# A MIXED METHOD POISSON SOLVER FOR THREE-DIMENSIONAL SELF-GRAVITATING ASTROPHYSICAL FLUID DYNAMICAL SYSTEMS

Comer Duncan
Department of Physics and Astronomy
Bowling Green State University
Bowling Green, OH 43403
and
Jim Jones
Computational Mathematics Group
Department of Mathematics
University of Colorado at Denver
Denver, CO 80217

## SUMMARY

A key ingredient in the simulation of self-gravitating astrophysical fluid dynamical systems is the gravitational potential and its gradient. This paper focuses on the development of a mixed method multigrid solver of the Poisson equation formulated so that both the potential and the Cartesian components of its gradient are self-consistently and accurately generated. The method achieves this goal by formulating the problem as a system of four equations for the gravitational potential and the three Cartesian components of the gradient and solves them using a distributed relaxation technique combined with conventional full multigrid V-cyles. The method is described, some tests are presented, and the accuracy of the method is assessed. We also describe how the method has been incorporated into our three-dimensional hydrodynamics code and give an example of an application to the collision of two stars. We end with some remarks about the future developments of the method and some of the applications in which it will be used in astrophysics.

## 1. Introduction

In recent years a number of astrophysicists [1]-[7] have developed simulation tools which build in increasingly realistic physics. The present work grew out of an ongoing effort by us to incorporate enough physics and to realize that physics with robust algorithms so that we can simulate both existing observed phenomena and make reliable predictions which the astronomers can utilize in making better observations and interpreting those observations. The ubiquitous existence of fluids and gravitation in the universe demands that, if we are to have even the most rudimentary simulation code, it must incorporate at least interacting fluids and gravitational physics. In this work, we restrict our attention to the weak-field, Newtonian limit of gravitation. The hydrodynamics code we have created also builds in the effects due to the Special Theory of Relativity, so the description of high speed phenomena is included. The restriction to weak-field gravity implies that the gravitational field is determined by the gravitational potential, which must be a solution to Poisson's equation in three dimensions subject to Dirichlet boundary conditions at the edges of the computational volume. In the coupled hydrodynamic-gravitational system, not only the potential but also its gradient is needed. The gradient contributes to the fluid's acceleration due to its self-gravity, inducing the momentum components to change.

The traditional procedure is to determine the potential by solving the Poisson equation with given Dirichlet boundary condition, then construct approximations to the components of the gradient via finite differencing the potential. However, in simulations of astrophysical gravitating fluids, the development of quite complex flows must be anticipated. Examples from astrophysics include supernova explosions, gravitational collapse, propagation of high-speed jets from active galactic nuclei, star collisions and disruptions in dense star clusters, and realistic models of the early universe. For most of these simulations, we need to compute the gradients of the gravitational potential as accurately as possible, which has motivated our development of an alternate approach to the gradient computation. Here we describe a method which can yield more robust gradients in systems that exhibit large variability in space. This is done using a distributed relaxation procedure coupled with full multigrid V-cycles and is described in Section 2. In Section 3 we present some tests of the method on three-dimensional systems. Section 4 presents our incorporation of it into the three-dimensional relativistic hydrodynamics code. Finally, we briefly describe an application of the code to the collision of two stars and comment on the applications for which the code can be used.

## 2. The Mixed Method Algorithm

The problems we are interested in are three-dimensional, and the results that we present in later sections are for such problems. However, in presenting the method, we will consider its two-dimensional version to make the description easier to understand and visualize. All components of the method, of both the discretization process and the multigrid algorithm, have natural three-dimensional analogs.

**a. The Finite Volume Element Discretization** Consider the following partial differential equation defined on some square domain $\Omega$ in $\mathcal{R}^2$:

$$\begin{cases} -\quad \nabla\cdot\nabla\phi &= f \quad \text{in} \quad \Omega, \\ \quad\phi &= g \quad \text{on} \quad \partial\Omega. \end{cases} \tag{1}$$

We let $u$ and $v$ denote the components of the gradient of -$\phi$:

$$(u,v)^t = -\nabla\phi$$

Then the partial differential equation may be written in the form of a first-order system in $\Omega$

$$\begin{cases} u &+ & \phi_x &= 0 & \text{(u equation)} \\ v &+ & \phi_y &= 0 & \text{(v equation)} \\ u_x &+ & v_y &= f & \text{(p equation),} \end{cases} \tag{2}$$

with boundary condition

$$\phi = g \text{ on } \partial\Omega.$$

Here the labels u,v, and p for the equations are introduced simply for convenience. To discretize this system, we follow the Finite Volume Element principles developed in [8]. Consider a uniform square mesh $\Omega^h$ with mesh size $h$ that covers $\Omega$. We introduce three sets of control volumes, one for each of the three equations in Eq.**2**. These volumes are shown in Fig. **1**. We denote by $\mathcal{U}$ the set of all volumes $\mathbf{U}$ that will be used to discretize the u equation in Eq.**2**. Similarly, we will use the notation $\mathcal{V}$ and $\mathcal{P}$ for the sets of volumes $\mathbf{V}$ and $\mathbf{P}$ for the v and p equations respectively. For our finite element space we consider the lowest order Raviart-Thomas elements on the triangulation given by the volumes $\mathcal{P}$:

$u^h$ is linear in $x$ and constant in $y$ on each $\mathbf{P} \in \mathcal{P}$,
$v^h$ is linear in $y$ and constant in $x$ on each $\mathbf{P} \in \mathcal{P}$,
$\phi^h$ is constant on each $\mathbf{P} \in \mathcal{P}$.

The location of the nodes for each of the unknowns with their indexing is also shown in Fig. **1**. We can now disretize the equations. We take the u equation in Eq.**2** and integrate it over each $\mathbf{U} \in \mathcal{U}$. As an example, let $\mathbf{U}_{i,j}$ be the volume in $\mathcal{U}$ that is centered at the interior $u^h$ node $(i, j)$. We then have

$$\int_{\mathrm{U}_{i,j}} (u + \phi_x)\, dxdy \qquad = 0,$$

which implies

$$\tfrac{h^2}{8}(u^h_{i-1,j} + 6u^h_{i,j} + u^h_{i+1,j}) + h(\phi^h_{i+1,j+1} - \phi^h_{i,j+1}) \quad = 0.$$

Integrating the v equation in Eq.**2** over an interior **V** volume yields a similar discrete expression involving nodal values of $v^h$ and $\phi^h$. Integrating the p equation in Eq.**2** over the volume in $\mathcal{P}$ centered at the interior $\phi^h$ node $(i, j)$; denote this volume by $\mathbf{P}_{i,j}$; we get

$$\int_{\mathrm{P}_{i,j}} (u_x + v_y)\, dxdy \qquad = \int_{\mathrm{P}_{i,j}} f dxdy$$

which implies

$$h(u^h_{i,j-1} - u^h_{i-1,j-1} + v^h_{i-1,j} - v^h_{i-1,j-1}) \quad = h^2 f_{i,j}.$$

Here, $f_{i,j}$ is the value of $f$ at the $\phi$ node $(i, j)$, which results from assuming that $f$ is (approximated by) a piecewise constant function on $\mathcal{P}$. The only remaining part of the discretization involves integrating the u equation in Eq.**2** over the "half size" **U** volumes on the left and right boundaries, and similarly integrating the v equation in Eq.**2** over the "half size" **V** volumes on the lower and upper boundaries. We illustrate this process by integrating the u equation in Eq.**2** over the volume $\mathbf{U}_{1,j}$ that has the boundary $u^h$ node $(1, j)$ as the midpoint of its left edge. We have

$$\int_{\mathrm{U}_{1,j}} (u + \phi_x)\, dxdy \qquad = 0,$$

which implies

$$\tfrac{h^2}{8}(3u^h_{1,j} + u^h_{2,j}) + h(\phi^h_{2,j+1} - \phi^h_{1,j+1}) \quad = 0$$

or

$$\tfrac{h^2}{8}(3u^h_{1,j} + u^h_{2,j}) + h(\phi^h_{2,j+1}) \qquad = h\phi^h_{1,j+1}.$$

Note that $\phi^h_{1,j+1}$ is on the boundary and hence is known. To summarize, the discretization has produced for each **U** volume a discrete version of the u equation in Eq.**2**, for each **V** volume a discrete version of the v equation in Eq.**2**, and for each **P** volume a discrete version of the p equation in Eq.**2**.

**b. The Multigrid Algorithm** We assume that the reader is familiar with the fundamentals of multigrid methods; good references are [9],[8], [10]. We consider a family of uniform square grids $\Omega^h$ that cover our region $\Omega$, where $h$ denotes the mesh size. Fig. **2** shows a coarse grid $\Omega^{2h}$, with twice the mesh size of the grid $\Omega^h$ in Fig. **1**. On each grid $\Omega^h$, we can apply the Finite Volume Element discretization process, and we write the discrete set of equations that this process generates as

$$L^h z^h = F^h, \tag{3}$$

where $z^h = (u^h, v^h, \phi^h)^t$ and $F^h = (fu^h, fv^h, f^h)^t$ and the unknowns, $u^h$, $v^h$, and $\phi^h$, are the nodal values of the corresponding functions on the grid $\Omega^h$. Note that the values of $\phi$ at nodes on the boundary are known so they are not included in $\phi^h$; however, as mentioned in the last section, these boundary values of $\phi$ do appear in the equations generated by integration over the **U** and **V** volumes near boundaries, resulting in the possibly nonzero terms $fu^h$ and $fv^h$ in Eq.**3**. In this section, we now define the basic components of relaxation, interpolation, and restriction that are necessary to implement a multigrid algorithm.

For the equations on a grid $\Omega^h$, we use a distributive relaxation process similar to that presented in [10]. We can think of relaxation as a three step process. First, we sweep over all of the $u^h$ nodes, change the value of $u^h_{i,j}$ so that the U equation at $(i, j)$ is satisfied. Second, we perform a similar Gauss-Seidel relaxation of all of the V equations. Note that these two steps, the U and V relaxation, are independent of each other and could be performed in parallel. Finally, we step over the $\phi^h$ nodes and change the value of $\phi^h_{i,j}$ and the values of $u^h$ and $v^h$ that lie on the edge of the volume $\mathbf{P}_{i,j}$, namely $u^h_{i,j-1}$, $u^h_{i-1,j-1}$,$v^h_{i-1,j}$, and $v^h_{i-1,j-1}$. We change these five unknowns so that the P equation at $(i, j)$ is satisfied and so that the residuals of the U equations at $(i, j-1)$ and $(i-1, j-1)$ and of the V equations at $(i-1, j)$ and $(i-1, j-1)$ are unchanged. To allow vectorization, the Gauss-Seidel relaxation performed in each step is done in a red/black ordering.

For defining interpolation operators, we use the same principles as outlined in [8]. The Finte Volume Element discretization is based on finite element spaces for the variables $u^h$,$v^h$, and $\phi^h$, so we can simply use the relationship between the finite element spaces on the different grids to define interpolation. To define the interpolation operator for $\phi$, which we denote as $I(\phi)^h_{2h}$, we note that $\phi^{2h}$ is constant on the grid $2h$ volume $\mathbf{P}_{I,J}$.

Referring to Figs. **1** and **2**, we thus have the following characterization of $\phi^h = I(\phi)^h_{2h}\phi^{2h}$:

$$\phi^h_{i,j} = \phi^h_{i+1,j} = \phi^h_{i,j+1} = \phi^h_{i+1,j+1} = \phi^{2h}_{I,J}.$$

To define the interpolation operator for $u$, which we denote as $I(u)^h_{2h}$, we note that $u^{2h}$ is linear in $x$ and constant in $y$ on the grid $2h$ volume $\mathbf{P}_{I,J}$. We thus have the following characterization of $u^h = I(u)^h_{2h}u^{2h}$. (See Figs. **1** and **2**)

$$\begin{aligned}
u^h_{i-1,j-1} &= u^h_{i-1,j} = u^{2h}_{I-1,J-1}\\
u^h_{i+1,j-1} &= u^h_{i+1,j} = u^{2h}_{I,J-1}\\
u^h_{i,j-1} &= u^h_{i,j} = 1/2(u^{2h}_{I-1,J-1} + u^{2h}_{I,J-1}).
\end{aligned}$$

The definition of the interpolation operator for $v$ is similar.

For defining restriction operators, we again use the same principles as outlined in [8]. In the correction scheme multigrid algorithm, which we use here, restriction operators are used to transfer right-hand sides and residuals of equations, not the unknowns themselves. The definitions of the restriction operators are based on the relationship between the volumes on the various grids. The idea is to lump several of the grid $h$ right-hand sides to produce the grid $2h$ right hand sides. To define the restriction operator for the P equation, which we denote as $I(P)^{2h}_h$, we note that a grid $2h$ volume $\mathbf{P}_{I,J}$ wholly contains four grid $h$ **P** volumes. We thus have the following characterization of $f^{2h} = I(P)^{2h}_h f^h$, referring again to Figs. **1** and **2**:

$$f^{2h}_{I,J} = f^h_{i,j} + f^h_{i+1,j} + f^h_{i,j+1} + f^h_{i+1,j+1}.$$

To define the restriction operator for the U equation, which we denote as $I(U)^{2h}_h$, we note that a grid $2h$ volume $\mathbf{U}_{I,J}$ in the interior of $\Omega$ wholly contains two grid $h$ **U** volumes and half of four others. We thus have the following characterization of $fu^{2h} = I(U)^{2h}_h fu^h$, again referring to Figs. **1** and **2**:

$$fu^{2h}_{I,J-1} = fu^h_{i+1,j} + fu^h_{i+1,j-1} + 1/2(fu^h_{i,j} + fu^h_{i,j-1} + fu^h_{i+2,j} + fu^h_{i+2,j-1}).$$

The relationship between **U** volumes at boundaries is different; for example, the grid $2h$ **U** volumes on the left boundary of $\Omega$ wholly contain two of the grid $h$ **U** volumes and half of two others, yielding Figs. **1** and **2**:

$$fu^{2h}_{1,J-1} = fu^h_{1,j} + fu^h_{1,j-1} + 1/2(fu^h_{2,j} + fu^h_{2,j-1}).$$

The definition of the restriction operator for the V equation is done in a similar fashion.

## 3. Tests of the Mixed Method Algorithm

A standard approach to Eq.**1** is to solve a discrete equation based on cell-centered finite differences for approximating $\phi$, then to use simple differencing of this approximation to get the components of its gradient. We performed some numerical tests to investigate what advantage, in terms of accuracy, the mixed method provides over this standard approach. These tests were for problems with exact solution $\phi(x,y,z) = \sin(k_1x)\sin(k_2y)\sin(k_3z)$ with $\Omega = [0,\pi]^3$. By varying $k_1$,$k_2$, and $k_3$, we were able to see the effect that oscillations in the solution had on the accuracy of the methods. Below are results for some of these tests on a grid with 32 cells in each direction.

| | | | MIXED METHOD | | STANDARD METHOD | |
|---|---|---|---|---|---|---|
| $k_1$ | $k_2$ | $k_3$ | $\phi_{err}$ | $(\phi_x)_{err}$ | $\phi_{err}$ | $(\phi_x)_{err}$ |
| 1 | 1 | 1 | $7.90E-4$ | $8.15E-4$ | $1.58E-3$ | $8.15E-4$ |
| 1 | 16 | 16 | $1.47E-1$ | $1.50E-1$ | $4.59E-1$ | $4.72E-1$ |
| 16 | 1 | 1 | $1.46E-1$ | $3.61E-0$ | $4.56E-1$ | $3.53E-0$ |
| 16 | 16 | 16 | $1.47E-1$ | $3.60E-0$ | $4.60E-1$ | $3.60E-0$ |

Here, $\phi_{err}$ and $(\phi_x)_{err}$ are pointwise $l_2$ norms of the error in $\phi$ and its $x$ derivative scaled by the volume term $h^3$. These results are indicative of results seen for other combinations of $k_1$,$k_2$, and $k_3$. For smooth solutions, the methods give nearly identical results. However, for oscillatory solutions, the mixed method gives more accurate results, particularly for $\phi$.

## 4. Incorporation of the Mixed Method Solver into the Three-Dimensional Hydrodynamics Code

**a.The Physics and the Code** The physics included in the present code consists of a perfect fluid with an adiabatic equation of state formulated in a generally covariant manner. The interval between events in spacetime is represented in the present work in the form

$$ds^2 \;=\; -(\alpha^2 - \beta_i\beta^i)dt^2 \;+\; \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt). \tag{4}$$

The function $\alpha$ is called the lapse and represents the lapse of proper time at a given spatial point. The vector field $\beta^i$ is called the shift vector and determines how much the spatial coordinates shift from one $t = constant$ slice to the next infinitesimally later one. The second rank symmetric tensor field $\gamma_{ij}$ is the metric tensor of the spatial geometry. In the general theory of relativity[12], the four-dimensional geometry of spacetime is dynamic and the lapse, shift, and three metric are related to the kinematical description of the coordinates of the observer and the spatial geometry. The fluid energy-momentum tensor must obey a local conservation law in order to be consistent with Einstein's theory. When supplemented with the conservation of Baryons, the conservation laws can be written in the following form:

*Rest-mass conservation*

$$\frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial t}(\gamma^{\frac{1}{2}}d) + \frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial x^i}(\gamma^{\frac{1}{2}}dv^i) = 0 \tag{5}$$

*Internal energy equation*

$$\frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial t}(\gamma^{\frac{1}{2}}e) + \frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial x^i}(\gamma^{\frac{1}{2}}ev^i) = -P(\frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial t}(\gamma^{\frac{1}{2}}W) + \frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial x^i}(\gamma^{\frac{1}{2}}Wv^i)) \tag{6}$$

*Momentum equation*

$$\begin{aligned}\frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial t}(\gamma^{\frac{1}{2}}S_j) + \frac{1}{\alpha\gamma^{\frac{1}{2}}}\frac{\partial}{\partial x^j}(\gamma^{\frac{1}{2}}S_jv^i) = &-\frac{\partial P}{\partial x^j} + (d+e+PW)W\frac{\ln\alpha}{\partial x^j} \\ &-\frac{S_i}{\alpha}\frac{\partial\beta^i}{\partial x^j} - \frac{1}{2W}\frac{S_kS_l}{(d+e+PW)}\frac{\partial\gamma^{kl}}{\partial x^j}\end{aligned} \tag{7}$$

The variables $d, e$, and $S_i$, which are used in the code, are defined as follows: $d = \rho W$, $e = \rho\epsilon W$, and $S_i = (\rho + \rho\epsilon + P)u_i$. Here $d$, $e$, and $S_i$ are respectively the coordinate mass density, internal energy density, and covariant components of the relativistic momentum density. Eq.(**5**-**7**) are the equations of general relativistic fluid dynamics in a general background spacetime. Since the present paper is restricted to the study of phenomena with weak gravitational fields, we introduce the following Newtonian approximations to the lapse, shift, and three-metric in Cartesian coordinates:

$$\alpha \simeq 1 + \phi \tag{8}$$

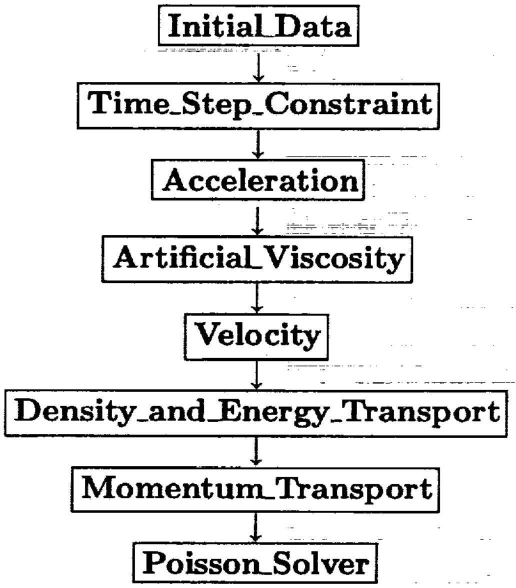
$$\begin{aligned}\beta^i &\simeq 0 &\quad (9)\\ \gamma_{ij} &\simeq \delta_{ij}. &\quad (10)\end{aligned}$$

The scalar field $\phi$ is the Newtonian gravitational potential and must satisfy the Poisson equation

$$\nabla^2\phi = 4\pi G\rho, \qquad (11)$$

in the computational volume and Dirichlet boundary conditions on the volume edges.

With the Newtonian approximation to the geometric variables it then follows that the self-gravity of the fluid contributes to the change in the momentum density through the term $\rho\nabla\alpha$. The value of $\alpha$ itself enters several places in the fluid equations. Thus, a complete characterization of the self-gravitating fluid dynamics requires both the lapse and its gradient vector. It is these quantities that our mixed method computes in a robust manner. Concerning the elements that constitute the hydrodynamics part of the code, the methods used may be characterized as explicit finite volume schemes. The physical variables $d, e$, and $S_i$ are the fundamental quantities. These variables are discretized on a staggered grid system with the conventions that scalar variables such as density are stored at zone centers, while vector variables are centered on the faces of the zones. The biggest challenge is by far to treat the advection of the physical variables as accurately as possible. This is especially true for the astrophysical applications, since complex flows abound. We want the code to be able to detect and track shocks adequately. The advection method implemented in the code is based on a monotonic advection algorithm due originally to Van Leer [11]. It is robust and tracks shocks reasonably well. The code uses artificial viscosity to smooth developing discontinuities over a few zones. For this we use an artificial viscosity pressure, which is a combination of linear and quadratic functions of the monotonized four-velocity differences. The code uses an adiabatic equation of state of the form $P = (\Gamma - 1)\rho\epsilon$, where $\Gamma$ is the parameter that characterizes the equation of state and can itself be a function of the thermodynamic variables and position. For the model stars we discuss here, $\Gamma$ is chosen to be a constant. The overall structure of a single computational step of the code is described in [7] and illustrated as follows:



At the end of the computational step the fully updated physical variables are available. The **Poisson_Solver** routine is invoked and it is here that we utilize our mixed method solver, which returns $\phi$ and $\nabla\phi$.

**b.Application to Collision of Stars** As a nontrivial application of the code, we present a summary of the results of using the mixed method Poisson solver in the simulation of the collision of two stars which are initially in equilibrium. The initial data were chosen so that the mass density and energy density correspond to two equilibrium spherical stars. We have chosen the $n = 1$ polytropic equation of state. This equation of state has the following functional forms for the initial mass density and energy density: $d = d_0 \frac{sin(\xi)}{\xi}$ and $e = e_0 \left(\frac{\sin\xi}{\xi}\right)^2$, where $\xi = \pi r/r_0$ and $r_0$ is the equilibrium radius of the star. The two model stars were placed with their centers displaced in the $z = 0$ plane. We show here the results of simulations in which the radii were chosen equal to $0.26R_{solar}$ and the central mass density $d_0$ equal to $6.6g/cm^3$. The central temperature of each star was chosen to be 4.0e06 K. The simulations shown here were all done with a $(66)^3$ grid. All computations were performed on the Ohio Supercomputer Center's Cray YMP8/864. The hydrodynamics part of the code has been highly vectorized.

Fig. **3a** shows the contours for the initial potential and its gradient components in the $z = 0$ plane for a run of an off-center collision. The stars were chosen initially to have a relative velocity comparable to the orbital velocity. Fig. **3b** is a plot of the density contours and velocity field in the z=0 plane. Subsequent motion is induced by the combined effects of the initial momentum and the self-gravity of the two stars. Because the stars attract each other, they develop accelerations toward each other and the hydrodynamics that results alters the density and energy distributions. Typical simulations were run for at least on the order of the gravitational free-fall time. Given the combined interactions of the hydrodynamics with self-gravity, we expect disruption of the two stars if the collision is sufficiently violent. Figs. **4a,b** show respective snapshots of the potential contours and gradient and density contours and velocities for late times in the off-center collision.

We conclude from these simulations and others that the mixed method Poisson solver produces physically acceptable results when combined with the three-dimensional hydrodynamics. This code is currently being used to simulate higher resolution runs and other multiple-star systems. We will be using the present code to treat the collision of two neutron stars and compute its final state and the amount of gravitational radiation emitted by such systems. Such computations are of importance because they can shed light on the astrophysics of the mergers of neutron stars as well as provide potentially important benchmarks of how much gravitational radiation should be expected from such encounters.

### Acknowledgements

The authors wish to thank Steve McCormick for many useful conversations about the mixed method algorithm. We thank Jeyakumari Khan for her assistance with the merging of the mixed method Poisson solver and the hydro code. C. Duncan acknowledges support from Cray Research and the Ohio Supercomputer Center, where the computations were performed.

## References

[1] R. L. Bowers and J. R. Wilson, *Numerical Modeling in Applied Physics and Astrophysics* , Jones & Bartlett, Boston, 1991.

[2] J. M. Stone and M. L. Norman, Astrophysical Journal Supplement Series **80**, 753 (1992).

[3] A. Abrahams, D. Bernstein, D. W. Hobill, E. Seidel, and L. Smarr, Physical Review **D45**, 3544 (1992).

[4] P. Laguna, H. Kurki-Suonio, and R. A. Matzner, Physical Review **D44**, 3077 (1991).

[5] A. Abrahams and C. R. Evans, Physical Review **D46**, R4117 (1992).

[6] T. Nakamura, *Proceedings of the Sixth Marcel Grossmann Meeting on General Relativity*, 1992.

[7] G. Comer Duncan, Proceedings of the First Midwest Relativity Conference, National Center for Supercomputing Applications, March, 1992; Bull. Am. Phys. Soc. (with G. Shastri), May 1992 OSAPS, Cincinnati.

[8] S. F McCormick, *Multilevel Adaptive Methods for Partial Differential Equations*, Vol. 6 in Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[9] William L. Briggs, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[10] A. Brandt, *Multigrid Techniques : 1984 Guide*, The Weizmann Institute of Science, Rehovot, Israel.

[11] B. Van Leer, J. Computational Physics **32**, 101 (1979).

[12] C. Misner, K. Thorne, and J. Wheeler *Gravitation*, Freeman, San Francisco, 1973.

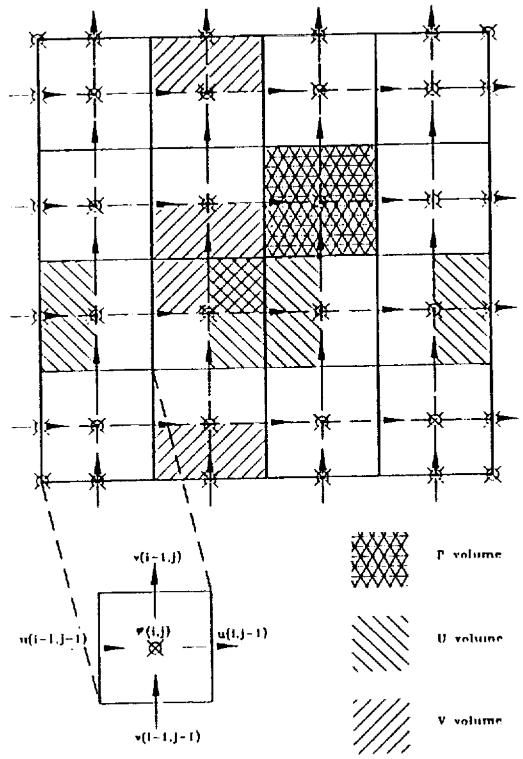


Figure 1

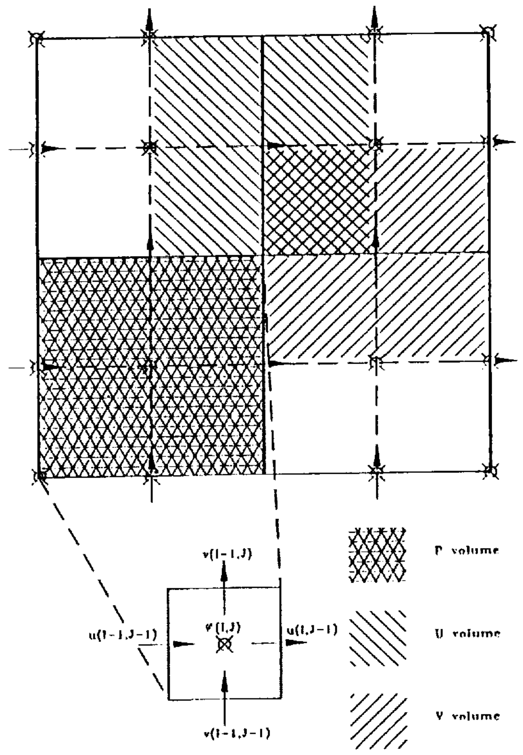


Figure 2

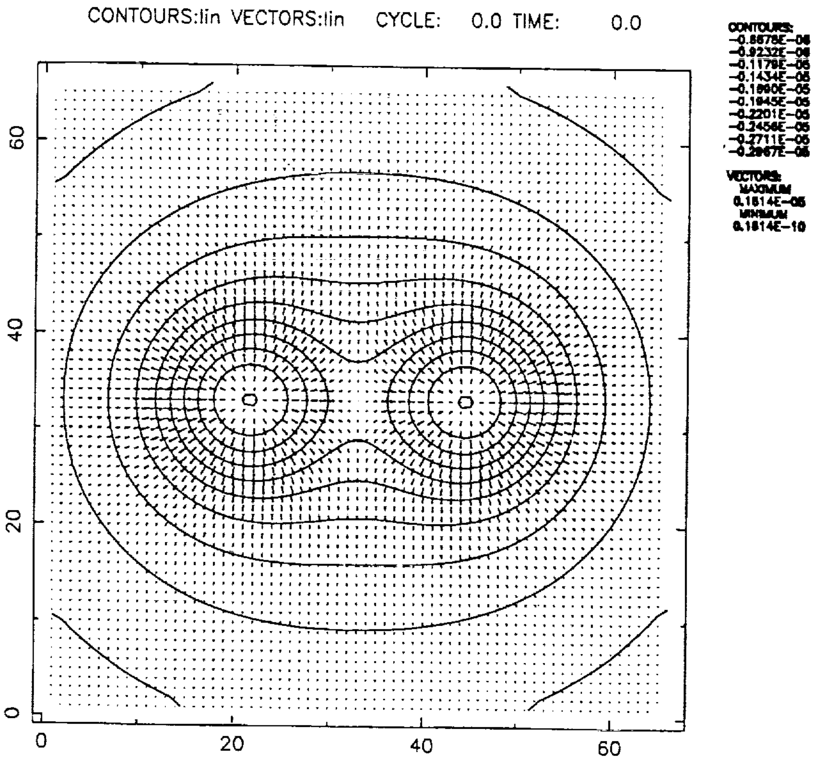


Figure 3a. Initial Gravitational Potential and Gradient

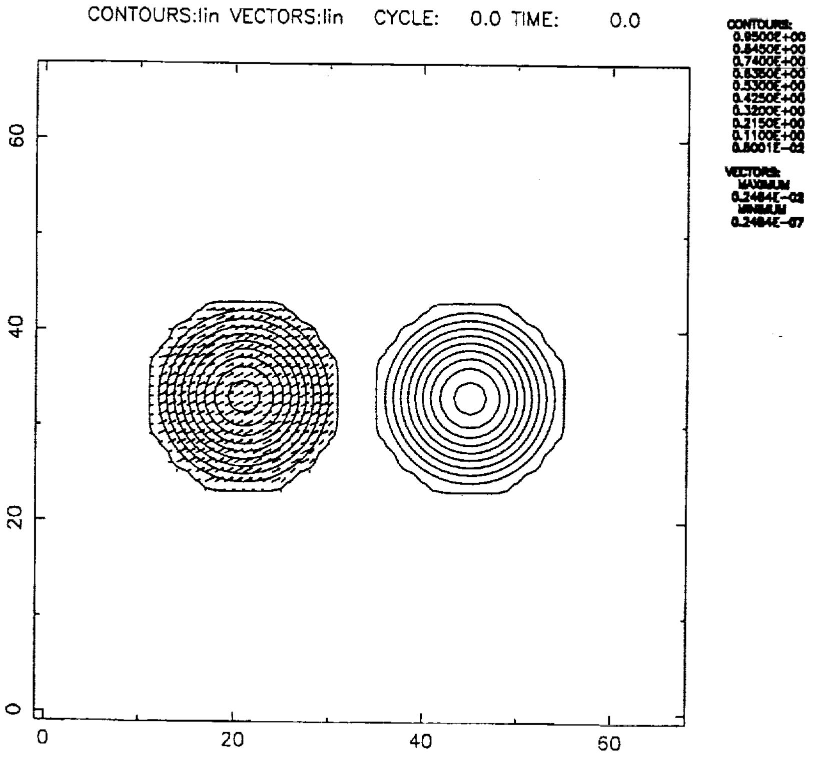


Figure 3b. Initial Density and Velocity

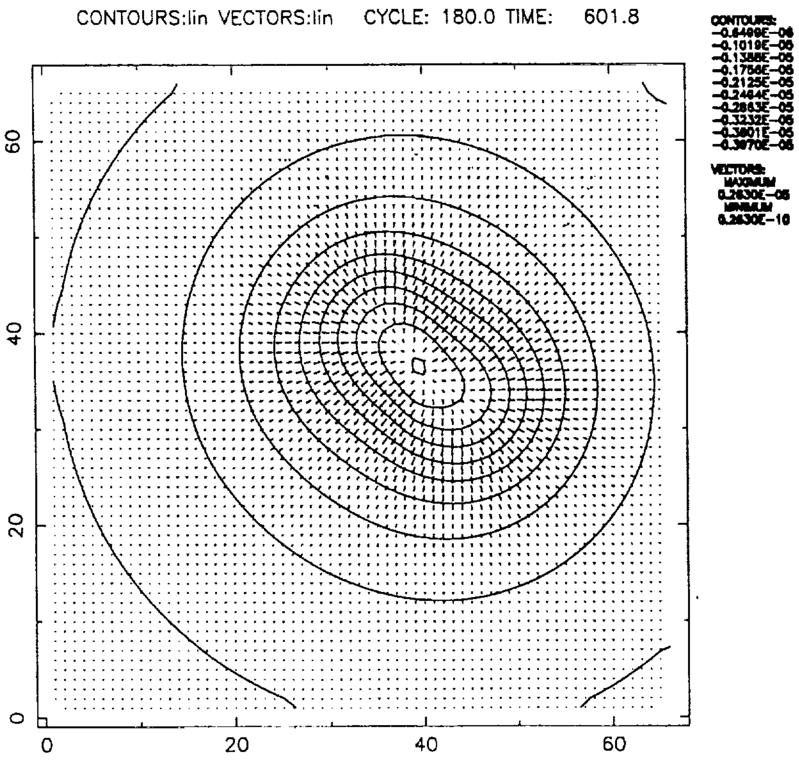


Figure 4a. Gravitational Potential and Gradient

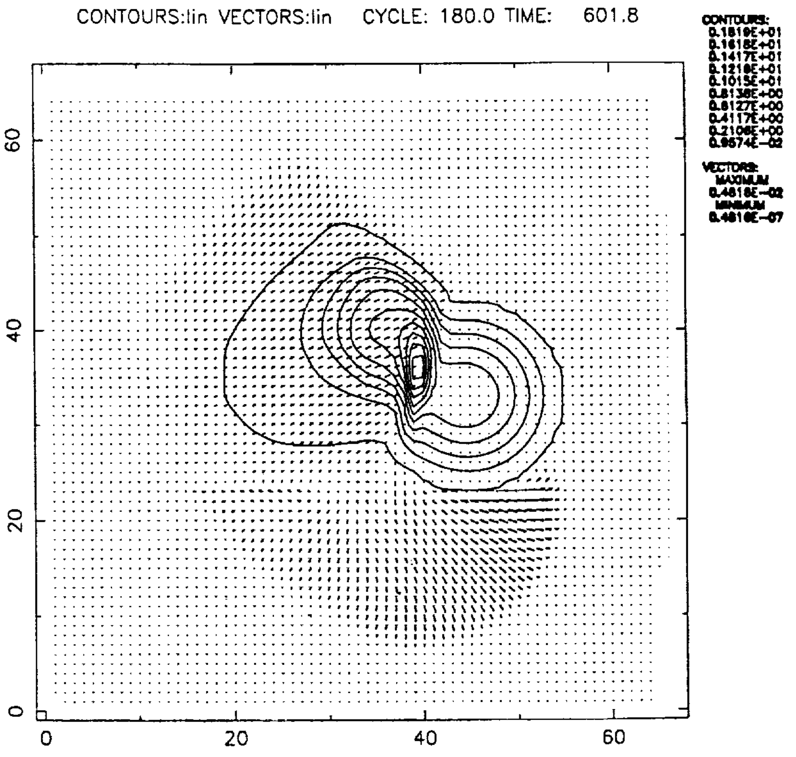


Figure 4b. Density and Velocity