Comparison of flux-correcting and spline algorithms for solving (3+1)-dimensional relativistic hydrodynamics

D. J. Dean,[1,2] C. Bottcher,[2] M. R. Strayer,[2] J. C. Wells,[2,3]
A. von Keitz,[4] Y. Pürsün,[4] D.-H. Rischke,[4] and J. A. Maruhn[5,4]

[1] *Kellogg Radiation Laboratory, California Institute of Technology, Pasadena, California 91125*
[2] *Center for Computationally Intensive Physics, Physics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831*
[3] *Vanderbilt University, Department of Physics and Astronomy, Nashville, Tennessee 37235*
[4] *Institute for Theoretical Physics, University of Frankfurt, Frankfurt, Germany*
[5] *Joint Institute for Heavy Ion Research, Holifield Heavy Ion Research Facility, Oak Ridge, Tennessee 37831*
(Received 16 July 1993)

We study flux-correcting and spline algorithms methods for solving relativistic hydrodynamic equations. Differences in the methods are contrasted, and comparisons are given for the ultrarelativistic collision of two nuclei within the framework of a one-fluid model. Both methods give surprisingly similar results, although differences in detail are apparent.

PACS number(s): 02.70.−c, 47.75.+f

I. INTRODUCTION

Presently ultrarelativistic heavy-ion collisions exhibit many of the properties of colliding fluids [1]. Laboratory energies of 200 GeV per nucleon are presently available, corresponding to relativistic Lorentz gamma factors $\gamma_{\text{c.m.}} \simeq 10$ in the center-of-mass frame of the colliding nuclei, while future experiments in the TeV energy regime are planned. For laboratory energies below 20 GeV per nucleon ($\gamma_{\text{c.m.}} \simeq 3$) the experimental results are fairly well reproduced using a relativistic hydrodynamic model [1]. However, the numerical procedures used in these calculations yield difficulties if applied to relativistic collisions with $\gamma_{\text{c.m.}} \geq 10$. Typical problems are related to the violation of causality, and arise as inconsistencies in the laboratory frame mass density, R, momentum density, $\mathbf{M}$, and energy density, E. In the dynamical evolution, the relativistic conditions, $E \geq R$ and $E \geq |\mathbf{M}|$, fail, and the calculated fluid velocities exceed the speed of light. It has been argued [2] that a fully implicit treatment of the relativistic hydrodynamic equations is required to obtain a consistent solution for large gamma factors; however, presently only explicit schemes are in use [3,4]. Nevertheless, fully implicit methods for solving these equations has also been developed [5–7]. Since analytic solutions for the hydrodynamic equations are known for only a few special cases, a comparison of different numerical algorithms used in a realistic calculation is worthwhile.

In this paper, we discuss two methods for solving the three-dimensional relativistic hydrodynamic equations of motion. One is based on the sharp and smooth transport algorithm (SHASTA), a version of the flux-corrected transport (FCT) algorithm [8–11]. The second algorithm employs a basis-spline collocation method, a high-order interpolation algorithm [12].

The two numerical algorithms have been applied to one-dimensional test cases [3,6]. In each test calculation, a reasonable agreement in comparison with the analytical solution was obtained. Both algorithms have been found to give satisfactory results for spherically expanding matter distributions. However, simulations of heavy-ion collisions exhibit a more complex behavior with the formation and propagation of multiple shocks. Thus it is of greater interest and importance to study three-dimensional test cases for which no analytical or semianalytical solutions exist. The principal aim of the current work is to carry out such a comparison for typical nucleus-nucleus collisions. The format of this paper is as follows. In Sec. II the hydrodynamic equations are defined. In Sec. III we introduce the two numerical methods. A discussion of the comparison is given in Sec. IV. In Sec. V we summarize our results.

II. THE HYDRODYNAMIC EQUATIONS

In general, the relativistic energy-momentum tensor in the ideal hydrodynamic limit [13] is

$$T^{\mu\nu} = (\varepsilon + P)u^{\mu}u^{\nu} - g^{\mu\nu}P \,, \tag{2.1}$$

where $\varepsilon(x)$ is the energy density, $P(x)$ the pressure in the local rest frame, $u^{\mu}(x) = (\gamma, \gamma\mathbf{v})$ the local four velocity, normalized such that $u^{\mu}u_{\mu} = 1$, and $g^{\mu\nu} = \text{diag}(+,-,-,-)$ the metric. The quantity $x = (t, \mathbf{x})$ represents the space-time coordinates. The hydrodynamic equations are given by

$$\partial_{\mu}T^{\mu\nu} = S^{\nu} \,, \tag{2.2}$$

where S^{ν} is a source term. If the particle number is conserved, one has in addition to consider the continuity equation

$$\partial_{\mu}j^{\mu} = 0 \,, \tag{2.3}$$

with conserved particle current $j^\mu = nu^\mu$ and rest frame mass density n. In practice j^μ is evolved by using the four-velocity calculated from Eq. (2.2).

We define

$$\Phi^0 = T^{00} = E = \gamma^2(\varepsilon + P) - P \,, \tag{2.4}$$

$$\Phi^i = T^{0i} = M^i = \gamma^2(\varepsilon + P)v^i \quad (i = 1,2,3) \,, \tag{2.5}$$

$$j^0 = R = \gamma n \,, \tag{2.6}$$

where E, $\mathbf{M}$, and R denote the quantities for energy, momentum, and mass densities in any fixed frame. In terms of the four-vector $\Phi^\mu = (\Phi^0, \mathbf{\Phi})$, the equations of motion (2.2) become

$$\frac{\partial \Phi^\mu}{\partial t} = -\nabla \cdot \mathbf{F}^\mu[\Phi] + S^\mu \,, \tag{2.7}$$

where $\mathbf{F}^\mu[\Phi]$ is a functional tensor of Φ^μ. $\mathbf{F}^\mu$ is defined as

$$\begin{pmatrix} \mathbf{F}^0 \\ \mathbf{F}^i \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi} \\ (\Phi^i \mathbf{\Phi})/(\Phi^0 + P) - \mathbf{g}^i P \end{pmatrix} \,, \tag{2.8}$$

where $i = 1,2,3$, and $\mathbf{g}^i$ is a tensor whose components are given by $g^{ii} = -1$, $g_{i \neq j} = 0$. In terms of Φ^μ, the rest frame energy density is

$$\varepsilon = \Phi^0 - \frac{\mathbf{\Phi}^2}{\Phi^0 + P} \,. \tag{2.9}$$

The set of equations [(2.2) and (2.3)] is closed by specifying the equation of state given in the form $P = P(\varepsilon, n)$. For simplicity we use an ideal gas equation of state which is given by

$$P = (\Gamma - 1)(\varepsilon - m) \,, \tag{2.10}$$

where Γ is the adiabatic index. We assume the ultrarelativistic limit such that $\varepsilon \gg n$, so that

$$P = \tfrac{1}{3}\varepsilon \tag{2.11}$$

is the resulting equation of state. Note that this decouples Eqs. (2.2) and (2.3). Thus we will consider only the quantities of Eq. (2.2) in our comparisons. A more realistic calculation would require a more complex equation of state [14]; however, for this numerical comparison we will use this simple choice.

III. METHODS FOR SOLVING THE RELATIVISTIC HYDRODYNAMIC EQUATIONS

A. Basis-spline collocation method

In this section we will discuss the basis-spline method for solving boundary-value differential equations, and we will show how to apply this method to the relativistic hydrodynamic conservation equations. It is important to investigate the properties of solutions obtained via higher-order methods. The lattice solution of differential equations on a discretized mesh of independent variables proceeds in two steps: (1) obtain a discrete representation of the functions and operators on the lattice, and (2) solve the resulting lattice equations using iterative techniques. Step (1) is an interpolation problem for which we take advantage of spline function techniques [12,15,16]. The use of the spline-collocation method leads to a matrix-vector representation on the collocation lattice with a metric describing the transformation properties of the lattice, as described below. Thus modern computational techniques for solving linear algebraic systems are directly applicable.

1. Time evolution of the conservation equations

The equations of motion for the fluid are given in Eq. (2.7). A small viscous source term $S^\mu(x)$ is also added to Eq. (2.7) to increase numerical stability. In a system where there is no viscosity, conserved quantities are given by

$$Q^\mu(t) = \int d^3x \, \Phi^\mu(t) \,. \tag{3.1}$$

Effects of the numerical viscosity term in Eq. (2.7) can be removed from the observables in the following way. During the course of the calculation, the numerical viscosity term remains small when compared with conserved quantities. Thus it may be treated as a perturbation on the system. Assuming that the source term contains only the numerical viscosity, we may derive a conserved four current at time t from Eq. (2.7)

$$\tilde{Q}^\mu(t) = Q^\mu(t) - \int d^3x \int_{t_0}^{t} d\tau \, S^\mu(\tau) \,, \tag{3.2}$$

such that

$$\frac{d\tilde{Q}^\mu}{dt} = 0 \,. \tag{3.3}$$

The modification to Q^μ is the history of the viscous source term, in the form of an integral over time, from initial time t_0. Computationally, these conservation equations hold to the level of accuracy we require in the calculation.

The set of conservation equations (2.7) may be cast in the form of matrix equations for the fields Φ^μ represented by their values $\Phi^\mu = \{\Phi^\mu_\alpha\}$ on a lattice of points $\mathbf{r}_\alpha$. We will employ the basis-spline collocation method on a Cartesian lattice, as explained below. The index α is now a shorthand notation for $(\alpha_x, \alpha_y, \alpha_z)$. In the collocation method, local functions are represented by their values at the collocation points. For example, $\mathbf{F} = \{F(\Phi)_\alpha\}$ is given by $\{F(\Phi_\alpha)\}$, thus making it easy to handle complex nonlinearities. Derivatives with respect to x, y, z are represented by matrices $\mathbf{D}_a$ where $a = 1,2,3$. The discrete form of Eq. (2.7) is then given by

$$\frac{\partial \Phi^\mu}{\partial t} = -\sum_{a=1}^{3} \mathbf{D}_a \mathbf{F}^{a\mu}[\Phi] + S^\mu \,. \tag{3.4}$$

The viscosity term is given by

$$S^{\mu} = \nu \sum_{a=1}^{3} \mathbf{D}_a^2 \Phi^{\mu} , \tag{3.5}$$

where ν is the numerical viscosity.

The time propagation is achieved through a nonlinear iteration of Eq. (2.7). We rewrite the conservation equations in a shorthand notation as

$$\frac{\partial \Phi}{\partial t} = \mathcal{F}[\Phi(t)] . \tag{3.6}$$

Consider the integral of Eq. (3.6) over a small, but finite time step Δt,

$$\int_{\Phi(t_0)}^{\Phi(t_0+\Delta t)} d\Phi = \int_{t_0}^{t_0+\Delta t} \mathcal{F}[\Phi(t)] dt . \tag{3.7}$$

Time is discretized in the sense that $t = m\Delta t + t_0$, where $m = 0, 1, \ldots$. We perform the integral in Eq. (3.7) to obtain

$$\Phi(t_0 + \Delta t) = \Phi(t_0) + \Delta t \mathcal{F}[\bar{\Phi}(t_0)] , \tag{3.8}$$

where we assume that $\mathcal{F}[\bar{\Phi}(t_0)]$ is a constant functional evaluated at the average value of the field $\bar{\Phi}(t_0)$ over the time step Δt.

We now consider the time-evolution algorithm for our hybrid space-time lattice, where we may rewrite Eq. (3.8) as

$$\Phi_{m+1,\alpha} = \Phi_{m,\alpha} + \Delta t \mathcal{F}[\bar{\Phi}_{m,\alpha}] , \tag{3.9}$$

where the average is

$$\bar{\Phi}_{m,\alpha} \equiv \theta_{m,\alpha} \Phi_{m+1,\alpha} + (1 - \theta_{m,\alpha}) \Phi_{m,\alpha} , \tag{3.10}$$

and

$$\theta_{m,\alpha} = \delta(m + \alpha, \text{even}) . \tag{3.11}$$

Equation (3.9) describes a forward finite-difference approximation to Eq. (3.6). However, this simple formula is an implicit equation because of the definition of the average field $\bar{\Phi}$ over the time step in Eq. (3.10). Using the definition Eq. (3.11) in Eq. (3.10) is known as hopscotch averaging [17] which divides the lattice into odd and even spatial points. This averaging $\bar{\Phi}_{m,\alpha}$ is defined, for example, by combining the odd points from $\Phi_{m+1,\alpha}$ with the even points from $\Phi_{m,\alpha}$. In two or three dimensions, a checkerboard pattern is used. Note that if we set $\theta_{m,\alpha} = 1$ for all space points, then the evolution scheme would be forward Euler; $\theta_{m,\alpha} = 0$ gives backward Euler, and $\theta_{m,\alpha} = 1/2$ results in Crank-Nicholson evolution.

We solve the implicit equation [Eq. (3.9)] using a two-step iterative process. In constructing the first step, we begin with a guess for the value of the field $\bar{\Phi}_{m,\alpha}$ by

$$\bar{\Phi}^k_{m,\alpha} \equiv \theta_{m,\alpha} \Phi^k_{m+1,\alpha} + (1 - \theta_{m,\alpha}) \Phi_{m,\alpha} , \tag{3.12}$$

where k is the iteration index which has values $k = 0, 1, \ldots$. The initial condition for the iterate is $\Phi^{k=0}_{m+1,\alpha} = \Phi_{m,\alpha}$, so that the initial value of the average field is also $\bar{\Phi}^{k=0}_{m,\alpha} = \Phi_{m,\alpha}$. This guess allows the computation of $\Phi^{k+1}_{m+1,\alpha}$ using Eq. (3.9) in the form

$$\Phi^{k+1}_{m+1,\alpha} = \Phi_{m,\alpha} + \Delta t \mathcal{F}[\bar{\Phi}^k_{m,\alpha}] . \tag{3.13}$$

In the second step of our iterative solution, we simply average the present iterate with the iterate before the previous one

$$\Phi^{k+1}_{m+1,\alpha} \leftarrow \tfrac{1}{2}(\Phi^{k-1}_{m+1,\alpha} + \Phi^{k+1}_{m+1,\alpha}) . \tag{3.14}$$

We then compute the integrated residual between $\Phi^k_{m+1,\alpha}$ and $\Phi^{k+1}_{m+1,\alpha}$, and iterate until the total residual falls below a prescribed threshold, usually 10^{-10} for 64-bit precision, or 10^{-5} for 32-bit precision. The total residual is given by

$$\rho_{m,k} = \sum_{\alpha} \left[\Phi^k_{m+1,\alpha} - \Phi^{k+1}_{m+1,\alpha} \right] . \tag{3.15}$$

This scheme damps out instabilities in which the odd and even points propagate independently. Usually less than ten iterations are needed to obtain convergence in Φ^{μ}.

2. *Splines*

Given a set of points or *knots* denoted by the set $\{x_i\}$, a spline function of order M, denoted by B_i^M, is constructed from continuous piecewise polynomials of order $(M - 1)$. These splines have continuous derivatives up to order $(M - 2)$. We only consider odd-order splines or even-order polynomials for reasons related to the choice of the collocation points. The ith spline is nonzero only in the interval (x_i, x_{i+M}). This property is commonly referred to as limited support. The knots are the points where polynomials making up the spline join. In the interval containing the tail region, the splines fall rapidly to zero. The explicit construction of the splines is explained elsewhere [15]. We can also construct exact derivatives of splines provided the derivative order does not exceed $(M - 1)$.

A continuous function $f(x)$, defined in the interval $(x_{\min}, x_{\max})$, can be expanded in terms of spline functions as

$$f(x) = \sum_i B_i^M(x) c^i , \tag{3.16}$$

where the quantities c^i denote the expansion coefficients. We can solve for the expansion coefficients in terms of a given, or to be determined, set of function values evaluated at a set of data points, more commonly known as *collocation points.* There are a number of ways to choose collocation points [15,16]; however, for odd-order splines, a simple choice is to place one collocation point at the center of each knot interval within the physical boundaries

$$x_{\alpha} = \frac{x_{\alpha+M-1} + x_{\alpha+M}}{2} , \quad \alpha = 1, \ldots, N . \tag{3.17}$$

Here, $x_M = x_{\min}$, $x_{N+M} = x_{\max}$, and N is the number of collocation points. Note that collocation points are denoted by Greek subscripts. We can now write a linear system of equations by evaluating Eq. (3.16) at these collocation points

$$f_\alpha = \sum_i B_{\alpha i} c^i \,, \tag{3.18}$$

where $f_\alpha \equiv f(x_\alpha)$ and $B_{\alpha i} \equiv B_i^M(x_\alpha)$. In order to solve for the expansion coefficients, the matrix $\mathbf{B}$ needs to be inverted; however, as it stands, the matrix $\mathbf{B}$ is not a square matrix, since the total number of splines with a nonzero extension in the physical region is $N + M - 1$. In order to perform the inversion, we need to introduce additional linear equations which represent the boundary conditions imposed on $f(x)$ at the two boundary points, x_M and x_{M+N}. The essence of the lattice method is to eliminate the expansion coefficients c^i using this inverse matrix. The details are discussed elsewhere [15]. Following the inversion, the coefficients are given by

$$c^i = \sum_\alpha \left(\mathbf{B}^{-1}\right)^{i\alpha} f_\alpha \,. \tag{3.19}$$

One can trivially show that all local functions will have a local representation in the finite-dimensional collocation space

$$L(x) \approx L(x_\alpha) \,, \quad x_i < x \le x_{i+M} \,. \tag{3.20}$$

The collocation representation of the operators can be obtained by considering the action of an operator $\mathcal{O}$ on a function $f(x)$

$$\mathcal{O}f(x) = \sum_i \mathcal{O}B_i^M(x) c^i \,. \tag{3.21}$$

If we evaluate the above expression at the collocation points x_α, we can write

$$\mathcal{O}f_\alpha = \sum_i \mathcal{O}B_{\alpha i} c^i \,. \tag{3.22}$$

Substituting from Eq. (3.19) for the coefficients c^i, we obtain

$$\begin{aligned} \mathcal{O}f_\alpha &= \sum_{i\beta} \mathcal{O}B_{\alpha i} \left(\mathbf{B}^{-1}\right)^{i\beta} f_\beta \\ &= \sum_\beta O_\alpha^\beta f_\beta \,, \end{aligned} \tag{3.23}$$

where we have defined the collocation space-matrix representation of the operator $\mathcal{O}$ by

$$O_\alpha^\beta = \sum_i \mathcal{O}B_{\alpha i} \left(\mathbf{B}^{-1}\right)^{i\beta} \,. \tag{3.24}$$

Note that the construction of the collocation space operators can be performed once and for all at the beginning of a calculation, using only the given knot sequence and collocation points. Due to the presence of the inverse in Eq. (3.24), the matrix O is not sparse. In practice, the operator $\mathcal{O}$ is chosen to be a differential operator such as ∇_x or ∇_x^2. By a similar construction, it is also possible to obtain the appropriate integration weights on the collocation lattice [15].

B. The SHASTA Method

SHASTA is an example of an algorithm that uses a stable but diffusive differencing scheme. We restrict ourselves to a brief overview of the principles of the SHASTA; the details can be found in the original papers by Boris and Book [8–10]. Since there are different variants of this algorithm, we describe below the exact version actually used.

The essential idea of the FCT method is to increase the stability of an arbitrary differencing scheme by introducing a corrective nonlinear diffusion step. This numerical dissipation is of conservative form. In a succeeding antidiffusion step, this diffusion is removed partially (flux limiter) to avoid spurious oscillations while retaining sharp profiles in cases of discontinuities or steep gradients. This numerical diffusion can either be introduced externally by means of an artificial viscosity term or can be inherent in the applied differencing scheme.

The basic algorithm is applicable to any one-dimensional equation in the conservative form

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(f v_x) = S(x,t) \tag{3.25}$$

describing the evolution of a conserved density f.

Assume that the values f_i, v_i, and S_i for the density, velocity, and source are given at grid point numbers i at the beginning of the time step. The grid is evenly spaced with a distance of Δx and the time step is Δt. Our version of SHASTA in detail proceeds as follows.

(1) Compute dimensionless velocities κ_i and numerical diffusion coefficients ν_i at the midpoints:

$$\kappa_{i+1/2} = v_{i+1/2} \frac{\Delta t}{\Delta x} \,, \tag{3.26}$$

$$\nu_{i+1/2} = \tfrac{1}{2}\left(\tfrac{1}{4} + \kappa_{i+1/2}^2\right) . \tag{3.27}$$

(2) Calculate the diffused untransported solution F^D, which does not contain the effects of the flow velocity but only those of the artificial viscosity. It is obtained using purely diffusive fluxes f^0:

$$f_{i+1/2}^0 = \nu_{i+1/2}(f_{i+1} - f_i) \,, \tag{3.28}$$

$$F_i^D = f_i + (f_{i+1/2}^0 - f_{i-1/2}^0) \,. \tag{3.29}$$

(3) Calculate the transport convection factors Q^+ and Q^-, which are motivated by a simple geometric picture of transport and redistribution into space-fixed cells [8]. This also contains diffusion in an implicit way, which can be approximated by the diffusion coefficients given above

$$Q_i^+ = \left[\frac{1}{2} - v_i \frac{\Delta t}{\Delta x}\right] \Bigg/ \left[1 + \frac{\Delta t}{\Delta x}(v_{i+1} - v_i)\right] \,, \tag{3.30}$$

$$Q_{i+1}^- = 1 - Q_i^+ \,. \tag{3.31}$$

(4) Applying these factors leads to the transported diffused solution F_i^{TD}:

$$F_i^{TD} = 4\,Q_i^{+2} f_{i+1/2}^0 - 4\,Q_i^{-2} f_{i-1/2}^0 + (Q_i^+ + Q_i^-) f_i + S\Delta t\,. \tag{3.32}$$

We now construct fluxes containing the transport effects only by subtracting the purely diffusive ones from the transport-diffused ones, yielding the intermediate fluxes δF^{TD} and f^T,

$$\delta F_{i+1/2}^{TD} = F_{i+1}^{TD} - F_i^{TD}\,, \tag{3.33}$$

$$f_{i+1/2}^T = f_{i+1/2}^0 + \nu_{i+1/2}(F_{i+1}^{TD} - F_{i+1}^D - F_i^{TD} + F_i^D)\,. \tag{3.34}$$

(5) Finally the antidiffusion step is performed: the diffusive fluxes are removed from the transport-diffused ones except where this would lead to the onset of instability in the guise of accentuation of existing extrema or creation of new ones. This leads to the standard FCT formulas

$$f_{i+1/2}^C = \mathrm{sgn}(f_{i+1/2}^T)\max\{0, \min[\mathrm{sgn}(f_{i+1/2}^T, \delta F_{i-1/2}^{TD}, |f_{i+1/2}^T|k), \mathrm{sgn}(f_{i+1/2}^T)\delta F_{i+3/2}^{TD}]\}\,, \tag{3.35}$$

$$\tilde{f}_i = F_i^{TD} - (f_{i+1/2}^C - f_{i-1/2}^C)\,, \tag{3.36}$$

yielding the values $\tilde{f}$ at the end of the time step with the help of the corrected fluxes $f_{i+1/2}^C$.

The one-dimensional algorithm may easily be extended to three dimensions using operator splitting methods.

The relativistic generalization of the SHASTA differs from the nonrelativistic method in only two aspects. In order to calculate the pressure, rest frame energy, and number densities, we need to solve Eqs. (2.1)–(2.6) for n, ε, and the flow velocity $\mathbf{v}$ in terms of the fields E, R, and $\mathbf{M}$. The solution is obtained from a single fixed-point equation [18]. Simple rearrangements yield

$$v^2 = \frac{M^2}{(E+P)^2} = f(v^2)\,,$$

with

$$P(\varepsilon, n) = P(E - M\sqrt{v^2}\,,\ R\sqrt{1-v^2})\,, \tag{3.37}$$

leading to the velocity components

$$v_i = \frac{M_i}{(E+P)}\,, \tag{3.38}$$

and

$$\varepsilon = E - \mathbf{M}\cdot\mathbf{v}\,, \quad n = \frac{R}{\gamma}\,. \tag{3.39}$$

The solution of this problem is a major part of the numerical effort and is a crucial ingredient of the relativistic SHASTA algorithm.

In addition, it is necessary to avoid the generation of cells with $R > E$ or $M > E$. This can be done by a modification of the flux limiting, e.g., by not carrying out the antidiffusion step in the respective cells. The simplest possibility is to readjust E to fulfill the conditions $E \geq M$ and $E \geq R$. The violation of the conservation laws introduced by these modifications is found to be of the order of the numerical accuracy.

IV. COMPARISON OF THE TWO METHODS

A. Comparisons for a specific collision

For comparative purposes, we have performed collision calculations with the two methods previously described. The calculations were performed for the one fluid system that models the initial state in the ^{28}Si+^{238}U nuclear system, each with a relativistic $\gamma = 10$ in the equal velocity frame. Rest frame initial conditions in both calculations were taken to be that of a Woods-Saxon density distribution

$$\varepsilon(r) = \frac{\varepsilon_0}{1 + \exp[(r - r_0)/a_0]}\,, \tag{4.1}$$

where $a_0 = 0.5$ fm is the nuclear skin thickness, r_0 is the average nuclear radius calculated from the number of nucleons N as

$$r_0 = 1.128N^{1/3} - 0.89N^{-1/3}\,, \tag{4.2}$$

and $\varepsilon_0 = 0.156$ GeV/fm^{-3} is the initial rest-frame nuclear energy density. Initially all Φ fields and currents are known. We stress that this is a schematic collision, as we are only using a one-fluid model with a particularly simple equation of state, and we have not included any current source terms. A more realistic calculation would include a realistic nuclear equation of state and, eventually, a color current term (in the form of a source), and possibly the concept of interpenetrating fluids [19,20].

In each calculation, the nuclei were positioned in the x, z collision plane with z being the collision axis. The rest frame energy density profile of each nucleus was boosted to the equal velocity frame with a Lorentz $\gamma = 10$. The centers of the nuclei were positioned at $z = \pm 1$ fm initially. We performed two studies, one with impact parameter $b = 0$ fm, and a second with impact parameter $b = 3$ fm. We compare the zero impact parameter collisions here, as results are similar in both cases. Surface plots of the laboratory density Φ^0 can be seen in Fig. 1, where the basis-spline calculations are shown in the left hand column, and SHASTA results are given in the right column at times $t = 0$ (top), $t = 1.6$ (middle), and $t = 3.2$ fm/c (bottom). The basis-spline method used a 32^3 mesh with $dx = dy = 1.0$ fm, $dz = 0.2$ fm, while the SHASTA method used a $n_x = n_y = 128$ (transverse), $n_z = 64$ (longitudinal) mesh, with $dx = dy = dz = 0.2$ fm. Although the results are very similar, a more detailed analysis indicates some differences at the level of ten percent. We show in Fig. 2 a cut along the

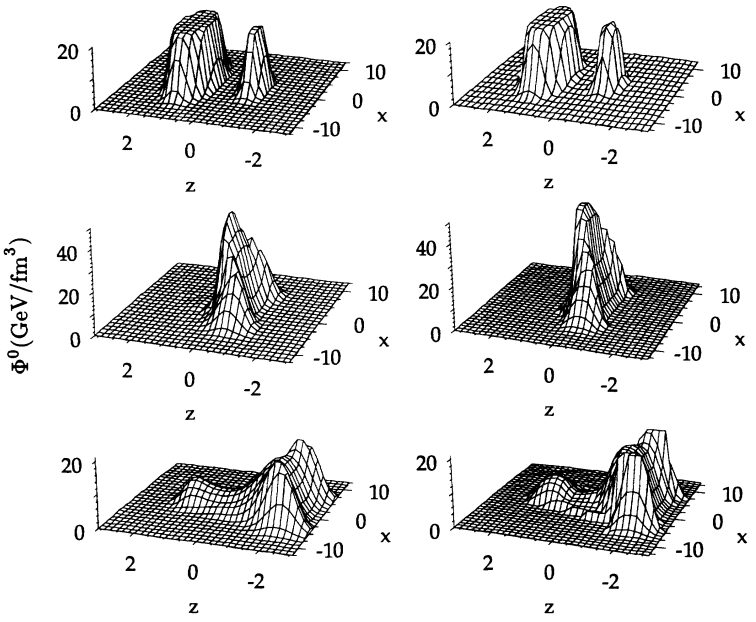

FIG. 1. We compare the time evolution of the scalar component Φ^0 for the two different numerical methods. The basis-spline calculation is shown in the left column, while SHASTA is given in the right column. Φ^0 is shown in the equal velocity frame at times $t = 0.0$ (top), $t = 1.6$ (middle), and $t = 3.2$ fm/c (bottom) for the zero impact parameter collision of Si+U at $\gamma = 10$ in the equal velocity frame.

z axis in the $x = y \approx 0.0$ plane at $t = 0.0$ (top), $t = 1.6$ (middle), and $t = 3.2$ fm/c (bottom). Differences seen in the initial conditions are due to small differences in the initial lattice setups. At $t = 1.6$ fm/c differences of a few percent are noticeable. Although similar behavior is seen in the final time step, one observes that the two methods have slightly different solutions at this stage of the calculation. We have also plotted in Fig. 3 a transverse cut, at $t = 3.2$ fm/c, at the local maximum of the target (residual U) distribution (left), and the projectile distribution (right). Differences in these distributions can be explained by the use of higher resolution in the SHASTA calculations.

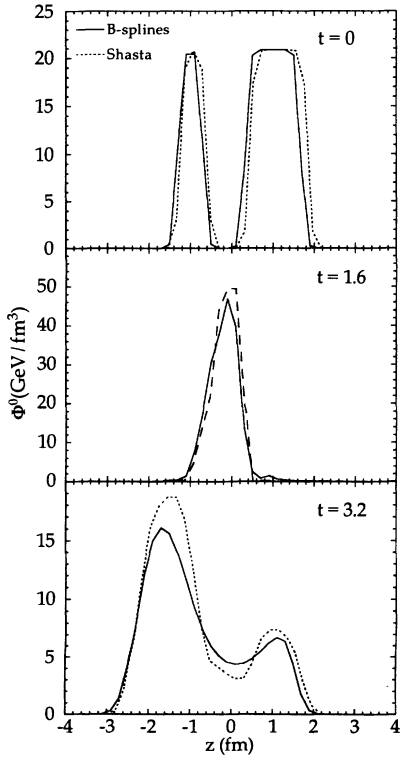

FIG. 2. We compare the time evolution of the scalar component Φ^0 along the collision axis (z axis) with $x = y = 0$. The different times are indicated in the figure.

We show in Fig. 4 the maximum rest frame energy density (right), and the maximum lab energy (left) as a function of the time. The maximum value attained shows a 7% difference. This maximum occurs at roughly 1.0 fm/c in both variables plotted. We have checked that these differences do not arise as a result of the initial conditions of the calculation.

In Fig. 5, we plot the longitudinal rapidity distributions of the laboratory energy density, $d\Phi^0/dy$ as a function of the rapidity variable, $y = 0.5\ln\left(\Phi^0 + \Phi^z\right) / \left(\Phi^0 - \Phi^z\right)$ for various times during the collision. It is obvious that in both cases the rapidity peak of the spectator fluid decreases as a function of time, but at differing rates. One also observes from Fig. 4 that in the final stages of the evolution the maximum laboratory energy density decreases more rapidly in the basis-spline method as compared to SHASTA. It should be noted that after $t = 2.0$ fm/c in Fig. 4 the maximum laboratory density resides in the residual nucleus peaks in both calculations. The SHASTA calculations use a higher resolution in the transverse direction than the basis-spline method. Thus the smearing of transverse shockwaves is larger for the latter method. This favors a rapid depletion of the target rapidity region as seen in Fig. 5 [20].

B. Computational comparisons

Calculations using the two methods described above have been performed on different computational plat-

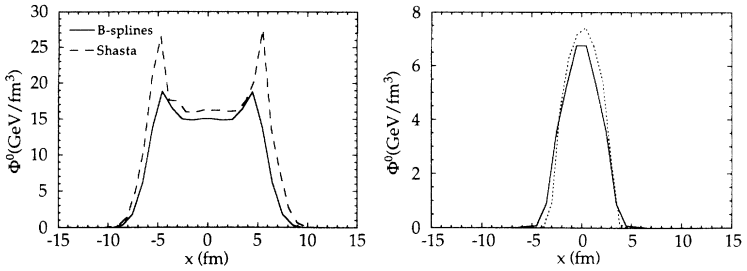

FIG. 3. We show a two dimensional cut for the Φ^0 distribution at the final time $t = 3.2$ fm/c. On the left is a transverse cut at the maximum of the residual U nucleus. On the right is a transverse cut through the residual of the Si nucleus.

forms. We briefly discuss the calculational timings and differences in this section.

The basis-spline calculations were performed on an Intel/i860 hypercube at Oak Ridge National Laboratory. The principal points concerning implementation of our hydrodynamic evolution code are as follows [7]. We perform all of our calculations on a lattice with N_x, N_y, and N_z collocation points in the x, y, and z directions, respectively. For the parallel implementation of the hydrodynamics code, it is necessary to partition the problem among the various processors. Suppose there are P processors. Our implementation requires that the x dimension is evenly distributed among the processors. Thus the vectors of each node are dimensioned $M_x = N_x/P$, N_y, N_z. In the hydrodynamics calculation approximately 40% of all operations are local to each processor. The remaining 60% of the computational effort we need to address are vector-matrix-multiply problems. One of the difficulties of present day parallel computing is the problem of node-to-node communication. The current Intel machines have central processors that are capable of 70 Mflops (maximum) for single precision operations, while the typical communications speed is roughly 1 Mbyte (iPSC/i860) of information passed per second. Thus efficient use of the parallelism is limited and depends on the frequency of message passing between nodes, and the length of messages passed. The basis-spline method includes the use of nonsparse derivative matrices, which requires that the entire local vector Φ be passed around the ring in a single calculation of the derivative. The inner loops of the $\nabla \cdot \mathbf{F}$ calculation described above perform at 16 MF on each node. Quantities that measure the overall code performance are the parallel speedup, $S(p)$, and the parallel efficiency, $\epsilon(p)$, defined as

$$S(p) = \frac{T(1)}{T(p)}\,, \quad \epsilon(p) = \frac{S(p)}{p}\,, \tag{4.3}$$

where $T(p)$ is the CPU time required to run the code on p nodes. For the 32-node calculation described in this work, the code speedup was $S(32) = 7.6$, and the code efficiency was $\varepsilon(32) = 0.24$. The total calculational time for the $N_x = N_y = N_z = 32$ run was 3.5 hours on a 32-node cube.

The SHASTA calculations were performed on an IBM 3090-600J VF at the Gesellschaft für Schwerionenforschung (GSI) in Darmstadt. One simulation Si + U with a grid size given above needs 120 minutes. For these calculations we used 120 Mbyte main storage.

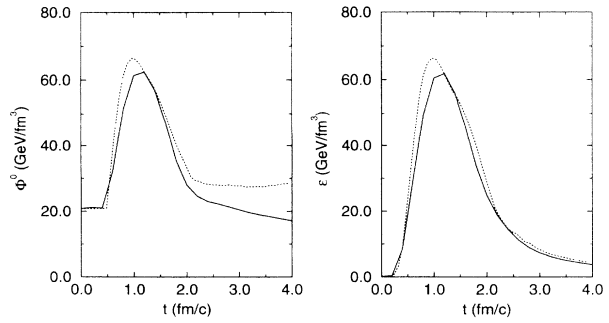

FIG. 4. Maximum rest frame energy density (right) and laboratory energy density (left) are shown as a function of time. Basis-spline results are given by the solid line, while SHASTA results are indicated by the dashed line.

V. CONCLUSIONS

In this paper we discussed and compared the basis-spline algorithm and the SHASTA finite difference technique for solving the relativistic hydrodynamic conservation equations. We performed a "nuclear" collision at zero impact parameter in the one fluid model. Considering that the two numerical techniques are based on totally different methods, and that we are examining an extreme hydrodynamic situation, we find the resulting agreement surprisingly good. The remaining differences are of the order of up to ten percent. For applications in heavy-ion collisions this is certainly sufficient in view of other uncertainties in the physical situation. Differences in the comparison may arise from the apparent differences in numerical dissipation in the two calculations.

We have used the very simple equation of state $P = \varepsilon/3$ in this one-fluid model. More complicated equations of state could be used to model the transition from hadronic matter to quark matter, and would include a mixed phase state. Calculations using these far more complicated (and more realistic) equations of state may lead to more pronounced differences when compared to each other. Investigations are currently under way to determine the effects of real viscosity during relativistic heavy-ion collisions. It has been suggested that the entropy can be measured as a signal of the equation of state. These calculations indicate that entropy production from numerical dissipation must be well understood before hydrodynamic codes can be used to investigate signals for phase transitions. In future work we will investigate the

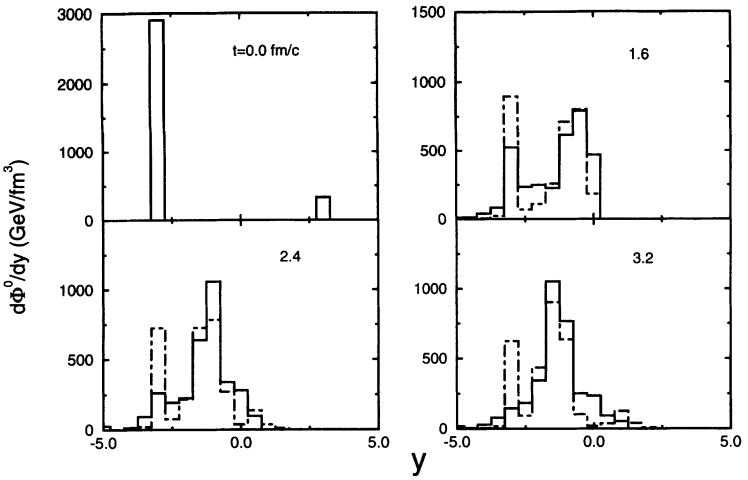

FIG. 5. The distribution of Φ^0 in the rapidity variable y is shown. The basis-spline results are given by the solid histogram, while the dashed line indicates SHASTA results.

numerical and physical dissipation of using more realistic equations of state.

ACKNOWLEDGMENTS

This research was sponsored in part by the U.S. National Science Foundation under Grant No. PHY90-13248, by the U.S. Department of Energy (DOE) Office of Scientific Computing under the High Performance Computing and Communications Program (HPCC), by DOE under Contract No. DE-AC05-84OR21400 managed by Martin Marietta Energy Systems, Inc., under Contract No. DE-FG05-87ER40376 with Vanderbilt University, and by the BMFT and GSI. The Joint Institute for Heavy-Ion Research has as member institutions the University of Tennessee, Vanderbilt University, and the Oak Ridge National Laboratory; it is supported by the members and by the Department of Energy through Contract No. DE-FG05-87ER40361 with the University of Tennessee. Numerical calculations were carried out on the ORNL iPSC/i860, and the Intel Touchstone Delta System operated by Caltech on behalf of the Concurrent Supercomputing Consortium. The SHASTA calculations were done on the GSI IBM 3090-600J VF in Darmstadt, Germany.

[1] G. Graebner, Ph.D. thesis, Frankfurt, 1985.
[2] M. L. Norman and K. A. Winkler, in *Astrophysical Radiation Hydrodynamics* (Reidel, Boston, 1986), p. 449.
[3] V. Schneider, U. Katscher, D.-H. Rischke, B. Waldhauser, J. A. Maruhn, and C.-D. Munz, J. Comput. Phys. **105**, 92 (1993).
[4] Y. Pürsün, Diploma thesis, University of Frankfurt, 1993; A. von Keitz, Ph.D. thesis, University of Frankfurt, 1993; U. Katscher, Ph.D. Thesis, University of Frankfurt, 1993.
[5] R. Cen, Ap. J. Suppl. Ser. **79**, 341 (1992); P. J. Mann, Comp. Phys. **67**, 245 (1991); P. A. Laguna, W. A. Miller, and W. H. Zurek, Ap. J. **404**, 678 (1993).
[6] D. J. Dean, C. Bottcher, and M. R. Strayer, Int. J. Mod. Phys. **C4**, 703 (1993).
[7] D. J. Dean, C. Bottcher, M. R. Strayer, and J. C. Wells, Int. J. Mod. Phys. **C4**, 1023 (1993).
[8] J. P. Boris and D. L. Book, J. Comput. Phys. **11**, 38 (1973).
[9] D. L. Book, J. P. Boris, and K. Hain, J. Comput. Phys. **18**, 248 (1975).
[10] J. P. Boris and D. L. Book, J. Comput. Phys. **20**, 397 (1976).
[11] R. Lohner, K. Morgan, M. Vahdati, J. P. Boris, and D. L. Book, Comm. Appl. Num. Meth. **4**, 717 (1988).
[12] C. Bottcher and M. R. Strayer, Ann. Phys. (N.Y.) **175**, 64 (1987).
[13] L. D. Landau, Izv. Akad. Nauk SSSR **17**, 51 (1953); S. Z. Belen'kii and L. D. Landau, Usp. Fiz. Nauk **56**, 309 (1955), reprinted in *Collected Papers of L. D. Landau*, edited by D. ter Haar (Gordon and Breach, New York, 1965), pp. 569 and 665.
[14] T. Ishii and S. Muroya, Phys. Rev. D **46**, 5156 (1992); M. Kataja, J. Letessier, P. V. Ruuskanen, and A. Tounsi, Z. Phys. C **55**, 153 (1992).
[15] A. S. Umar, J. Wu, M. R. Strayer, and C. Bottcher, J. Comput. Phys. **93**, 426 (1991).
[16] C. De Boor, *Practical Guide to Splines* (Springer-Verlag, New York, 1978).
[17] V. Vermuri and W. J. Karplus, *Digital Computer Treatment of Partial Differential Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1981), p. 162.
[18] B. Kämpfer (private communication).
[19] U. Katscher, D. H. Rischke, J. A. Maruhn, W. Greiner, I. N. Mishustin, and L. M. Satarov, Z. Phys. **A346**, 209 (1993).
[20] B. Waldhauser, D. H. Rischke, U. Katscher, J. A. Maruhn, H. Stöcker, and W. Greiner, Z. Phys. C **54**, 459 (1992).