$ , is related to the internal energy per unit mass,  $e$ , by

$$E = e + \frac{1}{2} \mathbf{v}^2. \quad (5)$$

In turn,  $\rho e$  and  $P$  are related by an equation of state, for which adopt the ideal gas law

$$P = (\gamma - 1)\rho e. \quad (6)$$

As written above, there are eight equations in eight unknowns. We assume periodic boundary conditions on these equations.

Our code methodology consists of three components to efficiently solve this system. The first component is a hyperbolic solver that employs an implementation of the Godunov method (see [12,26]) for solution of the Euler equations of gas dynamics. This solver accounts for the gravitational terms in these equations in a predictor–corrector fashion. The second major component of our code methodology is an elliptic solver that utilizes an AMR multigrid method to solve Poisson’s equation. These elements operate within the third component, an adaptive mesh refinement framework. The flow is discretized onto multiple grids at multiple levels of resolution according to preset refinement criteria applied at each step in the calculation. For the calculations described in this paper, we use a physically motivated refinement criterion, the Jeans condition. Both of the solvers must take into account the presence of a hierarchy of grids. However, as might be expected, this is substantially more complicated in the case of the elliptic solver than in the case of the hyperbolic solver. Below, we describe each of these components in some detail.

## 2.2. Hyperbolic solver

### 2.2.1. Overview of Godunov implementation

We can express the hydrodynamics equations (1)–(3) more simply by using a state-vector formulation. In one spatial dimension, this set of equations can be written

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S}. \quad (7)$$

The quantity  $\mathbf{Q}$  is the *state vector*, the components of which are the densities of mass, momentum, and total nongravitational energy:

$$\mathbf{Q} \equiv \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix}. \quad (8)$$

The flux vector  $\mathbf{F}$  is

$$\mathbf{F} \equiv \begin{bmatrix} \rho v \\ \rho v^2 + P \\ (\rho E + P)v \end{bmatrix} \quad (9)$$

and the source vector  $\mathbf{S}$  is

$$\mathbf{S} \equiv \begin{bmatrix} 0 \\ -\rho \frac{\partial \phi}{\partial x} \\ \rho v \frac{\partial \phi}{\partial x} \end{bmatrix}. \quad (10)$$

To solve the full 3-D system of hydrodynamics equations (1)–(3), we use an operator-split method in which we solve Eq. (7) for each coordinate direction independently in a cyclic sequence. In 3-D, this operator splitting has the form

$$\mathbf{Q}^{n+2} = L_x L_y L_z [L_z L_y L_x (\mathbf{Q}^n)], \quad (11)$$

where the superscript  $n$  indicates the  $n$ th time level and  $L_i$  is the update operator in the  $i$ th coordinate direction. That is, each three-dimensional update step from  $t^n$  to  $t^{n+1}$  consists of 3 one-dimensional update steps, one per coordinate direction. At the end of the cycle from  $t^n$  to  $t^{n+2}$ , the solution is second-order accurate. This is, of course, true only if the same timestep is used in the two steps involved in the cycle which advances data from  $t^n$  to  $t^{n+2}$ . For the 1-D update operators  $L_i$ , we use a scheme based on the higher-order Godunov method described by Colella and Glaz [12] and Bell et al. [3].

We can write Eq. (7) in finite difference form as

$$\mathbf{Q}_j^{n+1} = \mathbf{Q}_j^n - \frac{\Delta t}{\Delta x} (\mathbf{F}_{j+1/2}^{n+1/2} - \mathbf{F}_{j-1/2}^{n+1/2}) + \Delta t \mathbf{S}_j^{n+1/2}. \quad (12)$$

In this paper, the integral index  $j$  denotes a cell center, and the half-integral indices  $j \pm \frac{1}{2}$  identify boundaries between adjacent cells. To carry out this update of the state vector, we must know the flux vector at the half timestep and left and right cell edges, as well as the cell-centered source vector at the half timestep. The timestep is set by the Courant condition using a Courant number of 0.5.

In general terms, our scheme for obtaining  $\mathbf{F}_{j\pm 1/2}^{n+1/2}$  and  $\mathbf{S}_j^{n+1/2}$  can be thought of as a predictor–corrector method. It consists of several steps: (1) Rewrite Eq. (7) in a quasilinear form that directly incorporates the equation of state and uses a different state vector. (2) Construct a fourth-order monotone-limited approximation to the spatial variation of this state vector and modify it to suppress numerical effects. (3) Use this approximation to estimate conditions at  $x_{j\pm 1/2}$  and  $t^{n+1/2}$ . (4) Use these as initial conditions in a local Riemann problem at each cell edge, and solve each problem with an approximate numerical solver. The desired vector  $\mathbf{F}_{j\pm 1/2}^{n+1/2}$  results from the Riemann solution. (5) Modify the fluxes  $\mathbf{F}_{j\pm 1/2}^{n+1/2}$  by adding artificial viscosity. (6) Use Eq. (12) to construct a partially updated state vector lacking only the contribution of  $\mathbf{S}$ . (7) The first (density) component of  $\mathbf{S}$  is always zero, so that the final value of  $\rho_j^{n+1}$  is fortuitously obtained in what is an otherwise partial update step. Use this fact to construct  $\rho_j^{n+1/2}$  and, using this density, call the elliptic solver for a solution for  $(\nabla \phi)_j^{n+1/2}$ . With the cell-centered density and potential gradient at  $t^{n+1/2}$ , construct  $\mathbf{S}_j^{n+1/2}$  in full. This completes the method of obtaining  $\mathbf{F}_{j\pm 1/2}^{n+1/2}$  and  $\mathbf{S}_j^{n+1/2}$  that were the unknowns in the

update operators  $L_i$ . We effect the complete update of the state vector from  $\mathbf{Q}_j^n$  to  $\mathbf{Q}_j^{n+1}$  by using Eq. (12). Below, we describe each of these steps in more detail.

### 2.2.2. Characteristic analysis

The first step of the Godunov method is to use data defined at  $x_j$  and  $t^n$  to estimate the state of the gas at  $x_{j\pm 1/2}$  and  $t^{n+1/2}$ . This is accomplished by the method of characteristic tracing, described below. To carry out this method, we must first perform a characteristic analysis of Eq. (7). We begin by rewriting this equation plus the equation of state in quasilinear form

$$\frac{\partial \mathbf{q}}{\partial t} + \mathcal{A} \frac{\partial \mathbf{q}}{\partial x} = \mathbf{s}. \quad (13)$$

The quantity  $\mathbf{q}$  is a state vector of *primitive* variables:

$$\mathbf{q} \equiv \begin{bmatrix} \rho \\ v \\ P \\ \rho e \end{bmatrix}. \quad (14)$$

It contains one more dimension than the state vector  $\mathbf{Q}$  by virtue of its directly incorporating the equation of state into the system, a tactic that results in a minimization in the number of equation of state evaluations over the course of the integration. The source vector for this quasilinear system is purely gravitational in nature:

$$\mathbf{s} \equiv \begin{bmatrix} 0 \\ \frac{\partial \phi}{\partial x} \\ 0 \\ 0 \end{bmatrix}. \quad (15)$$

The matrix  $\mathcal{A}$  is

$$\mathcal{A} \equiv \begin{bmatrix} v & \rho & 0 & 0 \\ 0 & v & \frac{1}{\rho} & 0 \\ 0 & \gamma P & v & 0 \\ 0 & \rho e + P & 0 & v \end{bmatrix}. \quad (16)$$

We can construct a similarity transformation so that  $\mathcal{A} = \mathcal{R} \Lambda \mathcal{L}$ . Here  $\Lambda$  is a  $4 \times 4$  diagonal matrix of eigenvalues ( $\{\Lambda\}_{ij} \equiv \lambda_i \delta_{ij}$ ),  $\mathcal{R}$  is a  $4 \times 4$  matrix whose columns are formed from the set of right eigenvectors, and  $\mathcal{L} = \mathcal{R}^{-1}$  is a  $4 \times 4$  matrix whose rows are formed from the set of left eigenvectors. The eigenvalues, eigenvectors, and this similarity transformation will prove useful in the characteristic tracing step.

### 2.2.3. Characteristic tracing

The results of the characteristic analysis allow us to perform the characteristic tracing step. In characteristic tracing, we seek to use the state of the gas at each cell center at time  $t^n$  to estimate the state of the gas on each side of each cell interface at time  $t^{n+1/2}$ . As a first step, we can obtain

an equation for  $\mathbf{q}_{j+1/2}^{n+1/2}$  in terms of quantities at  $x_j$  and  $t^n$  by expanding the solution for  $\mathbf{q}$  in a Taylor series about  $x_j$  and  $t^n$ :

$$\mathbf{q}_{j+1/2}^{n+1/2} = \mathbf{q}_j^n + \frac{1}{2}\Delta x \left( \frac{\partial \mathbf{q}}{\partial x} \right)_j^n + \frac{1}{2}\Delta t \left( \frac{\partial \mathbf{q}}{\partial t} \right)_j^n. \quad (17)$$

To solve this, we first construct an approximate finite difference representation of the partial differential equation for  $\mathbf{q}$  (Eq. (13)):

$$\left( \frac{\partial \mathbf{q}}{\partial t} \right)_j^n + \mathcal{A}_j^n \left( \frac{\partial \mathbf{q}}{\partial x} \right)_j^n = \mathbf{s}_j^n. \quad (18)$$

It is necessary to include the effects of gravity in the characteristic tracing in order to attain a solution of second-order accuracy. Solving Eq. (18) for the temporal derivative and inserting the result in Eq. (17), we obtain

$$\mathbf{q}_{j+1/2}^{n+1/2} = \mathbf{q}_j^n + \frac{1}{2} \left( \mathcal{I} - \frac{\Delta t}{\Delta x} \mathcal{A}_j^n \right) \left( \frac{\partial \mathbf{q}}{\partial x} \right)_j^n \Delta x + \frac{1}{2} \Delta t \mathbf{s}_j^n, \quad (19)$$

where  $\mathcal{I}$  is the identity matrix. We consider  $\partial \mathbf{q} / \partial x$ .

We can define a difference vector

$$\delta \mathbf{q}_j^n \equiv \left( \frac{\partial \mathbf{q}}{\partial x} \right)_j^n \Delta x, \quad (20)$$

which is termed the *slope* of  $\mathbf{q}_j^n$ . We construct a fourth-order approximation to this function at cell centers. It is well known that higher-order methods produce significant post-shock oscillations if the raw slopes computed in this manner are utilized. We have therefore implemented the slope-flattening procedure of Colella and Woodward [13]. In flattening the slope, we form

$$\delta^{\text{lim}} \mathbf{q}_j^n = \chi_j^n \delta \mathbf{q}_j^n \quad (21)$$

by specifying flattening coefficients  $0 \leq \chi_j^n \leq 1$ . Following the slope flattening procedure, we represent the flattened slope as an expansion based on the right eigenvectors  $\mathbf{r}_i$ :

$$\delta^{\text{lim}} \mathbf{q}_j^n = \sum_{i=1}^4 (\alpha_i)_j^n (\mathbf{r}_i)_j^n. \quad (22)$$

The expansion coefficients  $\alpha_i$  are set according to

$$(\alpha_i)_j^n \equiv (\mathbf{l}_i)_j^n \cdot \delta^{\text{lim}} \mathbf{q}_j^n. \quad (23)$$

Inserting the right-eigenvector slope expansion in Eq. (19), we obtain the following expression for the state at  $x_{j+1/2}$  and  $t^{n+1/2}$ :

$$\mathbf{q}_{j+1/2}^{n+1/2} = \mathbf{q}_j^n + \frac{1}{2} \left( \mathcal{I} - \frac{\Delta t}{\Delta x} \mathcal{A}_j^n \right) \sum_{i=1}^4 (\alpha_i)_j^n (\mathbf{r}_i)_j^n + \frac{1}{2} \Delta t \mathbf{s}_j^n. \quad (24)$$

For input to the Riemann problem, we need the state on each side of the cell interface at  $x_{j+1/2}$ . We can obtain the left (–) state by discarding components in the  $\partial \mathbf{q} / \partial x$  term corresponding to

characteristics that do not originate in the cell to the left of  $x_{j+1/2}$ . To accomplish this we introduce a left projection operator and we apply this operator to Eq. (24) to obtain the left state at  $x_{j+1/2}$ :

$$\mathbf{q}_{j+1/2,-}^{n+1/2} = \mathbf{q}_j^n + \frac{1}{2} \sum_{k: (\lambda_k)_j^n > 0} \left[ 1 - \frac{\Delta t}{\Delta x} (\lambda_k)_j^n \right] (\alpha_k)_j^n (\mathbf{r}_k)_j^n + \frac{1}{2} \Delta t \mathbf{s}_j^n. \quad (25)$$

This is the final expression for the left state at  $x_{j+1/2}$  and  $t^{n+1/2}$ . We use this state as the left-hand initial condition in a Riemann problem at the interface at  $x_{j+1/2}$ . We can likewise use data at  $x_j$  to construct a “right state” to use as the right-hand (+) initial condition in a Riemann problem at the interface at  $x_{j-1/2}$ . This completes the characteristic tracing carried out to establish the initial conditions for the Riemann problem at  $t^{n+1/2}$  at each cell interface.

#### 2.2.4. Riemann solution

Given the left and right states at  $t^{n+1/2}$ , the predictor portion of the overall method is completed by solving the Riemann problem at each cell interface. This yields the non-gravitational quantities at  $t^{n+1/2}$  necessary to carry out the update of  $\mathbf{Q}_j^n$  to  $\mathbf{Q}_j^{n+1}$ . For an approximate Riemann solver, we use a simplified version of the method described by Colella and Glaz [12] that is constructed in the spirit of Bell et al. [3]. No modifications to this solver were required to account for self-gravity, which we already incorporated in a predictor fashion in the characteristic tracing and which we apply in a corrector fashion after solution of the Riemann problem.

#### 2.2.5. Artificial flux

In general, a flux vector  $\mathbf{F}_{j+1/2}^{n+1/2}$  found by solution of a Riemann problem can not be used in the update of  $\mathbf{Q}^n$  without modification. Spurious modes of oscillation that slope flattening alone can not remove are inherent to higher-order Godunov methods such as ours and can substantially degrade solution accuracy. To counter these modes, we apply artificial flux according to the form given by Colella and Woodward [13]. In their formulation, prior to the update of  $\mathbf{Q}^n$ , an artificial flux vector  $\mathbf{F}_{\text{artificial}}$  is added to the flux vector  $\mathbf{F}$  found from the Riemann solution. Explicitly,

$$\mathbf{F}_{\text{artificial}}^{n+1/2} = v \min[0, (\mathbf{D} \cdot \mathbf{v})_{j+1/2}^n] (\mathbf{Q}_{j+1}^n - \mathbf{Q}_j^n) \Delta x. \quad (26)$$

As the equation indicates, artificial flux is only introduced in regions of convergent flow. The term  $\mathbf{D} \cdot \mathbf{v}$  is a finite-difference approximation to the 3-D divergence  $\nabla \cdot \mathbf{v}$  and is defined at grid nodes (cell corners).

The parameter  $v$  is a dimensionless constant that we have taken to be  $v = 0.1$  for all calculations, following the suggestion of Colella and Woodward [13]. Note, however, that the dimensionless coefficient used here is an order of magnitude or more smaller than that used in non-Godunov schemes. In this sense, our code uses much less artificial viscosity than other common hyperbolic solvers.

Over the course of computing  $\mathbf{Q}^{n+1} = L_x L_y L_z (\mathbf{Q}^n)$ , we use the same term  $(\mathbf{D} \cdot \mathbf{v})^n$  in calculating the artificial flux vector during each 1-D update. It is evaluated once at step  $n$  at the beginning of the sequence. After  $\mathbf{Q}^{n+1}$  is obtained, then  $(\mathbf{D} \cdot \mathbf{v})^{n+1}$  is found for use in computing  $\mathbf{Q}^{n+2} = L_z L_y L_x (\mathbf{Q}^{n+1})$ .



### 2.2.6. Update step

At this point in the algorithm we have flux vectors  $\mathbf{F}_{j\pm 1/2}^{n+1/2}$  from solutions of the Riemann problems at cell interfaces, suitably modified by stabilizing contributions  $\mathbf{F}_{\text{artificial } j+1/2}^{n+1/2}$ . With these we can use Eq. (12) to construct a partially updated state vector  $\mathbf{Q}_j^{n+1,*}$  lacking only the contribution of  $\mathbf{S}_j^{n+1/2}$ . We complete the update from  $\mathbf{Q}_j^{n+1,*}$  to  $\mathbf{Q}_j^{n+1}$  by applying gravitational effects in a corrector fashion.

Note that the first (density) component of  $\mathbf{S}$  is always zero, as there are no sources or sinks of mass in the system. The fortuitous result is that the final value of  $\rho_j^{n+1}$  was obtained in what for the other state vector components was only a partial update step, i.e.,  $\rho_j^{n+1,*} = \rho_j^{n+1}$ . We use this density in the elliptic solver for Poisson's equation for the gravitational potential,

$$(\nabla^2 \phi)_j^{n+1} = 4\pi G \rho_j^{n+1} \quad (27)$$

to generate a solution for  $(\nabla \phi)_j^{n+1}$ . In the Poisson solver,  $\phi$  is defined at cell corners so that  $\nabla \phi$  is naturally defined at cell centers. We note that this is the only point during the course of the timestep that  $\phi$  is calculated. As mentioned earlier, we save this  $\nabla \phi$  for use in the source vector  $\mathbf{s}$  used in the next characteristic tracing. Explicitly, in the next step from  $t^{n+1}$  to  $t^{n+2}$ , we employ  $(\nabla \phi)_j^{n+1/2}$  to estimate  $\mathbf{q}_{j\pm 1/2,\mp}^{n+3/2}$  from  $\mathbf{q}_j^{n+1}$ .

All that remains is to adjust the momentum and total nongravitational energy components of  $\mathbf{Q}_j^{n+1,*}$ . The momentum density is straightforwardly updated via

$$(\rho \mathbf{v})_j^{n+1} = (\rho \mathbf{v})_j^{n+1,*} - \Delta t \frac{1}{2} [\rho_j^{n+1} \nabla \phi_j^{n+1} + \rho_j^n \nabla \phi_j^n] \quad (28)$$

to account for gravitational acceleration over the course of the interval  $\Delta t$ . In updating the total nongravitational energy density, we take advantage of the fact that the correction for self-gravity only directly affects the kinetic energy and must leave the internal energy unchanged:  $(\rho e)_j^{n+1} = (\rho e)_j^{n+1,*}$ . We can thus calculate the fully updated internal energy density

$$(\rho e)_j^{n+1} = (\rho e)_j^{n+1,*} - \frac{1}{2} \frac{1}{\rho_j^{n+1}} [(\rho \mathbf{v})_j^{n+1,*}]^2 \quad (29)$$

and then use it and the known fully updated momentum density to compute the fully updated total nongravitational energy density

$$(\rho E)_j^{n+1} = (\rho e)_j^{n+1} + \frac{1}{2} \frac{1}{\rho_j^{n+1}} [(\rho \mathbf{v})_j^{n+1}]^2. \quad (30)$$

This completes the update operation of the state vector from  $\mathbf{Q}_j^n$  to  $\mathbf{Q}_j^{n+1}$ .

### 2.2.7. AMR considerations

When solving a hyperbolic system on a hierarchy of computational grids, care must be taken that the solutions on finer grids are reflected on the neighboring and underlying coarser grids. Our code uses *refluxing* and *averaging down* procedures described in detail by Berger and Colella [4] to update the solutions on these coarser grids to account for solutions on the finer grids. In refluxing, the solution in a coarse cell adjacent to a fine grid is updated by adding the effects of a differential flux acting over  $\Delta t$  at its face adjacent to the refined region. This differential flux is equal to the difference between the flux at this face as computed on the coarse grid and the sum of the fluxes at this face as computed on the fine grid. In other words, fluxes taken from the fine grid are consistently incorporated into the solution in the neighboring cells of the coarse grid. The refluxing procedure

is implemented as a correction pass over coarse grid cells adjacent to fine grids after the fine grids have been advanced in time. The solution in a refined coarse cell is simply overwritten by averaging down the solutions on the fine cells it contains.

Consider the solution on a fine grid embedded in a coarser grid. In a nonself-gravitating system, the solution on the fine grid can only affect the solution on the coarse grid through the fluxes across the surface of the fine grid, and no signal can propagate faster than the sum of the flow speed and the sound speed. The result is that the coarser grid can be evolved on a commensurately coarser timestep than the fine grid, with each grid at each level evolving at the same Courant number. The procedures described above keep solutions on grids of different resolution in communication with one another and synchronized. However, in the case of a self-gravitating system, a time-dependent mass distribution on a fine grid has an immediate impact on the gravitational potential at all points on the coarser grid. The result is that the solutions on all grids are tightly coupled. The coarser grid could not evolve with a coarser timestep without application of a procedure more complicated than simple refluxing and averaging down. To avoid the need for such a procedure, we choose to evolve each grid in the computational volume using the same timestep, the smallest timestep in the volume. We find that the majority of cells are generally at the finest level of resolution, and it is this level that generally requires the finest timestep.

### 2.3. Elliptic solver

In describing the hyperbolic solver, we discussed the algorithm for a single grid and then presented the straightforward extensions necessary to run it in an adaptive mesh refinement framework. Each grid is advanced once per timestep, with only minor corrections due to other grids. In contrast, the solution of Poisson's equation for the gravitational potential is a much more tightly coupled process. The entire grid hierarchy is traversed repeatedly by our elliptic solver, with frequent communication between grids, as the algorithm iterates towards global convergence.

The multigrid method is a natural choice to use as an element of the elliptic solver in an AMR calculation. In this section we present an overview of some of the elements that distinguish our AMR multigrid method from a standard multigrid method. Textbook discussions of the latter can be found in [38,14].

#### 2.3.1. Stencils

Our algorithm is an iterative scheme for solving Poisson's equation simultaneously over all grids of the AMR hierarchy. We write this equation in finite difference form as

$$(\nabla^2 \phi)_{i+1/2,j+1/2,k+1/2} = 4\pi G \rho_{i+1/2,j+1/2,k+1/2}. \quad (31)$$

The elliptic solver takes  $\phi$ ,  $\nabla^2 \phi$  and  $\rho$  to be associated with nodes (cell corners), as opposed to the hyperbolic solver, which associated  $\mathbf{Q}$  with cells (cell centers) because this is the easiest implementation. For the solution  $\phi$  this is not a problem, since we seek  $\nabla \phi$  defined on cells for use in the hyperbolic solver in the terms  $\mathbf{S}$  and  $\mathbf{s}$ . We obtain  $\nabla \phi$  on cells from  $\phi$  on nodes via the formula

$$(\nabla \phi)_{i,j,k} = \frac{1}{4\Delta x} \begin{bmatrix} \phi_{+-} + \phi_{++} + \phi_{+-} + \phi_{++} - \phi_{--} - \phi_{--} - \phi_{--} - \phi_{--} \\ \phi_{-+} + \phi_{-+} + \phi_{-+} + \phi_{-+} - \phi_{--} - \phi_{--} - \phi_{--} - \phi_{--} \\ \phi_{-+} + \phi_{-+} + \phi_{-+} + \phi_{-+} - \phi_{--} - \phi_{--} - \phi_{--} - \phi_{--} \end{bmatrix}, \quad (32)$$

where  $\phi_{+-} = \phi_{i+1/2, j-1/2, k-1/2}$ ,  $\phi_{++} = \phi_{i+1/2, j-1/2, k+1/2}$ , etc. Note that this formula does not require any modification for use in an adaptive framework, since it does not use information beyond the edge of a given grid. For  $\rho$ , however, we face the difficulty of converting the cell-based data used by the hyperbolic solver into the node-based form used by the elliptic solver. On nodes inside a given grid this conversion is a simple averaging operation from the 8 cells surrounding each node, but at coarse-fine interfaces there are nodes that border both coarse and fine cells, and the averaging operation becomes more complicated and special procedures must be implemented in which values of the density must be obtained by weighted averages of the surrounding cells.

By averaging  $\rho$  on cells to yield  $\rho$  on nodes, data for the source term  $4\pi G\rho$  of Poisson's equation is formed. Similar issues are encountered in defining the Laplacian term  $\nabla^2\phi$  at interfaces. Node-based values of  $\phi$  are differenced to yield a node-based finite-difference approximation to the Laplacian term, but complexities arise in determining which values of  $\phi$  to difference. These procedures are discussed in detail in [37].

During application of the multigrid method, it is necessary to carry out a *restriction* operation in which data on fine grids is adapted for use on coarser grids. This is the final multigrid operation that requires special treatment at coarse-fine boundaries. Quantities at nodes are averaged in a weighted fashion to generate a value at the central node. This is illustrated in 2-D in [37] with generalization to 3-D straightforward.

### 2.3.2. Solver

Specifying the stencils at all points in the domain defines the linear finite-difference system representing Poisson's equation, and now we turn to the method of solving this system.

The standard multigrid method solves a linear system on a given grid by constructing and employing a set of underlying coarser grids. The coarser grids each have the same spatial extent as the finest grid. Like the standard multigrid method, our AMR multigrid method uses a set of underlying coarser grids. Unlike the standard method, however, these do not necessarily all have the same spatial extent. This is because the AMR method uses the AMR grid hierarchy as a portion of the multigrid hierarchy. Note, however, that the AMR hierarchy is augmented in two ways. First, while the levels of resolution in the AMR hierarchy typically differ by a factor of 4 in linear resolution, our multigrid implementation requires levels separated by a factor of 2 in linear resolution. The elliptic solver constructs new grids at levels of resolution between the AMR levels, as needed, to meet this requirement. These new levels are used only by the elliptic solver during execution of the multigrid algorithm. They are not involved in any other portion of the code. Second, the AMR structure is augmented at the bottom by introduction of grids coarser than the base AMR level. This is necessary since the multigrid method is applied to each AMR level.

Fig. 1a illustrates the relationships between grids at different levels of resolution. The diagram shows a standard multigrid V-cycle (cf. [38]) for a uniform mesh. In this figure, the vertical direction indicates resolution level, with finer levels on top. The horizontal direction indicates position in the multigrid cycle, and the dot denotes the resolution level at which the method operates at a given point in the sequence. The dashed line denotes the level on which the final solution is sought. Levels of coarser resolution are utilized in the sequence shown. Fig. 1b illustrates an example computational cycle of our AMR multigrid method. In this example, there are three AMR levels on which we seek the solution for  $\phi$  (denoted  $\ell = 0, 1, 2$ ), with factors of 4 in linear refinement separating them. The

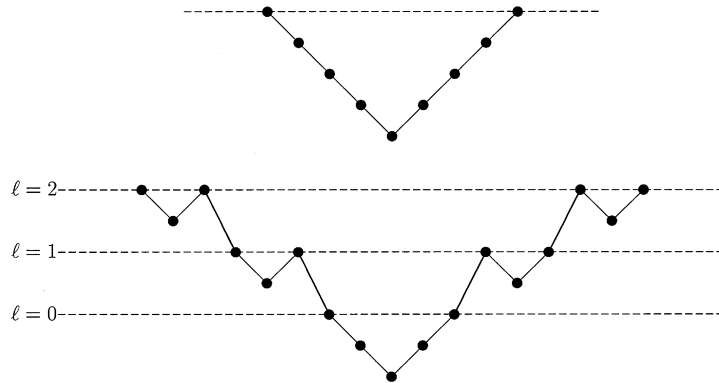


Fig. 1. Multigrid cycles. In these schematics, the vertical direction indicates level of resolution, with finer levels on top. The horizontal direction indicates position in the multigrid cycle, and the dot denotes the level at which the method operates at a given point in the sequence. (a) Standard multigrid V-cycle using four coarsenings to find the solution on the top (dashed) level. All levels occupy the same volume and communication between them is relatively straightforward. (b) AMR multigrid cycle for a case in which there are three AMR levels (dashed lines,  $\ell = 0, 1, 2$ ) separated by factors of 4 in linear refinement. Special temporary levels are inserted between and below the AMR levels so that the multigrid solver operates on grids differing by no more than a factor of 2 in linear refinement.

elliptic solver constructs and utilizes special temporary levels situated between and below the AMR levels. The standard multigrid V-cycle is modified with intermittent reversals as illustrated in order to produce the solution at each AMR level.

As mentioned in Section 2.1, our code employs periodic boundary conditions. In effect, the computational volume comprises an element of an infinite 3-D array of identical computational volumes, each of which contains an identical self-gravitating cloud. The gravitational field at any point in the computational volume is the sum of fields from the infinite number of “image” clouds. Because of this, it is necessary to ensure that the image clouds are separated by a distance large enough that their contributions to the field inside the “real” cloud are negligible. We typically employ a minimal center-to-center image cloud separation of  $d = 4R$ , where  $R$  is the initial cloud radius. In considering the uniform collapse problem we have shown that the effect of increasing the separation to  $d = 8R$  is merely to reduce the pull of the image clouds and hasten the collapse by of order 1%.

#### 2.4. AMR framework and the refinement criterion

The numerical algorithm used in this study was a high-resolution conservative finite-difference method for solving the compressible Euler equations. The basic finite difference method was a higher-order extension of Godunov’s method of a type discussed in [13]. This algorithm is second-order accurate for smooth flow problems, and has a robust and accurate treatment of discontinuities. It has been used quite extensively to compute unsteady shock reflections in gases, and has a demonstrated ability to resolve complex interactions of discontinuities found in astrophysical flows [22].

The supplementary technique we employ to further enhance the efficiency and resolution of our calculations are local adaptive mesh refinement. The adaptive mesh refinement algorithm (AMR) [5] is a dynamic regridding strategy based on an underlying rectangular discretization of the spatial

domain. During the course of the calculation, some pointwise measure of the error is computed at frequent intervals — typically every other time step. At those times, the cells that are identified are covered by a relatively small number of rectangular patches, which are refined by some even integer factor. Refinement is in both time and space, so that the calculation on the refined grids is computed at the same CFL number as that on the coarse grid. This procedure is applied recursively, i.e. the error on the refined grid monitored, and the regions with large errors covered by refined rectangular patches. The overall algorithm is fully conservative: the finite difference approximations on each level are in conservation form, as is the coupling at the interface between grids at different levels of refinement.

The version of the algorithm used for the present work is rather elaborate; a detailed description is given in [4]. However, there are several important features to this algorithm we wish to point out. The AMR uses a nested sequence of logically rectangular meshes to solve a PDE. In this work, we assume the domain is a single rectangular grid although it may be decomposed into several coarse grids. It is required that the discretized solution be independent of the particular decomposition of the domain into subgrids. Grids must be properly nested such that a fine grid should be at least one cell away from the boundary of the next coarser grid unless it is touching the boundary of the physical domain. However, a fine grid can cross a coarser grid boundary and still be properly nested. In this case, the fine grid has more than one parent grid.

AMR contains five separate components. The error estimation is used to estimate local truncation error and will be described subsequently. This determines where the solution is sufficiently accurate. The grid generator creates fine grid patches which cover the regions that need refinement. Data structure routines manage the grid hierarchy allowing access to the individual patches. Interpolation routines initialize a solution on a newly created fine grid and also provide the boundary conditions for integrating the fine grids. Flux correction routines ensure conservation at grid interfaces by modifying the coarse grid solution for coarse cells that are adjacent to a fine grid.

When all these components are assembled, a typical integration step proceeds as follows. The integration steps on different grids are interleaved so that before advancing a grid all the finer level grids have been integrated to the same time. One coarse grid cycle is then the basic unit of the algorithm. The mesh refinement factor in both space and time has been chosen most efficiently to be 4, although any even integer is possible. In practice, we use as many levels of refinement above the base coarse grid level as is required by the physics of the calculation. The regridding procedure is done every few time steps. The updating of the data on the locally refined grid structure is organized around the grouping of cells into rectangular grid patches, each one of which typically containing several hundred to several thousand grid cells. For example, the AMR code passes to a subroutine a rectangular grid of dependent variables and precomputed values in a set of ghost cells surrounding the grid, and assumes that the subroutine updates the values in the rectangular grid by one time step, as well as passing back the fluxes at cell edges that had been used in the update. The overheads in both CPU and memory associated with the adaptive mesh structure have been kept quite small, relative to other irregular grid schemes. Typically, 80–90% of the total execution time is spent advancing cells in time using the finite difference code, while the memory required is that needed to store two copies of the solution on all of the grids. These overheads are low because they are determined by the number of rectangles into which the AMR solution has been divided; as opposed to being determined by the number of grid cells, as is the case with the irregular grid adaptive algorithms.

In AMR, the computational volume consists of a hierarchical grid structure. A base Level 0 grid fills the computational volume, discretizing it on a rectangular grid with a resolution of  $\Delta x_0$  in each direction. Multiple Level 1 grids of finer resolution  $\Delta x_1 = \Delta x_0 / r_1$  may be embedded within it, where  $r_1 = 4$  is a typical choice. In turn, multiple Level 2 grids of resolution  $\Delta x_2 = \Delta x_1 / r_2$  may be embedded within Level 1, and so on. Grids at Level  $L$  always span an integral number of cells at Level  $L - 1$ , i.e., partial cell refinement is not permitted. Furthermore, a grid at Level  $L$  is always nested within a grid at Level  $L - 1$  such that there is a buffer region of Level  $L - 1$  cells surrounding it. In other words, a grid at Level  $L$  within a grid at Level  $L - 1$  never shares a boundary with the Level  $L - 1$  grid.

The computational volume is discretized on a base Level 0 grid with a full edge length equal to  $d$ . A typical calculation begins with this grid at  $R_8$  resolution (throughout this paper we denote a resolution of  $\Delta x = R/N$  by the shorthand  $R_N$ , as in [22], where  $R$  is the initial cloud radius). As mentioned in Section 2.3.2, we regularly utilize  $d = 4R$  to reduce image-cloud effects, and thus the Level 0 grid typically consists of  $32^3$  cells. We now discuss the routines by which the refinement decision is made and carried out within this grid.

A key component of an AMR code is the procedure by which the decision is made whether or not a given portion of the flow is adequately resolved. In our code, this procedure is broken into two steps. In the first step, a specified property is measured in each cell, and the cell is flagged for refinement if a specified algorithm indicates the measurement requires it. In the second step, the distribution of cells requiring refinement is analyzed to determine the number, sizes, and locations of grids to be inserted at the next finer level of resolution. These finer grids will always include every cell that was flagged for refinement, but they may also include additional cells that were not flagged. The degree to which the refinement is concentrated in the cells that require it is termed the *grid efficiency*. The grid efficiency is minimal when the smallest rectangular solid containing all flagged cells is refined. In this case the fraction of refined volume actually containing cells that required refinement may be very small. The grid efficiency is maximal when the only cells refined are those that were flagged.

In our calculations in this paper and in [36], the criterion for refinement to Level 1, typically of  $R_{32}$  resolution, is that a Level 0 ( $R_8$  resolution) cell contain gas identified as belonging to the cloud. In practice, we define the “cloud” as gas of density greater than or equal to half the initial value of the density at the surface of the cloud,  $\rho_{\text{cloud}}(R)$ . This maintains  $R_{32}$  resolution of surface gas that undergoes a modest expansion. Because the refinement criteria are evaluated before the first timestep is made, the cloud is refined to Level 1 ( $R_{32}$  in this example) from the beginning of the calculation. This usage of Level 1 as the initial effective base level of resolution is entirely motivated by the necessity to use the actual base Level 0 to remove the boundaries of the computational volume from the surface of the cloud. The cost incurred in employing a  $32^3$  Level 0 to accomplish this is negligible given that a typical calculation involves more than  $10^6$  cells. The criterion for refinement to the second and higher levels of resolution during the course of the simulation is more sophisticated, and we discuss it next.

We will begin by discussing our general approach to error estimation for fluid dynamic problems and then discuss the important refinement criteria (numerical Jeans condition) that we employ for self-gravitational hydrodynamics. Our process for deciding upon a suitable level of refinement depends on a reasonable method for estimating the local truncation error of the Godunov scheme; this error estimator determines where the solution accuracy is insufficient. We estimate this truncation

error by taking data on the grid where the error is being estimated and taking two timesteps. This solution is then combined with the data on the grid spatially coarsened by a factor of two and integrated for one timestep. The local truncation error can be shown to be proportional to the functional that is the difference of these two operations. We illustrate this type of Richardson extrapolation with a simple example.

Consider for simplicity a first-order hyperbolic equation in one spatial dimension to illustrate the method of error estimation,

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}. \quad (33)$$

We consider the finite difference replacement of Eq. (33) over a rectangular grid with  $h$  and  $k$  being the grid spacing in the  $x$  and  $t$  directions and grid point  $(X, T)$  is given by  $X = mh$ ,  $T = nk$  where  $m$  and  $n$  are integers. The functions satisfying the difference and differential equations at the grid points  $(X, T)$  are denoted by  $U_m^n$  and  $u_m^n$ , respectively. A second-order accurate finite-difference representation of Eq. (33) is the Lax–Wendroff scheme [29].

$$U_m^{n+1} = U_m^n + (r/2)(U_{m+1}^n - U_{m-1}^n) + (r^2/2)(U_{m+1}^n - 2U_m^n + U_{m-1}^n), \quad (34)$$

where the grid ratio  $r = k/h$ .

We can express the solution at  $mh, (n+1)k$  as the result of a linear operator  $L^{(1)}$  acting on the data at  $mh, nk$  as

$$U_m^{n+1} = L^{(1)} U_m^n. \quad (35)$$

By using Taylor expansions around  $U_m^n$  in space in Eq. (34) we can derive the operator  $L^{(1)}$ .

$$\begin{aligned} U_m^{n+1} = & (1 - r^2)U_m^n + \left(\frac{r + r^2}{2}\right) \left[ U_m^n + h \frac{\partial U}{\partial x} + (h^2/2) \frac{\partial^2 U}{\partial x^2} + (h^3/6) \frac{\partial^3 U}{\partial x^3} \right] \\ & + \left(\frac{-r + r^2}{2}\right) \left[ U_m^n - h \frac{\partial U}{\partial x} + (h^2/2) \frac{\partial^2 U}{\partial x^2} - (h^3/6) \frac{\partial^3 U}{\partial x^3} + \dots \right]. \end{aligned} \quad (36)$$

Collecting terms,

$$\begin{aligned} U_m^{n+1} = & \left[ 1 + rh \frac{\partial}{\partial x} + (r^2 h^2/2) \frac{\partial^2}{\partial x^2} + (rh^3/6) \frac{\partial^3}{\partial x^3} \right] U_m^n, \\ L^{(1)} \equiv & 1 + rh \frac{\partial}{\partial x} + (r^2 h^2/2) \frac{\partial^2}{\partial x^2} + (rh^3/6) \frac{\partial^3}{\partial x^3} + O(h^4). \end{aligned} \quad (37)$$

We introduce the local truncation error  $Z_m^{n+1}$  as the difference between the exact solution of the differential equation  $u_m^{n+1}$  and the difference equation  $U_m^{n+1}$ , i.e.,  $Z_m^{n+1} = u_m^{n+1} - U_m^{n+1}$ . The local truncation error is essentially the error in evolving the exact continuum solution  $u_m^n$  from  $t = nk$  to  $t = (n+1)k$  using the difference operator, Eq. (37),

$$Z_m^{n+1} = u_m^{n+1} - L^{(1)} u_m^n. \quad (38)$$

To compute  $Z_m^{n+1}$  we use Taylor expansions in time and Eq. (37),

$$\begin{aligned} Z_m^{n+1} &= u_M^N + k \frac{\partial u}{\partial t} + (k^2/2) \frac{\partial^2 u}{\partial t^2} + (k^3/6) \frac{\partial^3 u}{\partial t^3} + O(k^4) \\ &\quad - u_m^n - rh \frac{\partial u}{\partial x} - (r^2 h^2/2) \frac{\partial^2 u}{\partial x^2} - (rh^3/6) \frac{\partial^3 u}{\partial x^3} + O(h^4) \\ &= k \left( \frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} \right) + (k^2/2) \left( \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} \right) + (h^3/6) \left( r^3 \frac{\partial^3 u}{\partial t^3} - r \frac{\partial^3 u}{\partial x^3} \right). \end{aligned} \quad (39)$$

If we assume that the derivatives of  $u$  with respect to  $x$  and  $t$  are continuous and bounded, then we can use the chain rule with Eq. (33) to determine the truncation error

$$Z_m^{n+1} = (r/6)(r^2 - 1)h^3 \frac{\partial^3 u}{\partial x^3} \quad (40)$$

which is third order in the grid spacing  $h$ .

Similarly, we can consider the local truncation error in evolving the solution two consecutive timesteps on the original grid,

$$\begin{aligned} Z_m^{n+2} &= u_m^{n+2} - L^{(1)} u_m^{n+1} \\ &= u_m^{n+2} - L^{(1)} L^{(1)} u_m^n. \end{aligned} \quad (41)$$

To  $O(h^4)$  we find

$$L^{(1)} L^{(1)} = 1 + 2rh \frac{\partial}{\partial x} + 2r^2 h^2 \frac{\partial^2}{\partial x^2} + \left( \frac{r}{3} + r^3 \right) h^3 \frac{\partial^3}{\partial x^3}. \quad (42)$$

We likewise compute  $Z_m^{n+2}$  by Taylor expanding in time and using the operator in Eq. (42) to obtain

$$\begin{aligned} Z_m^{n+2} &= u_m^n + 2k \frac{\partial u}{\partial t} + 2k^2 \frac{\partial^2 u}{\partial t^2} + (4/3)k^3 \frac{\partial^3 u}{\partial t^3} + \cdots \\ &\quad - u_m^n - 2rh \frac{\partial u}{\partial x} - 2r^2 h^2 \frac{\partial^2 u}{\partial x^2} - (rh^3/3) \frac{\partial^3 u}{\partial x^3} - r^3 h^3 \frac{\partial^3 u}{\partial x^3}, \end{aligned} \quad (43)$$

whereupon we again use the chain rule with Eq. (33) to obtain,

$$Z_m^{n+2} = (1/3)r(r^2 - 1)h^3 \frac{\partial^3 u}{\partial x^3}. \quad (44)$$

We now consider the local truncation error  $\bar{Z}_m^{n+2}$  in evolving the exact solution from  $t = nk$  to  $(n+2)$  in a single timestep on a grid coarsened by a factor of 2 such that the grid spacing is  $2h$ .

$$\bar{Z}_m^{n+2} = u_m^{n+2} - L^{(2)} u_m^n, \quad (45)$$



where  $L^{(2)}$  is the operator  $L^{(1)}$  on a coarsened grid.

$$\begin{aligned}\bar{Z}_m^{n+2} &= u_m^n + 2k \frac{\partial u}{\partial t} + 2k^2 \frac{\partial^2 u}{\partial t^2} + (4k^3/3) \frac{\partial^3 u}{\partial t^3} \\ &\quad - u_m^n - 2rk \frac{\partial u}{\partial x} - 2r^2 h^2 \frac{\partial^2 u}{\partial x^2} - (4rh^3/3) \frac{\partial^3 u}{\partial x^3} \\ &= 2rh \frac{\partial u}{\partial x} + 2r^2 h^2 \frac{\partial^2 u}{\partial x^2} + (4r^3 h^3/3) \frac{\partial^3 u}{\partial x^3} \\ &\quad - 2rk \frac{\partial u}{\partial x} - 2r^2 h^2 \frac{\partial^2 u}{\partial x^2} - (4rh^3/3) \frac{\partial^3 u}{\partial x^3}.\end{aligned}\quad (46)$$

Upon cancellation of terms in Eq. (46) we obtain

$$\bar{Z}_m^{n+2} = (4rh^3/3)(r^2 - 1) \frac{\partial^3 u}{\partial x^3}.\quad (47)$$

Subtracting Eq. (44) from Eq. (47) and using Eqs. (41) and (45) we find

$$(1/6)(L^{(1)}L^{(1)}u_m^n - L^{(2)}u_m^n) = (rh^3/6)(r^2 - 1) \frac{\partial^3 u}{\partial x^3},\quad (48)$$

whereupon using Eq. (40) we find for the local truncation error

$$\begin{aligned}Z_m^{n+1} &= (1/6)(L^{(1)}L^{(1)} - L^2)u_m^n \\ &= u_m^{n+1} - L^{(1)}u_m^n.\end{aligned}\quad (49)$$

Thus, the local truncation error  $Z_m^{n+1}$  is formally equivalent to taking two timesteps on the original grid and combining this with one timestep on a factor of two coarsened grid. This Richardson's extrapolation forms the first step for our error estimation in the criteria for identifying grid cells for grid refinement. In practice, we may estimate the local truncation error by monitoring the gradients of the density or the internal energy if we are interested in following shock waves in the flow. The choice of the error tolerance allowed on the truncation error is one of experimentation and may vary from one problem to the next.

In the isothermal calculations presented in this paper and in [36], refinement to Level 2 and above is controlled by application of the Jeans condition. This criterion can be augmented or replaced by others as necessary. The Jeans condition used to make the cell-by-cell refinement decision is a physically motivated resolution constraint discussed by Truelove et al. [36]. We summarize the main points here. Jeans [17,18] analyzed the linearized equations of 1-D isothermal GHD for a medium of infinite extent and found that perturbations on scales larger than the Jeans length,

$$\lambda_J \equiv \left( \frac{\pi c_s^2}{G\rho} \right)^{1/2}\quad (50)$$

are unstable. Thermal pressure can not resist the self-gravity of a perturbation larger than  $\lambda_J$ , and runaway collapse results. Truelove et al. [36] showed that the errors generated by numerical GHD solvers can act as unstable perturbations to the flow. In a simulation with variable resolution, cell-scale errors introduced in regions of coarser resolution can be advected to regions of finer resolution, affording these errors the opportunity to grow. The unstable collapse of numerical perturbations can lead to

substantial fragments, a process termed artificial fragmentation. The strategy for avoidance of artificial fragmentation is to maintain sufficient resolution of  $\lambda_J$ . Defining the Jeans number

$$J \equiv \frac{\Delta x}{\lambda_J}, \quad (51)$$

Truelove et al. [36] found keeping  $J \leq 0.25$  avoided artificial fragmentation in isothermal evolution of a collapse spanning 7 decades of density, the approximate range separating typical molecular cloud cores from nonisothermal protostellar fragments. The constraint that  $\lambda_J$  be resolved is termed the Jeans condition. Although it has been shown to hold only for isothermal evolution, it is reasonable to expect that it is necessary (although not necessarily sufficient) for nonisothermal collapse as well.

As a side effect of confining cell-sized perturbations to a length scale at which they are thermally damped, resolution of  $\lambda_J$  also ensures that gradients developed in isothermal flow by gravity are well resolved. Formation of structure on scales of  $\lambda_J$  and larger is a general feature of isothermal GHD flow since smaller fluctuations are damped but larger ones collapse. For example, in the self-similar solution for isothermal cylindrical collapse [15], the radial scale height of the cylinder is  $\lambda_J$ . As another example, an isothermal self-gravitating atmosphere has a scale height of essentially  $\lambda_J$ . Lack of resolution of gradients within simulated flow triggers the injection of artificial viscosity, which is generally intended to be introduced in small amounts only for numerical stability (in our Godunov scheme) or shock mediation (in many common non-Godunov schemes). Introducing excess amounts of artificial viscosity renders the problem solved different from the inviscid problem posed. Continuous resolution of  $\lambda_J$ , however, keeps the flow inviscid and prevents artificial slowing of gravitational collapse. It is important to note, however, that the Jeans condition is a necessary but not, in general, sufficient condition to ensure convergence. Non-gravitational effects can generate structure on scales below  $\lambda_J$ . In particular, the transition to nonisothermal evolution may produce structure on smaller scales than the local Jeans length.

Our implementation of the Jeans condition as the refinement criterion is straightforward. In the first step of the refinement procedure, the  $J$  of each cell is calculated, and the cell is flagged for refinement if its  $J$  exceeds a specified  $J_{\max}$ , usually  $J_{\max} = 0.25$ . Fig. 2 illustrates how this proceeded for an isothermal simulation of the Gaussian cloud considered by Truelove et al. [36]. The initial resolution was  $R_{32}$ , the initial peak density was  $\rho = 10^{-16.8} \text{ g cm}^{-3}$ , and the factor of linear refinement between levels of resolution was 4.

The Jeans condition on  $\Delta x$  fundamentally differs from the Courant condition on  $\Delta t$ , although at first the two conditions might appear analogous. The Courant condition arises from a modal stability analysis of finite difference equations derived from the Euler partial differential equations (PDEs) of hydrodynamics (see, e.g., [32]). It is entirely a consequence of the finite differencing and has no physical counterpart in the PDEs. A modal stability analysis of finite difference equations derived from the GHD PDEs does not yield the Jeans condition, but rather a generalized Courant condition that includes effects of gravity (cf. [37]).

The Jeans condition arises because perturbations on scales above  $\lambda_J$  are physically unstable, and discretization of the GHD PDEs introduces perturbations on all scales above  $\Delta x$ . It is essential to keep the  $\lambda_J$  as resolved as possible in order to diminish the initial amplitude of perturbations that exceed this scale. Resolution of  $\lambda_J$  also suppresses the introduction of artificial viscosity and its accompanying effect of artificially slowing the collapse.

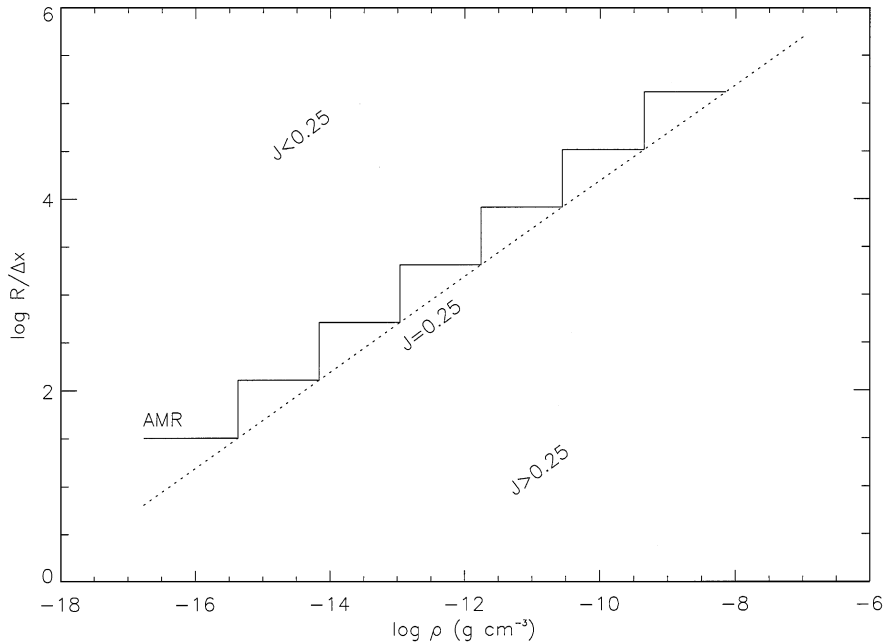


Fig. 2. An example of the dependence of resolution  $R/\Delta x$ , where  $R$  is the initial cloud radius, upon density when AMR maintains  $J_{\max} = 0.25$  and each additional level of resolution represents an additional factor of 4 in linear refinement. This particular case corresponds to calculations made by Truelove et al. [36].

### 2.5. Equation of state

Our AMR code is designed to accommodate a general equation of state. This functionality will prove useful when one follows the evolution of collapsing molecular clouds beyond the transition from isothermal to nonisothermal evolution around  $\rho \sim 10^{-12} \text{ g cm}^{-3}$ . To do so requires implementation of a model of radiative transfer (e.g., [31]), which could be expected to yield an equation of state similar to the one found by Winkler and Newman [39]. In our previous work [36,37], we focused on the evolution of molecular clouds in the density regime below this transition, where such clouds efficiently lose thermal energy through radiation and therefore remain isothermal. In recent work [25] we have used a nonisothermal EOS that employs a combined isothermal and nonisothermal component such that  $P(\rho) = \rho c_s^2 + K\rho^\gamma$  where  $\gamma = \frac{5}{3}$  and  $K$  is chosen so that the isothermal and adiabatic components balance at a critical density determined by detailed 1-D radiation-hydrodynamic calculations of Masunaga et al. [27]. A typical critical density for this transition is  $5 \times 10^{-14} \text{ g cm}^{-3}$ .

Isothermal gas behaves as though its constituent particles possess an infinite number of internal degrees of freedom, i.e.,  $\gamma = 1$ . We approximate isothermality but continue to use an adiabatic equation of state by setting  $\gamma = 1.001$  in the ideal gas law. Since  $P \propto \rho^\gamma$  during isentropic evolution, this leads to a spurious pressure increase of only 1% after a density increase of 6 orders of magnitude.

### 3. Applications to rotating molecular clouds

In this section we briefly describe recent high-resolution results obtained with AMR for the gravitational collapse of isothermal, rotating, uniformly dense clouds [37]; isothermal centrally condensed clouds [36] and nonisothermal collapse of uniformly dense clouds [25]. These calculations illustrate the unprecedented resolution possible with the AM methodology.

#### 3.1. Isothermal uniform clouds

The uniform cloud is an overly idealized model of astrophysical clouds but is very useful as a first approximation. Boss [6] counts among the virtues of the uniform cloud the availability of simple analytic forms for global quantities, the absence of structure parameters requiring numerical variation, and the presence of a simple backdrop against which to study growth of input perturbations.

Among the more useful global quantities for characterizing clouds are the energy ratios  $\alpha \equiv E_{\text{thermal}}/|E_{\text{gravitational}}|$  and  $\beta_{\Omega} \equiv E_{\text{rotational}}/|E_{\text{gravitational}}|$ .

Burkert and Bodenheimer [9] studied a uniform cloud whose properties were chosen to reproduce certain characteristics of the Gaussian cloud examined by Boss [7] and more recently studied by Truelove et al. [37]. Like the Gaussian cloud, this cloud has  $M = 1M_{\odot}$  and  $R = 5 \cdot 10^{16}$  cm, which in the uniform case yield  $\rho_0 = 10^{-17.4}$  g cm<sup>-3</sup>. Its energy ratios are also the same as for the Gaussian cloud:  $\alpha = 0.26$  and  $\beta_{\Omega} = 0.16$ . The isothermal uniform cloud thus has  $c_s = 0.167$  km s<sup>-1</sup> and a rotation rate  $\Omega = 7.2 \cdot 10^{-13}$  rad s<sup>-1</sup>. The cloud is perturbed by  $\rho \rightarrow \rho \times [1 + 0.1 \cos(2\phi)]$ , a 10%  $m = 2$  mode seed perturbation of the same amplitude as that used in the Gaussian problem. This cloud begins with  $\lambda_j = 1.17R$ . Truelove et al. [37] used an initial resolution of  $R_{32}$  and dynamically refined so as to ensure  $J_{\text{max}} = 0.25$ . In this problem the fiducial radius beyond which cloud gravity dominates perturbation pressure is  $r_{\text{crit}} = 0.14R$ , so that with  $R_{32}$  we still linearly resolve this scale with more than four cells.

They evolved the isothermal collapse over a density increase of more than 8 decades. At a time when the density has reached  $\rho_{\text{max}} = 10^{-12.4}$  g cm<sup>-3</sup>, near the end of the isothermal regime, the cloud has produced two local density maxima (a binary), each of which is proceeding to form an isothermal singularity. Each fragment has a mass  $M_f = 0.032M_{\odot}$ . The calculation was continued until after  $\rho$  has increased by 8.1 decades. At this point, the code used resolution at the level of  $R_{131072}$ , representing six levels of refinement beyond the original  $R_{32}$ ; the finest cells are only 5.4 solar radii in size. This resolution, only possible with AMR, is effectively equivalent to a fixed grid calculation of  $10^{15}$  uniform cells. We conclude that the solution to the isothermal, inviscid collapse problem posed is collapse to singular filaments that do not subfragment. This happens first in the binary fragments, each of which forms its own filament. The bar between the binary components tends to form a third filament. One could also regard the entire binary and bar structure as a single filament with a free-fall time that varies along its length, from shortest at the ends to longest at the center. Calculations that artificially slow the singular collapse at some artificially determined evolutionary point (as by using an arbitrarily determined finest resolution) expose a collapsing filament to the possibility of fragmentation from perturbations present in the simulation. Such is the case with the calculations of BB93, whose fixed grid resolution results in artificial fragmentation of the bar connecting the binary. The fragments they obtain are entirely resolution dependent. In contrast, the work of Truelove et al. [37] and Inutsuka and Miyama [16] support a scenario whereby: collapse

of a rotating cloud first forms a disk, and perturbations within the disk gas result in fragmentation forming linear filaments. These filaments collapse to singularities in the absence of an intervening thermal or viscous effect.

### 3.2. Isothermal centrally condensed clouds

Poorly understood fragmentation appears in several published solutions to two prominent problems: (1) the collapse of a Gaussian cloud with a 10%  $m=2$  perturbation studied by Boss [7] and (2) the related problem of the collapse of a perturbed uniform cloud first treated by Burkert and Bodenheimer [9] discussed previously. The discovery of artificial fragmentation by Truelove et al. [36] prompted them to reexamine these problems.

Consider the Gaussian cloud problem studied by Boss [7], the isothermal collapse of a  $1 M_{\odot}$  spherical cloud. The  $R = 5 \cdot 10^{16}$  cm cloud is initially set in solid-body rotation at  $\Omega = 10^{-12}$  rad s $^{-1}$  with a Gaussian radial density dependence,  $\rho(r) = \rho_c e^{-(r/R_1)^2}$ , where  $\rho_c = 10^{-16.8}$  g cm $^{-3}$  and  $R_1 = 0.58R$ . The soundspeed is  $c_s = 0.19$  km s $^{-1}$ . To stimulate fragmentation, Boss applied a “seed” perturbation:  $\rho(r) \rightarrow \rho(r) \times [1 + 0.1 \cos(2\phi)]$ , where  $\phi$  is the azimuthal angle. This problem been revisited many times [20,10] Klapp et al. [20] found the fragmentation seen in their calculations to be resolution-dependent. Burkert and Bodenheimer [9] treated the problem with their fixed multiple-grid code and used finer resolution than preceding workers. They made two runs in which the finest resolution was fixed at  $R_{512}$  and  $R_{2048}$ , respectively, with the latter differing from the former in use of finer resolution only in a limited region about the origin. From a similarity in gross fragment morphology between their runs, they concluded their results were numerically converged. The outcome of their simulations yielded multiple fragmentation. Truelove et al. [36] showed that these simulations were contaminated by the growth of numerical perturbations due to excessive violation of the Jeans condition on the grid. Lack of adequate resolution resulted in enhanced artificial viscosity and thus artificial fragmentation. This has the consequence of producing physically inappropriate viscosity in regions where other gradients become unresolved. In the case of collapse of a filamentary singularity, significant gradients exist on the scale of the cylindrical scale height, which is essentially  $\lambda_J$  [15]. When such a cylindrical collapse is evolved on a grid of fixed finest resolution,  $\lambda_J$  will eventually become poorly resolved, and artificial viscosity will appear. Growth of  $\rho_{\max}$  will then subside with  $J_{\max} \geq 1$ . They found that in Burkert and Bodenheimer [9]  $R_{512}$  calculation, artificial viscosity kept  $\rho_{\max} \leq 10^{-11.0}$  g cm $^{-3}$ , with  $J_{\max} = 2.39$ . In their  $R_{2048}$  calculation, it kept  $\rho_{\max} \leq 10^{-9.43}$  g cm $^{-3}$ , with  $J_{\max} = 3.65$ .

Truelove et al. [36] carried out a series of calculations of this problem using increasingly finer resolution in excess of that used by Burkert and Bodenheimer [9]. First, they performed runs that began with  $R_{32}$  and utilized AMR to maintain  $J \leq J_{\max}$ . They examined the results of runs at  $J_{\max} = 0.5$  and 0.25, respectively. The  $J_{\max} = 0.5$  case resulted in a binary, but the  $J_{\max} = 0.25$  case produced a single filament. To confirm that the single-filament structure represented the converged morphology, they performed an additional run in which they globally doubled the resolution over the  $R_{32}$  and  $J_{\max} = 0.25$  case by starting with  $R_{64}$  and utilizing  $J_{\max} = 0.125$ . They carried this highly resolved run to  $\rho_{\max} = 10^{-11.2}$  g cm $^{-3}$  and found only the single-filament structure. The  $R_{32}$  and  $J_{\max} = 0.5$  run had already displayed binary fragmentation by this  $\rho_{\max}$ . They concluded that the binary formation found using  $R_{32}$  and  $J_{\max} = 0.5$  is a resolution-dependent numerical artifact. In general, as shown by Inutsuka and Miyama [16], perturbed, unstable, inviscid, isothermal clouds form filaments that

increase in density faster than linear perturbations along them can grow to fragments. The results of Truelove et al. [36] confirm this showing that the outcome of the collapse of the Gaussian cloud is a singular isothermal filament.

### 3.3. Nonisothermal uniform clouds

In recent work using high-resolution AMR, Klein et al. [25] followed the collapse of an initially rigidly rotating, uniform isothermal cloud. They used a two-component barotropic equation of state which makes the transition from an isothermal equation in the optically thin regime, to a polytropic equation of state in the optically thick regime in a smooth fashion as described in Section 2.5. The initial conditions are identical to the isothermal uniform cloud previously discussed and an  $m=2$  perturbation is applied to the initial density. In the following, we discuss the evolution of this nonisothermal collapse. Although a similar calculation has been recently studied by Bate and Burkert [2], none of the calculations in the literature to date have been able to follow the subsequent evolution of the fragments over dynamical (orbital) timescales, while still adhering to the Jeans criterion. To our knowledge, this is the first calculation to do so. Fig. 3 represents a greyscale 2-D slice of log density through the equatorial plane  $t = 1.41 \cdot 10^{12}$  s after start of the collapse. The darkest shades represent the highest density. Velocity vectors indicating the direction and magnitude of the flow are superimposed. The cloud initially collapses to an isothermal disk and the formation of an elongated filamentary bar with the first signs of fragmentation in the bar. A strong isothermal shock above the plane of the disk is soon established. After  $t = 1.46 \cdot 10^{12}$  s (Fig. 4) the isothermal bar becomes optically thick and the accretion flow onto the bar is arrested resulting in the growth of nonaxisymmetric perturbations in the bar. Fragmentation in the bar results in the formation of binary spherical cores connected by a prominent bar. The core–bar system is embedded in a two armed spiral, derivative of a  $m = 2$  perturbation. The low angular momentum of shocked gas accreted in the bar allows the bar to be directly accreted onto the cores. The binary separation decreases as the cores increase their mass by accretion from the bar and the bar dissipates. Fig. 5 shows a snapshot of log density about 0.5 rotation periods later at  $t = 1.51 \cdot 10^{12}$  s. Protostellar disks have now formed around the cores with the cores at their closest orbital separation  $\sim 44$  AU. The disks accrete gas directly from the long spiral arms. The cores accrete by Roche Lobe overflow of the contacting protostellar disks and each core has  $0.08M_{\odot}$  with a radius  $\sim 10$  AU. These first collapse cores have central temperatures  $T_c \sim 400$  K and have reached densities of  $3 \cdot 10^{-11} \text{ g cm}^{-3}$ . The masses and radii of these first collapse cores are in good agreement with the detailed 1-D radiation-hydrodynamic calculations of Masunaga et al. [27]. The core accretion luminosity  $\sim 2 \cdot 10^{31} \text{ erg}^{-1}$  obtained in the 3-D collapse calculations is considerably less than that found in the 1-D simulations  $\sim 4 \cdot 10^{32} \text{ erg}^{-1}$  due to the large outward angular momentum transport in the 3-D simulation resulting in less matter accreting onto the protostellar cores. This effect is totally absent in 1-D simulations where all the matter flows radially inward.

It is important to point out that the initial rapid growth of the cores is due to accretion of matter from the bar. As the bar dissipates, the cores have ended their first phase of growth and collapse and proceed to grow more slowly by direct accretion from the surrounding protostellar disks. At  $t = 1.52 \cdot 10^{12}$  s (Fig. 6) the cores begin to separate and the surrounding disks become morphologically distinct. The protostellar disks and cores become a fully detached binary at  $t = 1.56 \cdot 10^{12}$  s (Fig. 7) with the disks attached to the long spiral. The scale of the disks, of order 100 AU, is consistent

Uniform cloud with 10%  $m=2$  perturbation.

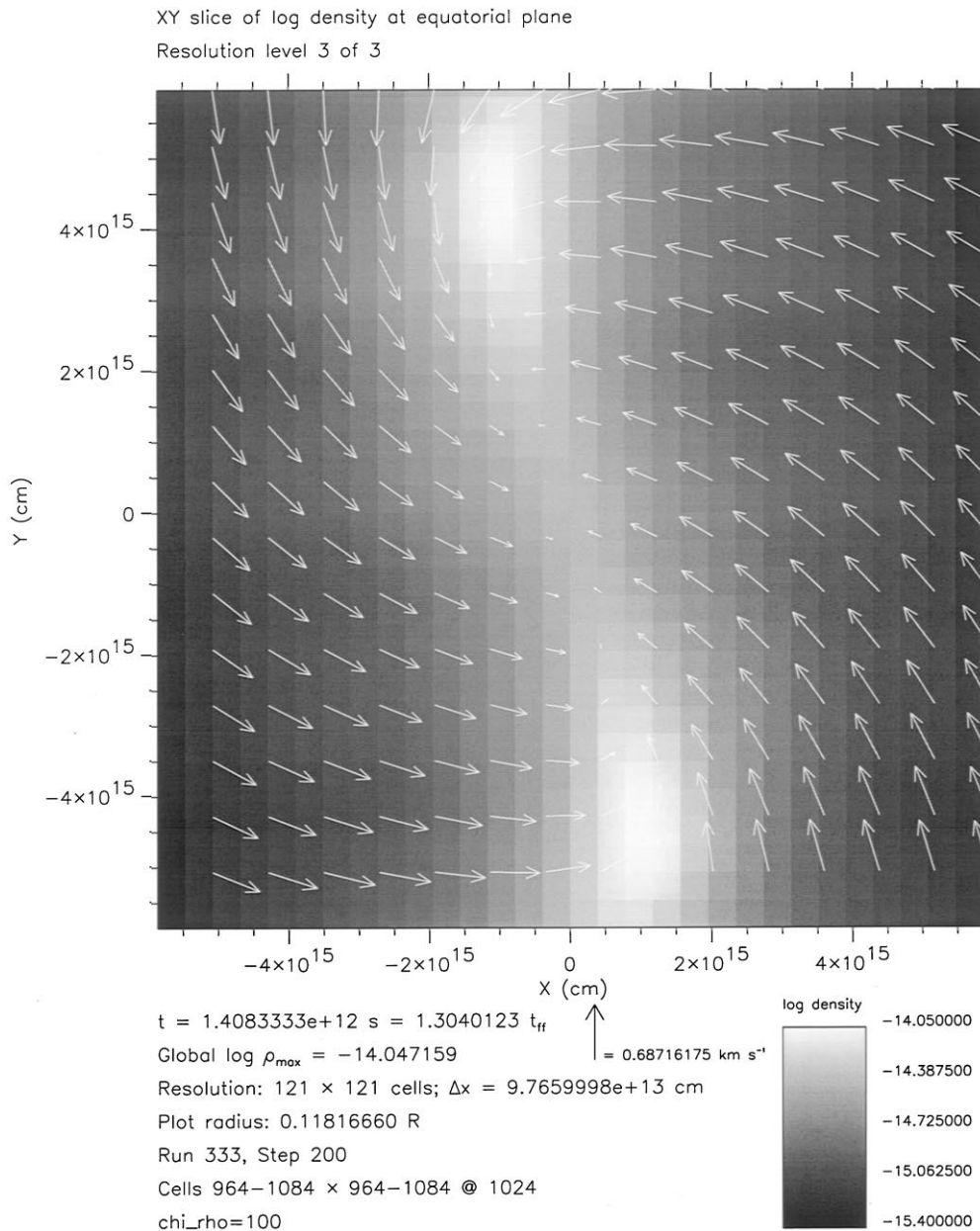


Fig. 3. Equatorial slice of log density of the uniform nonisothermal cloud subject to a 10%  $m=2$  azimuthal perturbation. This slice is taken at the time ( $t = 1.41 \cdot 10^{12} \text{ s}$ ) when the cloud has collapsed into an equatorial disk and fragmentation in the disk is evident as two elongated linear fragments. The velocity vectors indicate that the fragments are accreting predominantly from the surrounding disk.

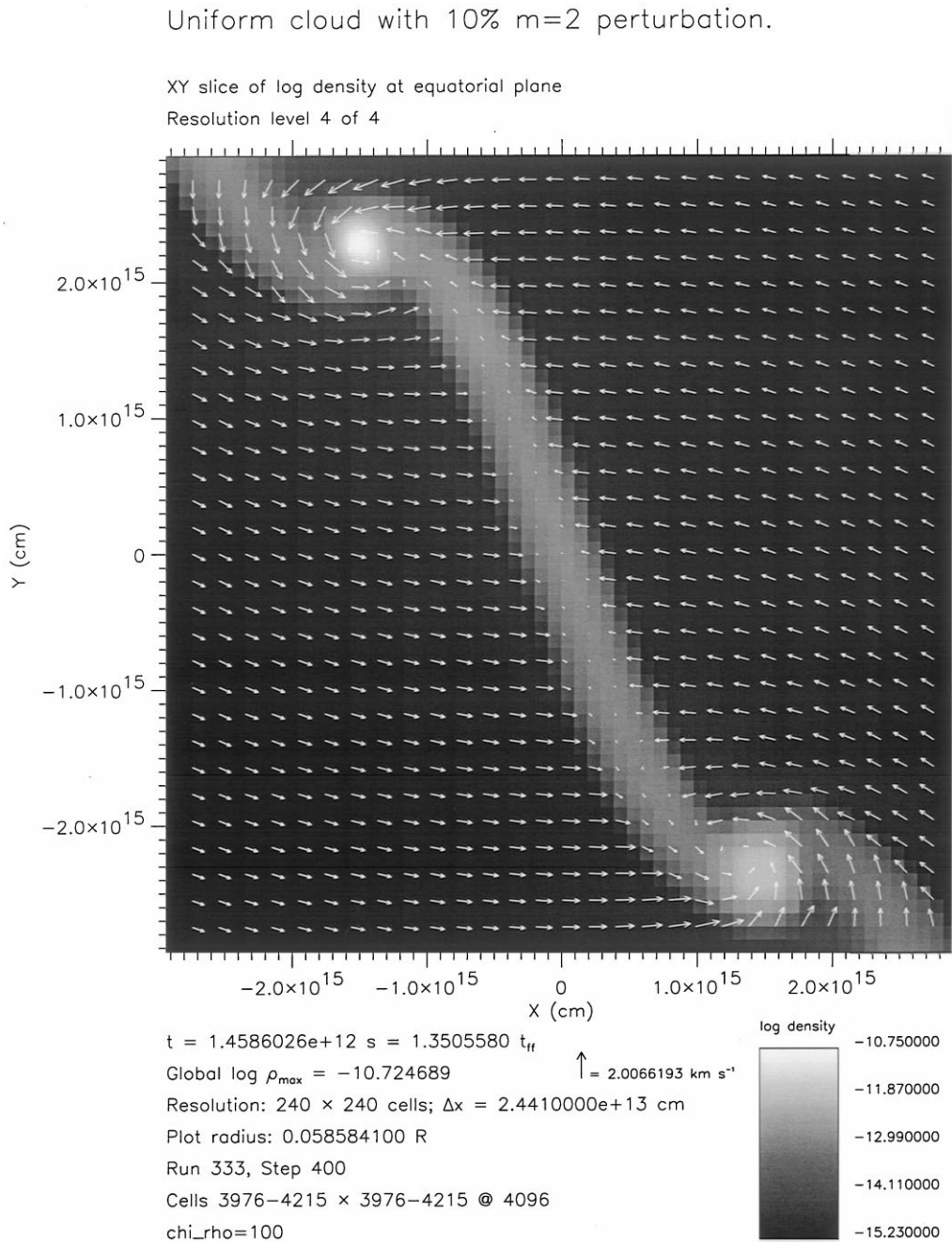


Fig. 4. Equatorial slice of log density at  $t = 1.46 \cdot 10^{12} \text{ s}$ . The binary cores are connected by a dense bar. Accretion flows from the surrounding disk which feed the bar result in accretion shocks on the surface of the bar.



Uniform cloud with 10%  $m=2$  perturbation.

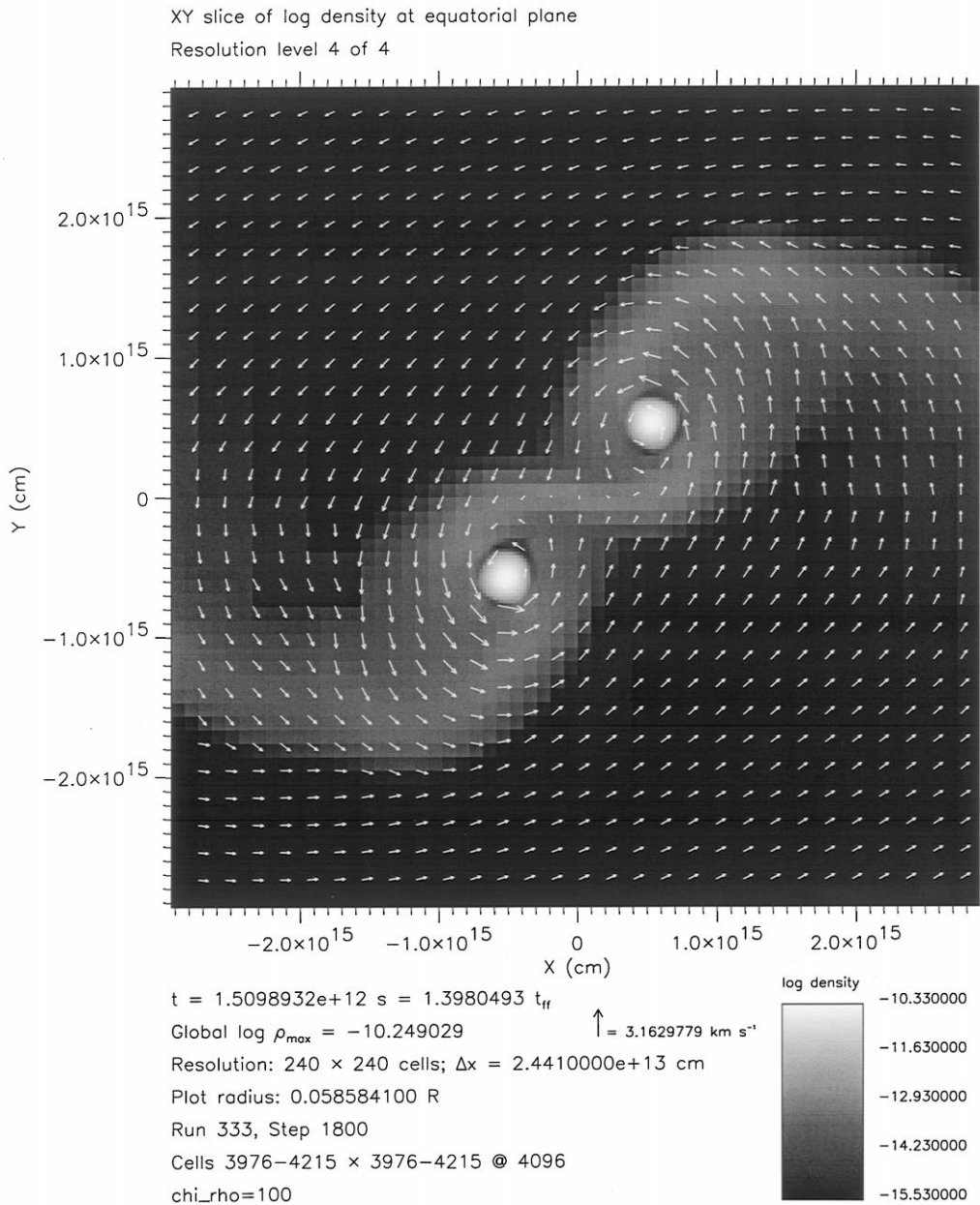


Fig. 5. Equatorial slice of log density at  $t = 1.51 \cdot 10^{12} \text{ s}$ . The first evidence of the formation of protostellar disks forming out of the spiral arms is seen at this time.

Uniform cloud with 10%  $m=2$  perturbation.

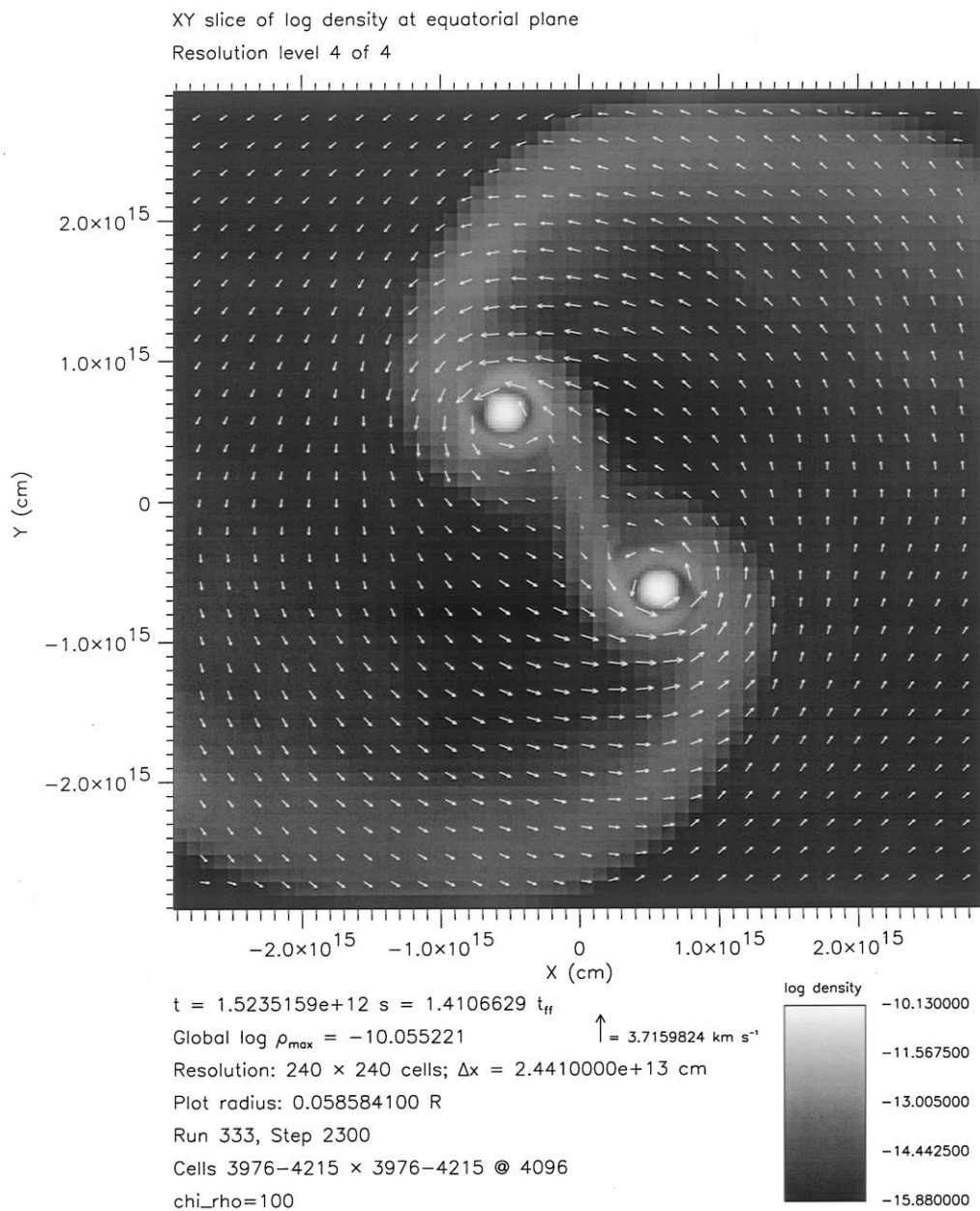


Fig. 6. Equatorial slice of log density at  $t = 1.52 \cdot 10^{12} \text{ s}$ .

Uniform cloud with 10%  $m=2$  perturbation.

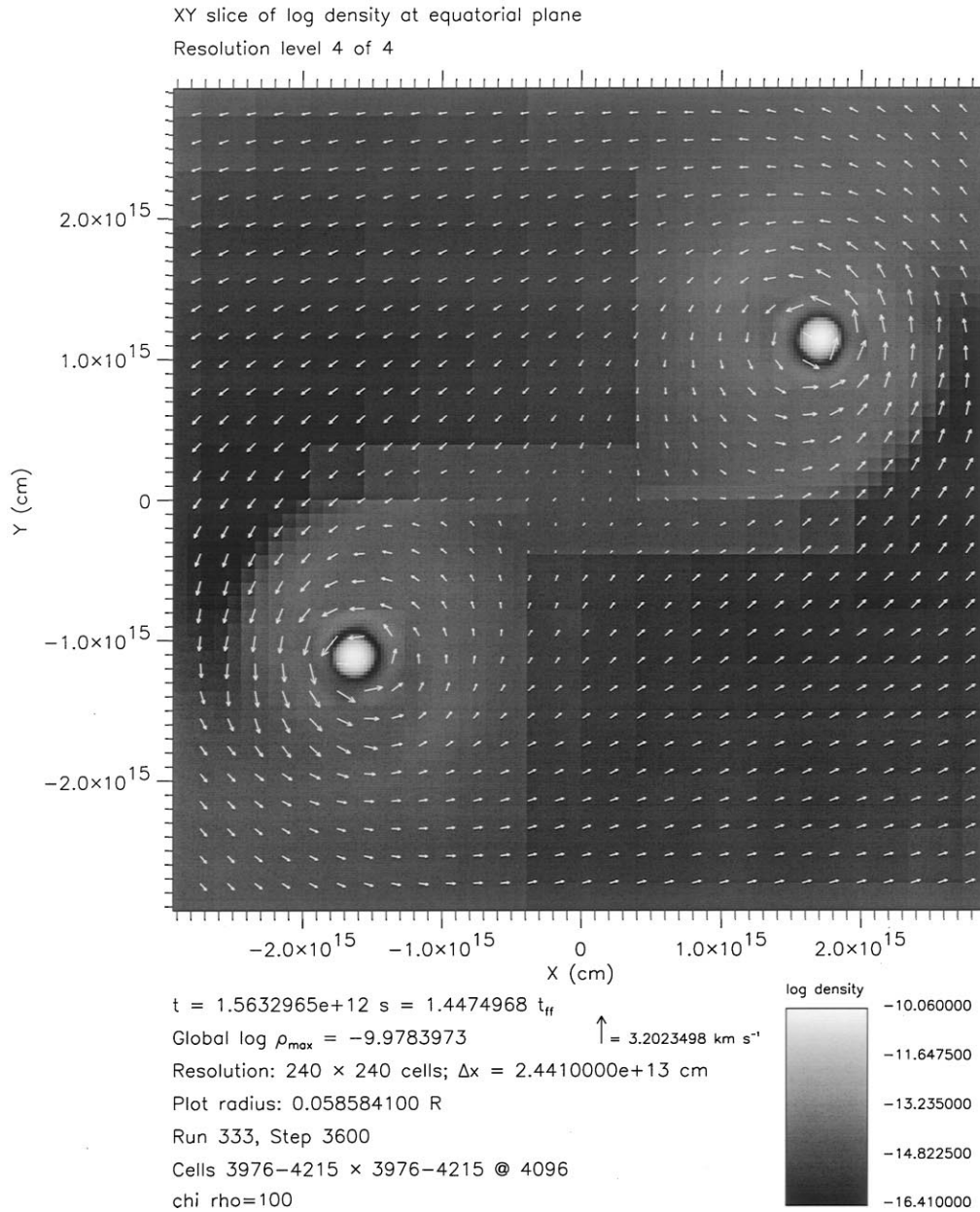


Fig. 7. Equatorial slice of log density at  $t = 1.56 \cdot 10^{12} \text{ s}$ . Protostellar disks detach but remain attached to large armed spiral in the disk surrounding the protostellar binary.

Uniform cloud with 10%  $m=2$  perturbation.

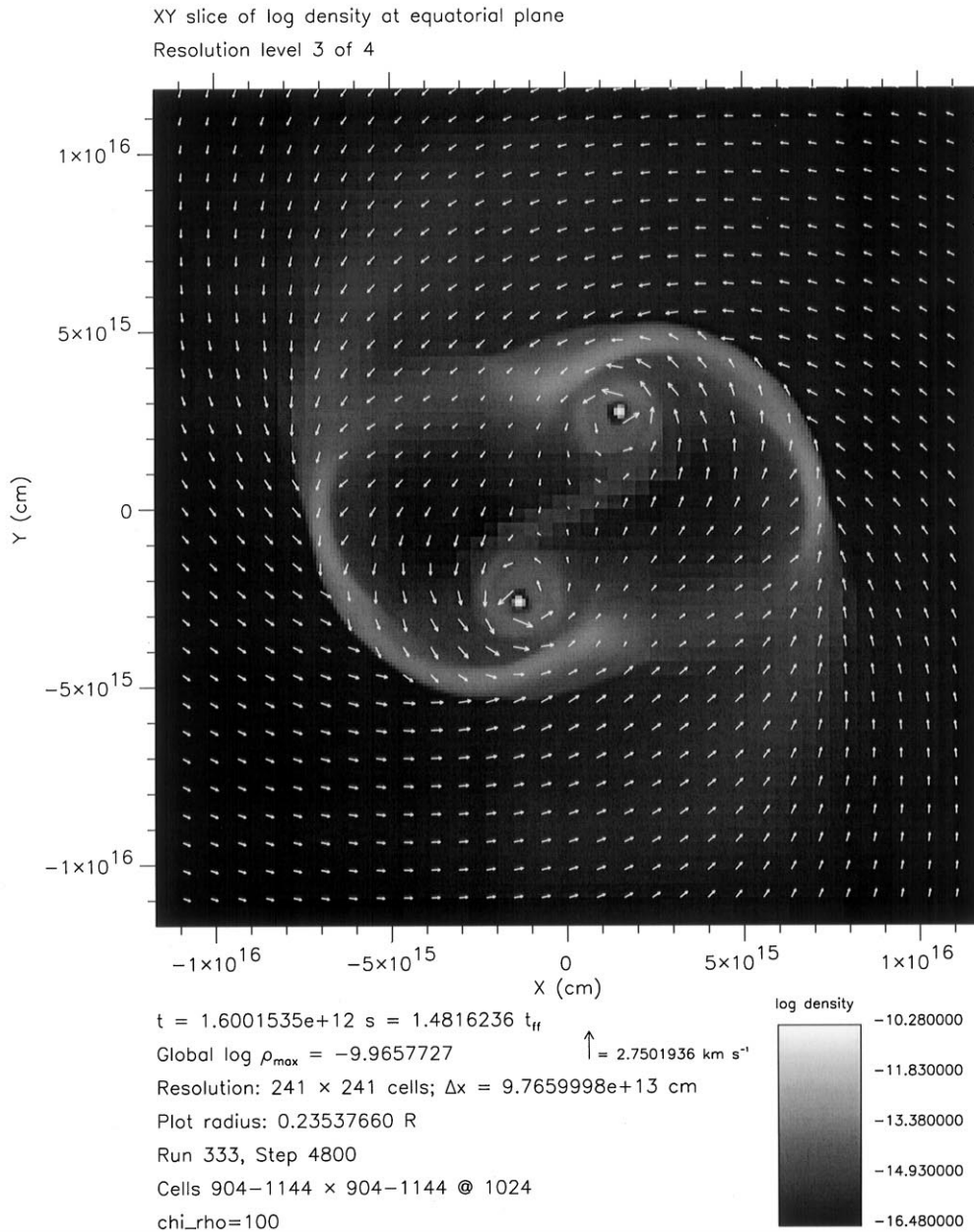


Fig. 8. Equatorial slice of log density at  $t = 1.60 \cdot 10^{12} \text{ s}$ . The protostellar disks have interacted with the surrounding spiral arms and perturbations in the spiral arms are evident. Fifty per cent of the initial mass of the cloud has accreted into the binary cores, protostellar disks and spiral arms at this time.

with observations of gaseous disks surrounding single T-Tauri stars and debris disks surrounding systems such as Beta Pictoris. Finally, at  $t = 1.6 \cdot 10^{12}$  s (Fig. 8) the binary protostellar disks/core system has moved back into the surrounding spiral arms. The cores have 20% of the mass of the initial cloud and the arms have 27% of mass of the initial cloud at this time. The protostellar disks comprise about 2% of the cloud mass.

#### 4. Conclusions

In this paper we have presented a powerful new method for numerical solution of 3-D self-gravitational hydrodynamic problems. This method combines a Godunov hydrodynamics integrator with a multigrid gravity solver in an adaptive mesh refinement framework. Guided by the Jeans condition for isothermal problems, and the use of Richardson extrapolation, AMR efficiently expends computational resources only when and where the features of the flow demand it. We presented results of this methodology applied to the collapse and fragmentation of isothermal molecular clouds that are both initially uniform and centrally condensed, evolved over 8–9 decades of collapse and subject to initial  $m=2$  mode perturbations. We found binary fragmentation results in good agreement with published calculations after evolution over the first half of this logarithmic range. By automatically inserting ever finer grids to maintain resolution of the Jeans length, these calculations did not generate the substantial artificial viscosity found in published calculations that used fixed finest resolutions. This allowed the collapse to be continued across the remainder of the above density range, revealing the final fate of isothermal, inviscid collapse: the formation of *singular filaments that do not subfragment*. This provides confirmation of the scenario outlined by Inutsuka and Miyama [16]: collapse of a rotating cloud first forms a disk, and perturbations within the disk gas result in fragmentation forming linear filaments. These filaments collapse to singularities in the absence of an intervening thermal or viscous effect. Finally, we presented preliminary calculations of the collapse on an initially uniform molecular cloud through the phase of isothermal collapse, and well into the nonisothermal phase. Through the use of high resolution AMR and satisfaction of the Jeans condition on the grid, we show the formation of protostellar disks surrounding a pair of binary cores, and discuss the role of the bar connecting the binary pair in the accretion onto the cores. The disks formed are consistent with observations of gaseous disks surrounding T-Tauri stars.

#### Acknowledgements

I express deep thanks to my collaborators Christopher McKee and Robert Fisher and Kelly Trulove who have participated in all phases of this work. My research on star formation is supported by a grant from NASA's Astrophysics Theory Program to the Center for Star Formation Studies. I acknowledge additional support under the auspices of the US Department of Energy at the Lawrence Livermore National Laboratory under contract W-7405-Eng-48. I wish to thank the Pittsburgh Supercomputing Center for provision of Cray C90 resources through grant AST940011P.

## References

- [1] M.R. Bate, I.A. Bonnell, N.M. Price, *Mon. Not. R. Astron. Soc.* 277 (1995) 362.
- [2] M.R. Bate, A. Burkert, *Mon. Not. R. Astron. Soc.* 288 (1997) 1060.
- [3] J.B. Bell, P. Colella, J. Trangenstein, *J. Comput. Phys.* 82 (1989) 362.
- [4] M.J. Berger, P. Colella, *J. Comput. Phys.* 82 (1989) 64.
- [5] M.J. Berger, J. Olinger, *J. Comput. Phys.* 53 (1984) 484.
- [6] A.P. Boss, *Astrophys. J.* 319 (1987) 149.
- [7] A.P. Boss, *Nature* 351 (1991) 298.
- [8] A.P. Boss, *Astrophys. J.* 468 (1996) 231.
- [9] A. Burkert, P. Bodenheimer, *Mon. Not. R. Astron. Soc.* 264 (1993) 798.
- [10] A. Burkert, P. Bodenheimer, *Mon. Not. R. Astron. Soc.* 280 (1996) 1190.
- [12] P. Colella, H.M. Glaz, *J. Comput. Phys.* 59 (1985) 264.
- [13] P. Colella, P.R. Woodward, *J. Comput. Phys.* 54 (1984) 174.
- [14] G. Golub, J.M. Ortega, *Scientific Computing*, Academic, New York, 1993.
- [15] S. Inutsuka, S.M. Miyama, *Astrophys. J.* 388 (1992) 392.
- [16] S. Inutsuka, S.M. Miyama, *Astrophys. J.* 480 (1997) 681.
- [17] J.H. Jeans, *Philos. Trans. A* 199 (1902) 1.
- [18] J.H. Jeans, *Astronomy & Cosmogony*, Cambridge University Press, London, 1928.
- [19] H. Kang et al., *Astrophys. J.* 430 (1994) 83.
- [20] J. Klapp, L. Sigalotti, F. DeFelice, *Astron. Astrophys.* 273 (1993) 175.
- [21] R.I. Klein, C.F. McKee, in: J. Franco et al. (Eds.), *Numerical Simulations in Astrophysics*, Cambridge, New York, p. 251.
- [22] R.I. Klein, C.F. McKee, P. Colella, *Astrophys. J.* 420 (1994) 213.
- [23] R.I. Klein, C.F. McKee, D.T. Woods, in: A. Ferrara et al. (Eds.), *Physics of the Interstellar Medium and Intergalactic Medium*, Astronomical Society of the Pacific, San Francisco, 1995, p. 366.
- [24] R.I. Klein, D.T. Woods, *Astrophys. J.* 497 (1998) 777.
- [25] R.I. Klein, R. Fisher, C.F. McKee, J.K. Truelove, in: S. Miyama, K. Tomisaka, T. Hanawa (Eds.), *Numerical Astrophysics 98*, Kluwer Press, Dordrecht, 1999, in press.
- [26] R.J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser, Boston, 1992.
- [27] H. Masunaga, S. Miyama, S. Inutsuka, *Astrophys. J.* 495 (1998) 346.
- [29] A.R. Mitchell, D.F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, Wiley-Interscience, New York, 1980.
- [30] J.J. Monaghan, *Ann. Rev. Astron. Astrophys.* 30 (1992) 543.
- [31] E.A. Myhill, A.P. Boss, *Astrophys. J. (Supp.)* 89 (1993) 345.
- [32] R.D. Richtmyer, K.W. Morton, *Difference Methods for Initial-Value Problems*, Interscience, New York, 1967.
- [33] L. Di G. Sigalotti, J. Klapp, *Astrophys. J.* 474 (1997a) 710.
- [34] L. Di G. Sigalotti, J. Klapp, *Astron. Astrophys.* 319 (1997b) 547.
- [35] L. Spitzer, *Physical Processes in the Interstellar Medium*, Wiley, New York, 1978.
- [36] J.K. Truelove, R.I. Klein, C.F. McKee, J.H. Holliman II, L.H. Howell, J.A. Greenough, *Astrophys. J.* 489L (1997) 179.
- [37] J.K. Truelove, R.I. Klein, C.F. McKee, J.H. Holliman II, L.H. Howell, J.A. Greenough, D.T. Woods, *Astrophys. J.* 495 (1998) 821.
- [38] P. Wesseling, *An Introduction to Multigrid Methods*, Wiley, New York, 1992.
- [39] K.-H.A. Winkler, M.J. Newman, *Astrophys. J.* 238 (1980) 311.
- [40] K.-H. Winkler, M. Norman, 71, in: D. Reidel (Ed.), *Astrophysical Radiation Hydrodynamics*, Dordrecht, 1986.
- [41] D.T. Woods, R.I. Klein, J.I. Castor, C.F. McKee, J. Bell, *Astrophys. J.* 461 (1996) 767.