LA-8908-MS

C.3

ORBEOS:

A Spherical, Fully Multidimensional Flux-Corrected Transport Hydrodynamics Code

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos. New Mexico 87545

UNITED STATES
DEPARTMENT OF ENERGY
CONTRACT W-7405-ENG. 36

LA-8908-MS
UC-32
Issued: July 1981

ORBEOS:
A Spherical, Fully Multidimensional Flux-Corrected Transport Hydrodynamics Code

Anthony J. Scannapieco

ORBEOS:

A SPHERICAL, FULLY MULTIDIMENSIONAL

FLUX-CORRECTED TRANSPORT HYDRODYNAMICS CODE

by

Anthony J. Scannapieco

ABSTRACT

ORBEOS is a two-dimensional (r-θ), Eulerian hydrodynamics code with tabular equation of state (EOS), that uses a fully multidimensional flux-corrected transport algorithm to solve the fluid equations. This flux-corrected transport (FCT) algorithm, developed by S. T. Zalesak, is Cartesian and accurate to any desired order in space and to second order in time. The algorithm used in ORBEOS was modified for spherical geometry and is accurate in space to fourth order. Although this algorithm has been used before to solve the continuity equation, it is used in ORBEOS for the first time as the basic algorithm in a full hydrocode. The physical equations solved in the code and their finite-difference representations are described, and it is shown why the fully multidimensional algorithm better represents incompressible flows and does not lead to virtual sources as does the older one-dimensional FCT algorithm using Strang-type time-step differencing to handle multidimensional problems. Several test problems are presented and discussed.

I. ORBEOS

Studies of fluid instabilities in imploding spherical systems made it obvious that a computer code that could follow the evolution of the instabilities into the far nonlinear regime and run in a reasonable time on the CDC 7600 computers (i.e., in less than 60 min of CPU time) was needed. We wrote a code that would specifically satisfy these two requirements. That code is ORBEOS.

ORBEOS is a two-dimensional, Eulerian hydrocode that utilizes tabular equations of state for real substances; it also has the capability of running with an ideal gas equation of state. The geometry of the code is spherical (i.e., r - θ) in which all fluid quantities are independent of the azimuthal angle ϕ. This translates into a system in which the z-axis is the axis of rotational symmetry, as shown in Fig. 1. All fluid quantities depend only on r and θ. The physics contained in the code is that of a compressible fluid. (Later, more physics was introduced; however, later versions of the code and the physics contained therein will not be discussed in this report.) This choice of limiting the physics was made to isolate the basic aspects of the fluid instabilities without the complications introduced by a multitude of physical mechanisms.

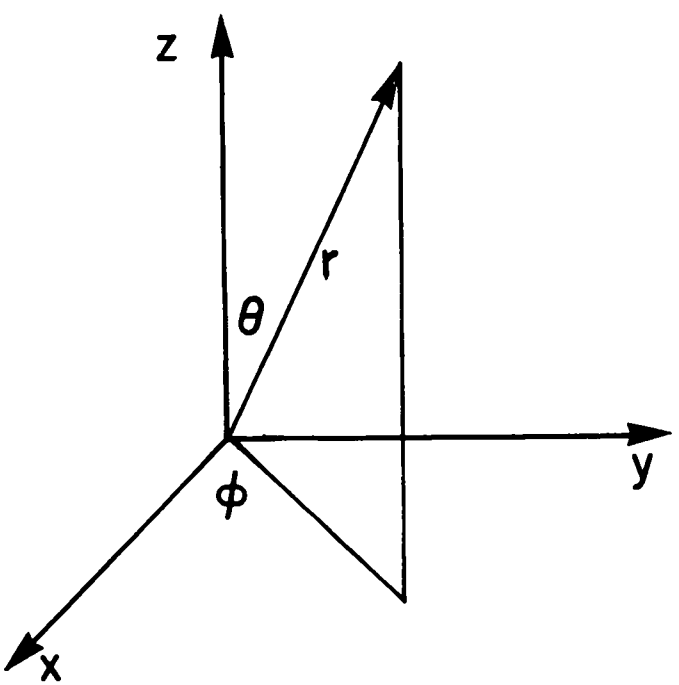

Fig. 1. Geometry of ORBEOS. All fluid quantities are independent of ϕ.

The physical equations chosen for study with the code are, in component form,

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho U_r\right) + \frac{1}{r \sin\Theta}\frac{\partial}{\partial \Theta}\left(\rho U_\theta \sin\Theta\right) = 0 \quad , \tag{1}$$

$$\frac{\partial \rho U_r}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho U_r U_r\right) + \frac{1}{r \sin\Theta}\frac{\partial}{\partial \Theta}\left(\rho U_r U_\theta \sin\Theta\right) - \rho\frac{U_\theta U_\theta}{r} = -\frac{\partial P}{\partial r} \quad , \tag{2}$$

$$\frac{\partial \rho U_\theta}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho U_\theta U_r\right) + \frac{1}{r \sin\Theta}\frac{\partial}{\partial \Theta}\left(\rho U_\theta U_\theta \sin\Theta\right) + \rho\frac{U_\theta U_r}{r} = -\frac{1}{r}\frac{\partial P}{\partial \Theta} \quad , \tag{3}$$

$$\frac{\partial T}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 U_r T\right) + \frac{1}{r \sin\Theta}\frac{\partial}{\partial \Theta}\left(U_\theta T \sin\Theta\right) =$$

$$-\frac{T}{C_v \rho}\left[\left(\frac{\partial P}{\partial T}\right)_\rho - \left(\frac{\partial \varepsilon}{\partial T}\right)_\rho \rho\right]\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 T_r\right) + \frac{1}{r \sin\Theta}\frac{\partial}{\partial \Theta}\left(U_\theta \sin\Theta\right)\right] \quad , \tag{4}$$

$$P = P(\rho, T) \quad , \tag{5}$$

$$\varepsilon = \varepsilon(\rho, T) \quad , \tag{6}$$

$$C_v = \left(\frac{\partial \varepsilon}{\partial T}\right)_\rho \quad , \tag{7}$$

where

ρ = fluid density ,
U_r = radial fluid velocity ,
U_θ = angular fluid velocity ,
T = temperature ,
P = scalar pressure ,
ε = specific energy , and
C_v = specific heat at constant density .

Equation (1), above, is the equation for mass conservation, Eqs. (2) and (3) are the equations of radial and angular momentum density transport, and Eq. (4) is the energy equation written in terms of the fluid temperature.

Equations (5), (6), and (7) are equations-of-state relations that, for ORBEOS, are obtained from tables compiled for real substances.

For a discrete variable representation of the above system of equations see Fig. 2, which is the finite grid representation of the continuous geometry shown in Fig. 1. Note that the discrete geometry is two-dimensional (i.e., r - Θ) because of the symmetry of the system with respect to ϕ. Figure 2 shows the indexing conventions used in ORBEOS. The computational mesh is uniformly spaced in both r and Θ with mesh spacings of Δr in radius, and $\Delta\Theta$ in angle so that

$$r_I = \Delta r*(I - 1) \quad , \tag{8}$$

and

$$\Theta_J = \Delta\Theta*(J - 1) \quad . \tag{9}$$

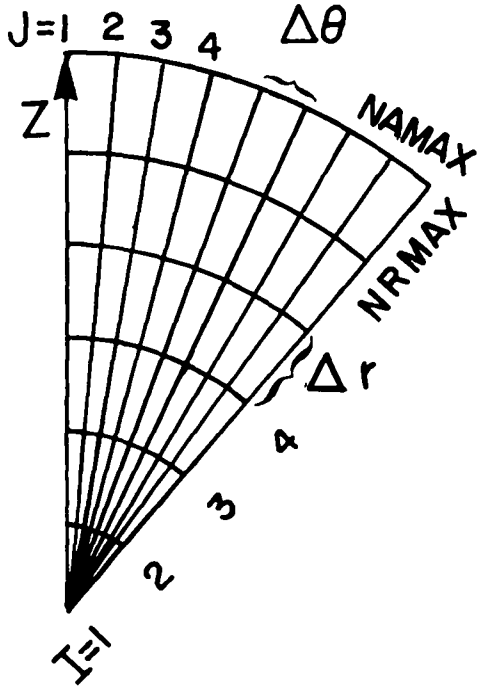

Fig. 2. ORBEOS finite grid representation of continuous geometry shown in Fig. 1.

A crucial aspect of the code is its boundary conditions.

ORBEOS's boundary conditions were chosen to best represent the physical systems in which we are interested.

Along the rotational axis of symmetry (J = 1, I = 1, NRMAX) the boundary conditions are "physical"; they are

(1) symmetric in ρ, T, and ρU_r,

$$\rho(I,-1) = \rho(I,2)$$

$$T(I,-1) = T(I,2)$$

$$\rho U_r(I,-1) = \rho U_r(I,2) \quad ;$$

and (2) reflecting in ρU_θ,

$$\rho U_\theta(I,-1) = -\rho U_\theta(I,2) \quad , \quad \rho U_\theta(I,1) = 0 \quad .$$

Along the surface of maximum Θ(J = NAMAX; I = 1, NRMAX) the boundary conditions are reflecting; they are

(1) symmetric in ρ, T, and ρU_r,

$$\rho(I,\ NAMAX+1) = \rho(I,\ NAMAX-1)$$

$$T(I,\ NAMAX+1) = T(I,NAMAX-1)$$

$$\rho U_r(I,\ NAMAX+1) = \rho U_r(I,NAMAX-1) \ ;$$

and (2) reflecting in ρU_θ,

$$\rho U_\theta(I,\ NAMAX+1) = -\rho U_\theta(I,\ NAMAX-1) \quad , \quad \rho U_\theta(I,\ NAMAX) = 0 \ .$$

At the origin point (J = 1, NAMAX; I = 1) the boundary conditions are "physical"; they are

(1) symmetric in ρ, and T,

$$\rho(-1,J) = \rho(2,J)$$

$$T(-1,J) = T(2,J) \quad ;$$

(2) reflecting in ρU_r,

$$\rho U_r(-1,J) = -\rho U_r(2,J) \quad ; \quad \rho U_r(1,J) = 0 \ ;$$

and (3) $\rho U_\theta(1,J) = 0$.

Along the mesh points (I = 1, J = 1, NAMAX), all fluid quantities of a given type (e.g., density, temperature, momentum density) are set equal to the average value of that particular quantity for all indices representing the origin point.

Along the surface of maximum r (J = 1, NAMAX; I = NRMAX), two types of boundary conditions are possible, either reflecting or transmitting. The reflecting boundary conditions are

(1) symmetric in ρ, T, and ρU_θ ,

$$\rho(\text{NRMAX+1, J}) = \rho(\text{NRMAX-1, J})$$

$$T(\text{NRMAX+1, J}) = T(\text{NRMAX-1, J})$$

$$\rho U_\theta(\text{NRMAX+1, J}) = \rho U_\theta(\text{NRMAX-1, J}) \ ;$$

and (2) reflecting in ρU_r ;

$$\rho U_r(\text{NRMAX+1, J}) = -\rho U_r(\text{NRMAX-1, J}) \quad , \quad \rho U_r(\text{NRMAX, J}) = 0 \ .$$

The transmitting boundary conditions are

(1) $\rho(\text{NRMAX+1, J}) = \rho(\text{NRMAX-1, J})$

(2) $T(\text{NRMAX+1, J}) = T(\text{NRMAX-1, J})$

(3) $\rho U_r(\text{NRMAX+1, J}) = \rho U_r(\text{NRMAX-1, J})$

and (4) $\rho U_\theta(\text{NRMAX+1, J}) = \rho U_\theta(\text{NRMAX-1, J})$.

These transmitting boundary conditions in reality represent "no-force" boundary conditions along the surface of maximum r. This choice of transmitting boundary is not the only choice possible in a Eulerian code.

The designations "physical", "reflecting", and "transmitting" associated with a boundary are better understood if we realize that "physical" indicates that the geometrical symmetries dictate the boundary conditions. For example, along the axis of rotational symmetry and at the origin, the geometry requires that the boundary conditions be given as above. However, the reflecting and transmitting boundary conditions at the surfaces of maximum Θ and maximum r are somewhat artificial. Only for $\Theta_{max} = \pi/2$ is the boundary condition on the surface of maximum Θ a physical boundary condition; at any other value of Θ_{max} the boundary conditions on that surface are only an approximation to reality;

the boundary condition on the surface of maximum r can only be chosen to best represent the physical problem under investigation.

To obtain the discrete variable representation of Eqs. (1) through (7) an integration of each equation over the volume element

$$\delta V_{ij} = 2\pi \int_{r_{i-1/2}}^{r_{i+1/2}} \int_{\Theta_{j-1/2}}^{\Theta_{j+1/2}} r^2 \sin\Theta \, d\Theta \, dr \, , \tag{10}$$

where

$$r_{i+1/2} = (r_{i+1} + r_i)/2 \quad , \tag{11}$$

and

$$\Theta_{j+1/2} = (\Theta_{j+1} + \Theta_j)/2 \tag{12}$$

are carried out; each result is then divided by δV_{ij}. Along the axis of rotational symmetry and at the origin of coordinates, the divergence theorem,

$$\oint_S \underset{\sim}{A} \cdot d\underset{\sim}{S} = \int_V \underset{\sim}{\nabla} \cdot \underset{\sim}{A} \, dV \quad , \tag{13}$$

was used.

If these integrations over the volume element, above, are carried out for each of Eqs. (1) through (4), the following set of finite-difference equations is obtained.

$$\frac{\partial \rho_{ij}}{\partial t} + \frac{3}{(r_{i+1/2}^3 - r_{i-1/2}^3)} \left[(r^2 \rho U_r)_{i+1/2,j} - (r^2 \rho U_r)_{i-1/2,j} \right]$$

$$+ \frac{3(r_{i+1/2}^2 - r_{i-1/2}^2)\left[(\rho U_\theta \sin\Theta)_{i,j+1/2} - (\rho U_\theta \sin\Theta)_{i,j-1/2} \right]}{2(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\Theta_{j-1/2} - \cos\Theta_{j+1/2})} = 0 \, , \tag{14}$$

$$\frac{\partial(\rho U_r)_{ij}}{\partial t} + \frac{3}{(r_{i+1/2}^3 - r_{i-1/2}^3)}\left[(r^2\rho U_r U_r)_{i+1/2,j} - (r^2\rho U_r U_r)_{i-1/2,j}\right]$$

$$+ \frac{3(r_{i+1/2}^2 - r_{i-1/2}^2)\left[(\rho U_r U_\theta \sin\Theta)_{i,j+1/2} - (\rho U_r U_\theta \sin\Theta)_{i,j-1/2}\right]}{2(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\Theta_{j-1/2} - \cos\Theta_{j+1/2})}$$

$$- \frac{(\rho U_\theta U_\theta)}{r_i}_{i,j} + \frac{(P_{i+1/2,j} - P_{i-1/2,j})}{(r_{i+1/2} - r_{i-1/2})} = 0 \quad , \tag{15}$$

$$\frac{\partial(\rho U_\theta)_{ij}}{\partial t} + \frac{3}{(r_{i+1/2}^3 - r_{i-1/2}^3)}\left[(r^2\rho U_\theta U_r)_{i+1/2,j} - (r^2\rho U_\theta U_r)_{i-1/2,j}\right]$$

$$+ \frac{3(r_{i+1/2}^2 - r_{i-1/2}^2)\left[(\rho U_\theta U_\theta \sin\Theta)_{i,j+1/2} - (\rho U_\theta U_\theta \sin\Theta)_{i,j-1/2}\right]}{2(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\Theta_{j-1/2} - \cos\Theta_{j+1/2})}$$

$$+ \frac{(\rho U_\theta U_r)}{r_i}_{i,j} + \frac{(P_{i,j+1/2} - P_{i,j-1/2})}{r_i(\Theta_{j+1/2} - \Theta_{j-1/2})} = 0 \quad , \tag{16}$$

$$\frac{\partial T_{i,j}}{\partial t} + \frac{3}{(r_{i+1/2}^3 - r_{i-1/2}^3)}\left[(r^2 T U_r)_{i+1/2,j} - (r^2 T U_r)_{i-1/2,j}\right]$$

$$+ \frac{3(r_{i+1/2}^2 - r_{i-1/2}^2)\left[(T U_\theta \sin\Theta)_{i,j+1/2} - (T U_\theta \sin\Theta)_{i,j-1/2}\right]}{2(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\Theta_{j-1/2} - \cos\Theta_{j+1/2})}$$

$$+\frac{3[P_{i,j} - C_{v_{i,j}}\rho_{i,j}T_{i,j} - (\partial P/\partial T)_{i,j}\rho_{i,j}^2]}{C_{v_{i,j}}\rho_{i,j}(r_{i+1/2}^3 - r_{i-1/2}^3)}$$

$$\times [(r_{i+1/2}^2 U_{r_{i+1/2,j}} - r_{i-1/2}^2 U_{r_{i-1/2,j}}) + (r_{i+1/2}^2 - r_{i-1/2}^2)$$

$$\times \frac{(\sin\Theta_{j+1/2}U_{\theta_{i,j+1/2}} - \sin\Theta_{j-1/2}U_{\theta_{i,j-1/2}})}{2(\cos\Theta_{j-1/2} - \cos\Theta_{j+1/2})}] = 0 \quad , \text{ and} \tag{17}$$

$$\delta V_{i,j} = \frac{2\pi}{3}(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\Theta_{j-1/2} - \cos\Theta_{j+1/2}) \quad . \tag{18}$$

At this point attention must turn to the type of numerical technique to be used in solving the finite-difference set of Eqs. (14) through (18). In ORBEOS we used a generalization to spherical geometry of a technique described by Zalesak in 1978.[1] Note that in using this technique no explicit artificial viscosity is introduced into the code.

Before proceeding to a discussion of the algorithm employed in ORBEOS, we must introduce the units of measurement used in the code. These units are neither cgs nor mks, but those commonly used in the laser-fusion program.

Unit of measurement	Parameter
cm	length
shk , [1(shk) ≡ 10^{-8}(s)]	time
g	mass
keV	temperature
Jk , [1(Jk) ≡ 10^{12}(erg)]	energy

Note that as yet we have not considered how to advance the finite-difference equations in time. This is done in the following section.

II. THE ALGORITHM

In May 1978, S. T. Zalesak published a paper in which he describes a new alternative form for the flux-limiting phase of the flux-corrected transport (FCT) algorithms of Boris, Book, and Hain.[2-4]

He points out several advantages of his new flux limiter over the flux limiters previously used. The new flux limiter

1) is generalizable to multidimensions without resort to Strang-type time splitting,[5]
2) eliminates "clipping" for vanishing velocity and reduces clipping at finite velocities, and
3) makes it possible, for the first time, to carry out multidimensional FCT calculations for problems not amenable to Strang-type time splitting, such as those involving incompressible or nearly incompressible flow.

The third point, above, is most crucial for any algorithm to be used in a hydrocode whose main purpose is to study fluid stability. To understand this assertion, consider the basic equation solved by the FCT algorithm,

$$\frac{\partial w}{\partial t} + \underset{\sim}{\nabla} \cdot \underset{\sim}{f} = 0 \quad , \tag{19}$$

where w is a generalized density and $\underset{\sim}{f}$ is a generalized flux. Let us consider specifically the situation where $w \equiv \rho$ (i.e., a fluid density) and $\underset{\sim}{f} \equiv \rho \underset{\sim}{v}$ (i.e., a fluid momentum density), where $\underset{\sim}{v}$ is the local fluid velocity. Equation (19) becomes

$$\frac{\partial \rho}{\partial t} + \underset{\sim}{\nabla} \cdot \rho \underset{\sim}{v} = 0 \quad , \tag{20}$$

which is just the equation for conservation of mass. If we further assume that the system is two dimensional and Cartesian, Eq. (20) written in component form becomes

$$\frac{\partial \rho}{\partial t} + v_x \frac{\partial \rho}{\partial x} + v_y \frac{\partial \rho}{\partial y} + \rho \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) = 0 \quad . \tag{21}$$

Next, assume that $\underset{\sim}{v}$ is a known function of ρ

$$\underset{\sim}{v} = \underset{\sim}{v}(\rho) \quad . \tag{22}$$

Equation (21) can be advanced in time in two ways: by Strang-type time-step splitting, or by the use of a fully two-dimensional algorithm. A typical Strang-type time-step splitting scheme is shown below.

$$v_x^{o} \equiv v_x(\rho^{o}) \quad , \tag{23}$$

$$\rho^{hx} = \rho^{o} - \frac{\delta t}{2}\left(v_x^{o}\,\frac{\delta \rho^{o}}{\delta x} + \rho^{o}\,\frac{\delta v_x^{o}}{\delta x}\right) \quad , \tag{24}$$

$$v_x^{hx} = v_x(\rho^{hx}) \quad , \tag{25}$$

$$\rho^{x} = \rho^{o} - \delta t\left(v_x^{hx}\,\frac{\delta \rho^{o}}{\delta x} + \rho^{o}\,\frac{\delta v_x^{hx}}{\delta x}\right) \quad , \tag{26}$$

$$v_y^{x} = v_y(\rho^{x}) \quad , \tag{27}$$

$$\rho^{hy} = \rho^{x} - \frac{\delta t}{2}\left(v_y^{x}\,\frac{\delta \rho^{x}}{\delta y} + \rho^{x}\,\frac{\delta v_y^{x}}{\delta y}\right) \quad , \tag{28}$$

$$v_y^{hy} = v_y(\rho^{hy}) \quad , \text{ and} \tag{29}$$

$$\rho^{n} = \rho^{x} - \delta t\left(v_y^{hy}\,\frac{\delta \rho^{x}}{\delta y} + \rho^{x}\,\frac{\delta v_y^{hy}}{\delta y}\right) \quad , \tag{30}$$

where $\delta/\delta x$ and $\delta/\delta y$ represent finite-difference approximations to the x and y derivatives, and where the superscript o stands for old, hx for half-step x, x for full step x, hy for half-step y, and n stands for new. If we resubstitute into the algorithm and write ρ^{n} in terms of old quantities, o , the expression for ρ^{n} becomes, to second order in δt,

$$\rho^n = \rho^o - \delta t \left(\left(1 - \frac{\delta t}{2} \frac{\partial v_x^o}{\partial \rho^o} \frac{\delta \rho^o}{\delta x} - \delta t \frac{\partial v_y^o}{\partial \rho^o} \frac{\delta \rho^o}{\delta y} \right) \left(v_x^o \frac{\delta \rho^o}{\delta x} + \rho^o \frac{\delta v_x^o}{\delta x} \right) \right.$$

$$+ \left(1 - \frac{\delta t}{2} \frac{\partial v_y^o}{\partial \rho^o} \frac{\delta \rho^o}{\delta y} \right) \left(v_y^o \frac{\delta \rho^o}{\delta y} + \rho^o \frac{\delta v_y^o}{\delta y} \right)$$

$$- \delta t \left\{ \frac{1}{2} \rho^o \frac{\delta}{\delta x} \left[\frac{\partial v_x^o}{\partial \rho^o} \left(v_x^o \frac{\delta \rho^o}{\delta x} + \rho^o \frac{\delta v_x^o}{\delta x} \right) \right] + v_y^o \frac{\delta}{\delta y} \left(v_x^o \frac{\delta \rho^o}{\delta x} + \rho^o \frac{\delta v_x^o}{\delta x} \right) \right.$$

$$\left. \left. + \rho^o \frac{\delta}{\delta y} \left[\frac{\partial v_y^o}{\partial \rho^o} \left(v_x^o \frac{\delta \rho^o}{\delta x} + \rho^o \frac{\delta v_x^o}{\delta x} + \frac{v_y^o}{2} \frac{\delta \rho^o}{\delta y} + \frac{\rho^o}{2} \frac{\delta v_y^o}{\delta y} \right) \right] \right\} + O(\delta t^2) \right) . \quad (31)$$

We see that if $\delta v_x^o/\delta x + \delta v_y^o/\delta y \equiv 0$ in the above algorithm, this is true only to first order in δt. The condition that the flow be incompressible is not met by this Strang-type time-step splitting algorithm.

Alternatively, a fully two-dimensional algorithm, similar to that described by Zalesak, would be

$$\underset{\sim}{v}^o = \underset{\sim}{v}(\rho^o) \quad , \quad (32)$$

$$\rho^h = \rho^o - \frac{\delta t}{2} \left[v_x^o \frac{\delta \rho^o}{\delta x} + v_y^o \frac{\delta \rho^o}{\delta y} + \rho^o \left(\frac{\delta v_x^o}{\delta x} + \frac{\delta v_y^o}{\delta y} \right) \right] \quad , \quad (33)$$

$$\underset{\sim}{v}^h = \underset{\sim}{v}(\rho^h) \quad , \text{ and} \quad (34)$$

$$\rho^n = \rho^o - \delta t \left[v_x^h \frac{\delta \rho^h}{\delta x} + v_y^h \frac{\delta \rho^h}{\delta y} + \rho^h \left(\frac{\delta v_x^h}{\delta x} + \frac{\delta v_y^h}{\delta y} \right) \right] \quad . \quad (35)$$

Clearly, if $\delta v_x/\delta x + \delta v_y/\delta y \equiv 0$, then the above algorithm preserves this condition to all orders of δt. Thus, if the flow is incompressible, this algorithm will mirror that fact exactly.

In fluid stability studies this feature of the fully two-dimensional algorithm is important, because if $(\nabla \cdot \mathbf{v})$ should be physically zero but is not numerically zero, a virtual source is introduced into the continuity equation that can lead to anamolous peaks and valleys in the solution.

For reasons mentioned above we used in ORBEOS the fully multidimensional FCT technique described by Zalesak, though generalized to r - Θ geometry.

A natural consequence of using a two-dimensional algorithm is the possibility of treating problems all the way to the origin of coordinates in spherical geometry, which is not true for a one-dimensional FCT algorithm using Strang-type time-step splitting.

For details of the technique, see Ref. 1. The basic philosophy of the FCT technique, simply stated, is that two algorithms are used to carry out the transport of the fluid quantities: An algorithm low-order, in space, that is highly diffusive but gives smooth results, is combined, through the medium of a flux-limiter, with a high-order algorithm that is very nondiffusive but causes ripples in the solution. The combination produces an algorithm that is accurate to any desired order in space and to second order in time.

III. RESULTS

ORBEOS has been used to solve numerous problems; three are discussed herein: the imploding spherical shell, the spitting cone, and the imploding spherical shell with perturbations. Note that in none of these cases was artificial viscosity introduced to handle shocks.

We monitored how well the code was conserving mass and total energy. In each case mass was conserved to an accuracy better than 1%, whereas total energy was conserved to better than 3%.

A word about the peculiarities of the three-dimensional display package used in ORBEOS (i.e., DISPLA): the radius and angle are displayed in a rectangular fashion. Three facts must be kept in mind when viewing the figures.

(1) The line at r = 0 in the figures is in reality a point and all quantities along that line are equal.

(2) The line at Θ = 0 is the axis of rotational symmetry.

(3) The values along the axis labeled density are, in all cases, on a base-10 logarithmic scale.

A. Imploding Shell

As the name implies, a shell was imploded symmetrically towards the origin of coordinates. An ideal equation of state was assumed. The maximum angle, Θ_{max}, was 45°, whereas the maximum radius, r_{max}, was 50 μm. The computational grid contained 51 equally spaced points in the radial direction and 5 equally spaced points in the angular direction (i.e., Δr = 1 μm and $\Delta\Theta$ = 11.25°). The shell was 10 μm thick and centered at the radius r = 75 μm. Shell density was 1.0 g/cm^3. The shell contained a gas at density 0.02 g/cm^3, referred to as the "fuel," and the outside of the shell was surrounded by a 10-μm-thick layer of a gas at 0.1 g/cm^3. Initially, the pressure of the outside layer of gas was to be 10 times the constant pressure throughout the shell and the fuel. The boundary condition at r_{max} was reflecting. See Fig. 3.

During the initial phase of the implosion, a shock passes through the shell and compresses the shell to a density of ~4 g/cm^3, as shown in Figs. 4 and 5. After initial compression, the shell begins to move inward compressing the low-density fuel. As the shell implodes, several shocks pass into the fuel and reflect at the origin. As the shell approaches the origin, the densities of both the shell and the fuel increase (compare Fig. 6 with Figs. 4 and 5) until the shell and fuel configuration reflects at the origin. Immediately after reflection (Fig. 7) the density is ~36 g/cm^3.

In this problem we tested the capabilities of ORBEOS in the radial direction without introducing the complications of angular motion. Reflection of the shocks at the origin of coordinates was of particular interest.

B. Spitting Cone

A cone of 45° half-angle (i.e., Θ_{max}) was chosen with a constant density of 1.0 g/cm^3 throughout. The temperature in a sheet between Θ = 25° and Θ = 45°, running from r = 0 to r = r_{max} on the outside of the cone, was chosen to be 25 times higher than the temperature inside this hot sheet. Again, an ideal gas was chosen as the computational medium. This initial configuration leads to an implosion of the gas in the hot sheet towards the axis of rotational symmetry, Θ = 0°. The imploding shock reaches the axis of rotational symmetry near the origin first, and proceeds outward in r toward r_{max}; hence, the name Spitting Cone. Figures 8, 9, and 10 show the time evolution of this system.

The number of computational points in the radial direction was 26, whereas in the angular direction 25 points were used. The increments in radius and angle were uniform and equal to $\Delta r = 1$ μm and $\Delta\Theta = 1.87^{\circ}$, respectively. In this problem we tested the capabilities of ORBEOS in both the angular and radial directions. How the code handled shocks reflected at the axis of rotational symmetry was of particular interest. The boundary condition at r_{max} was reflecting.

C. Imploding Sphere with Perturbations

A shell was assumed to implode a low-density gas. However, the shell contained a Taylor-unstable sinusoidal perturbation in radius on its outside surface. This problem models the Taylor instability in a converging spherical system. An aluminum equation of state was chosen for the computational medium. The maximum angle, Θ_{max}, was 90° so that the reflecting boundary condition on the surface, Θ_{max}, was physical; this represents a simulation of a full sphere. The maximum radius was 50 μm. The computational grid contained 51 equally spaced points in radius and 36 equally spaced points in angle (i.e., $\Delta r = 1$ μm and $\Delta\Theta = 2.57^{\circ}$).

The initial fluid density on the mesh is given algebraically as

$$\rho = 0.02 \text{ g/cm}^3 , \qquad r < 36 \text{ μm}$$

$$\rho = 1.0 \text{ g/cm}^3 , \qquad 36 \leq r \leq 39 \text{ μm}$$

$$\rho = (0.1, \rho_\theta) \text{ g/cm}^3, \qquad 39 < r \leq 50 \text{ μm} ,$$

where

$$\rho_\theta = \exp\{[r[1 - 0.022 \cos(6\pi\Theta/\Theta_{max})] - 39]^2/D^2\} , \tag{36}$$

and

$$D = 5.0/\sqrt{-\ell n\,(0.1)} \text{ μm} . \tag{37}$$

The initial pressure was

$$P = 0 \text{ Jk/cm}^3 \quad r \leq 39 \text{ μm} \ ,$$

$$P = 1.0 \times 10^{-4} \text{ Jk/cm}^3 \quad 39 < r \leq 50 \text{ μm} \ ,$$

and the boundary condition at r_{max} was reflecting.

The wavelength of the initial perturbation was ~21 μm. Figure 11 shows the initial density profile. As the system evolved, the perturbations grew in amplitude as seen in Figs. 12, 13, and 14.

A diagnostic was introduced into the code specifically for this problem.

$$\alpha = \frac{\int \rho U_r^2 dV}{\int \rho (U_r^2 + U_\theta^2) dV} \ , \tag{38}$$

which is the ratio of total radial kinetic energy to total kinetic energy. In this problem even though the fingers appear to have grown to large amplitude, the smallest value of α was ~0.97, which means that only 3% of the total kinetic energy was in the angular direction. Therefore, the structures form predominantly due to differences in the radial momentum at different angles.

REFERENCES

1. S. T. Zalesak,"Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids," J. Comp. Phys. 31, 335 (1979).

2. J. P. Boris and D. L. Book,"Flux-Corrected Transport. I. SHASTA,a Fluid Transport Algorithm that Works," J. Comp. Phys. 11, 38 (1973).

3. D. L. Book, J. P. Boris, and K. Hain,"Flux-Corrected Transport II. Generalizations of the Method," J. Comp. Phys. 18, 248 (1975).

4. J. P. Boris and D. L. Book,"Flux-Corrected Transport. III. Minimal Error FCT Algorithms," J. Comp. Phys. 20, 397 (1976).

5. D. Gottlieb,"Strang-Type Difference Schemes for Multidimensional Problems," SIAM J. Num. Anal. 9, 650 (1972).

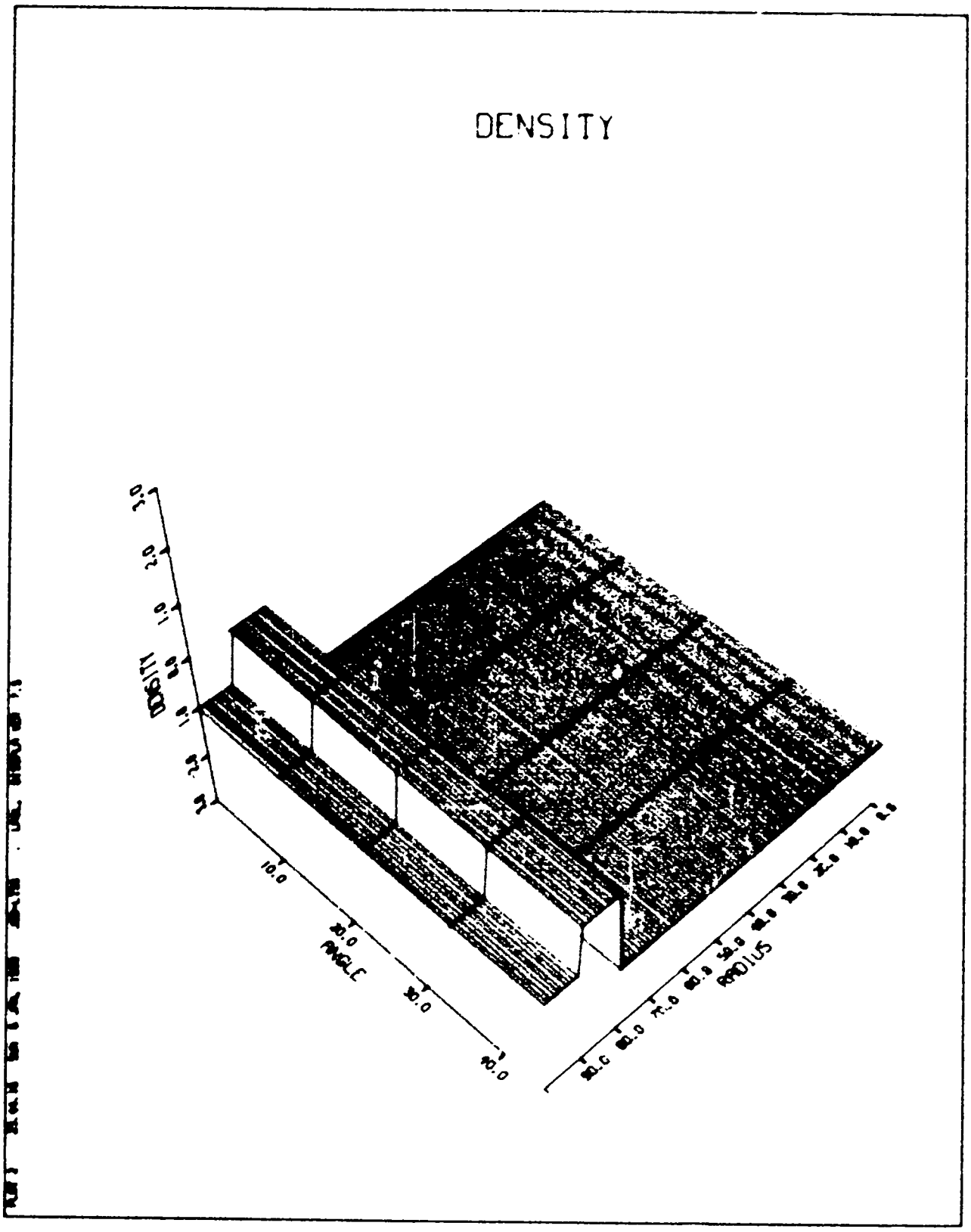

Fig. 3. Initial density configuration of imploding spherical shell. See text for details.

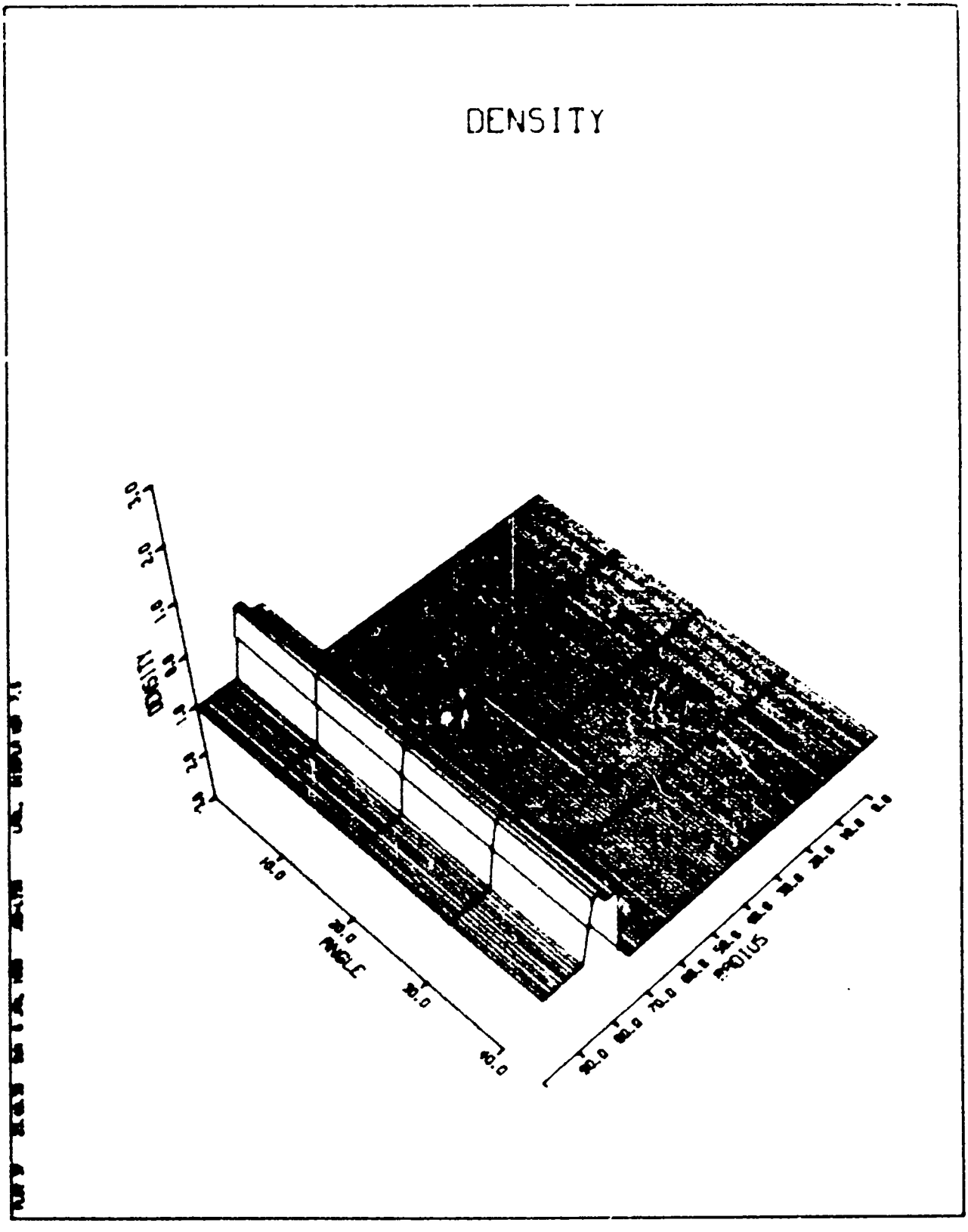

Fig. 4. Shock beginning its passage through spherical shell.

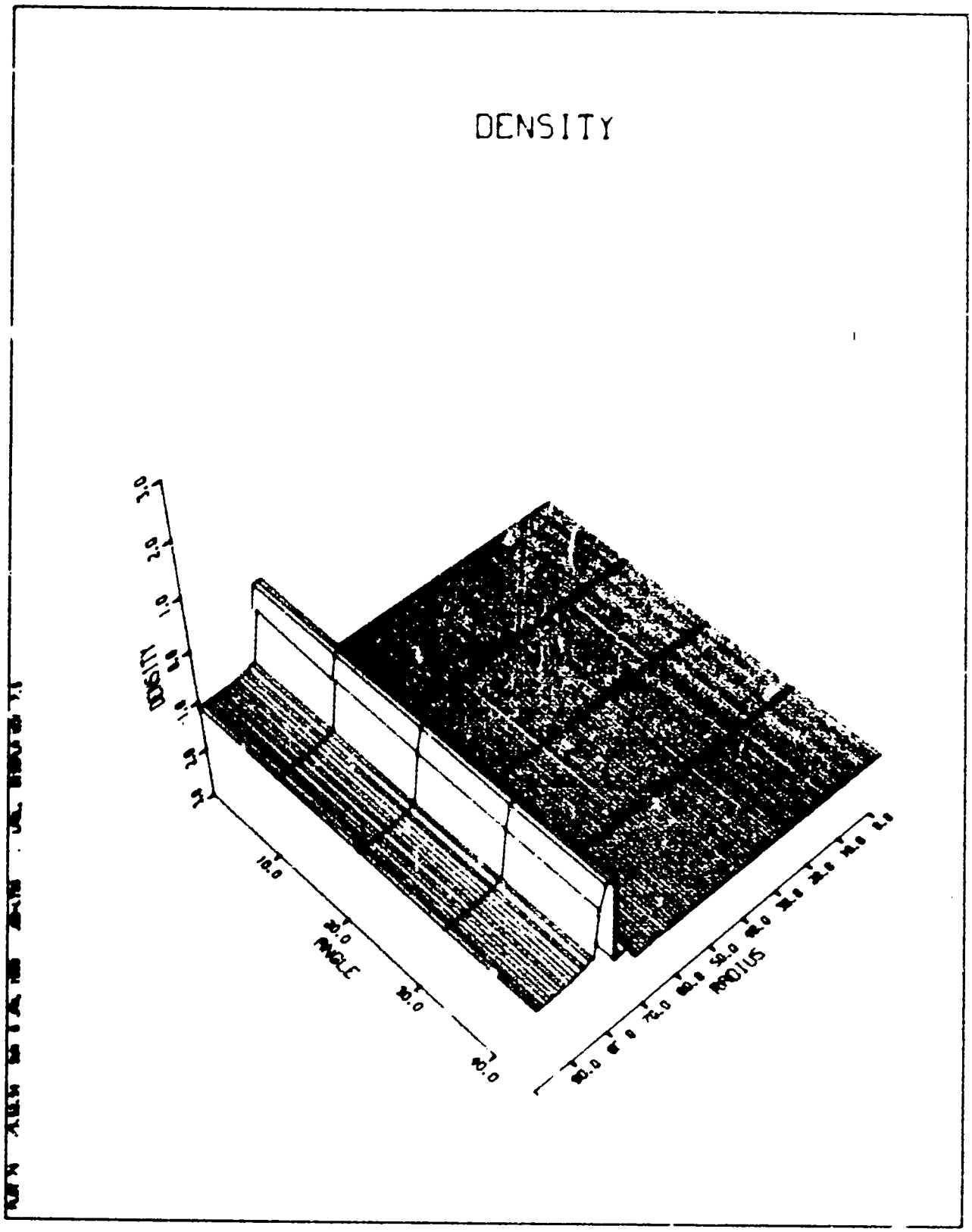

Fig. 5. Compression of original shell after passage of shock.

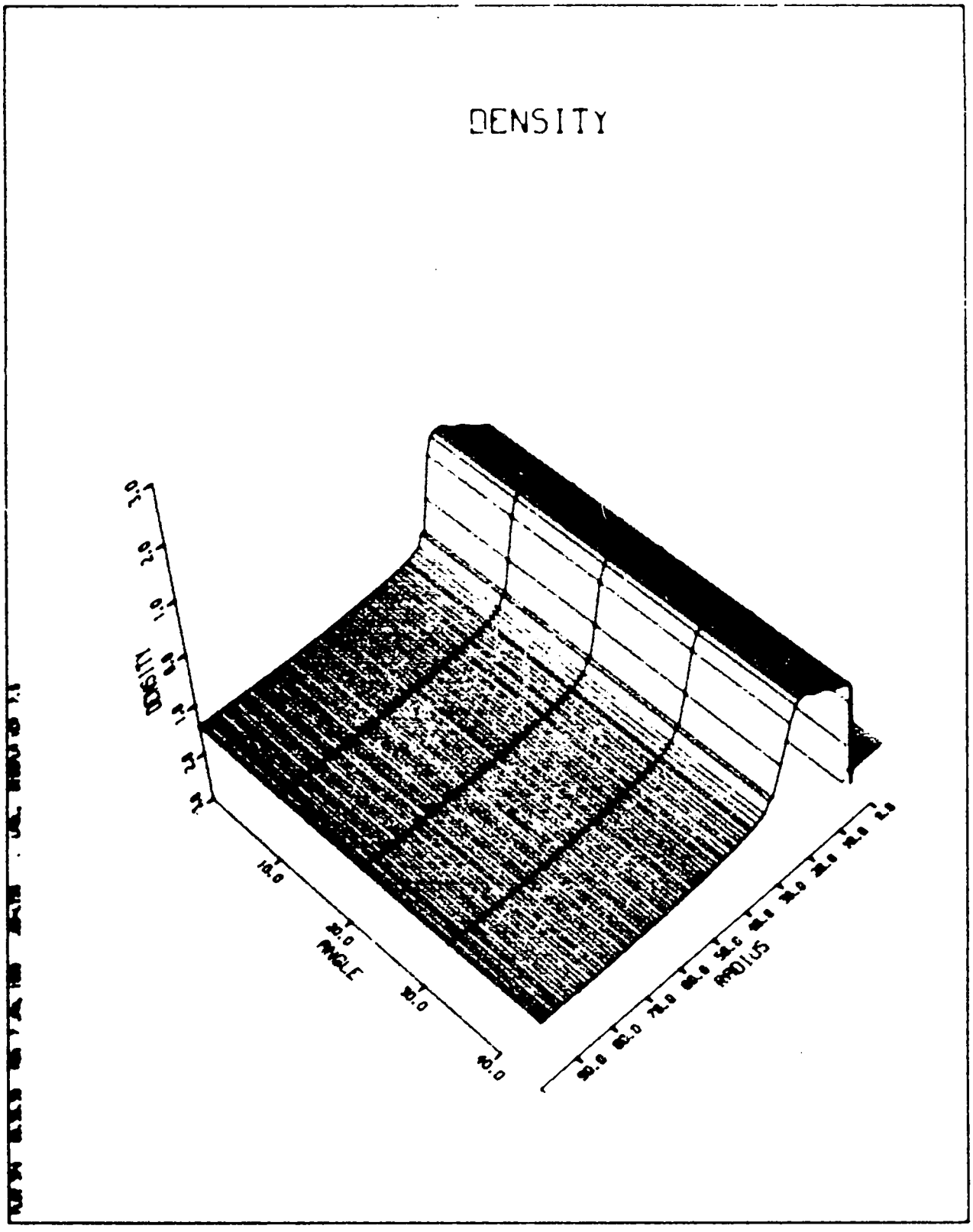

Fig. 6. Imploding shell just prior to hitting the origin of coordinates.

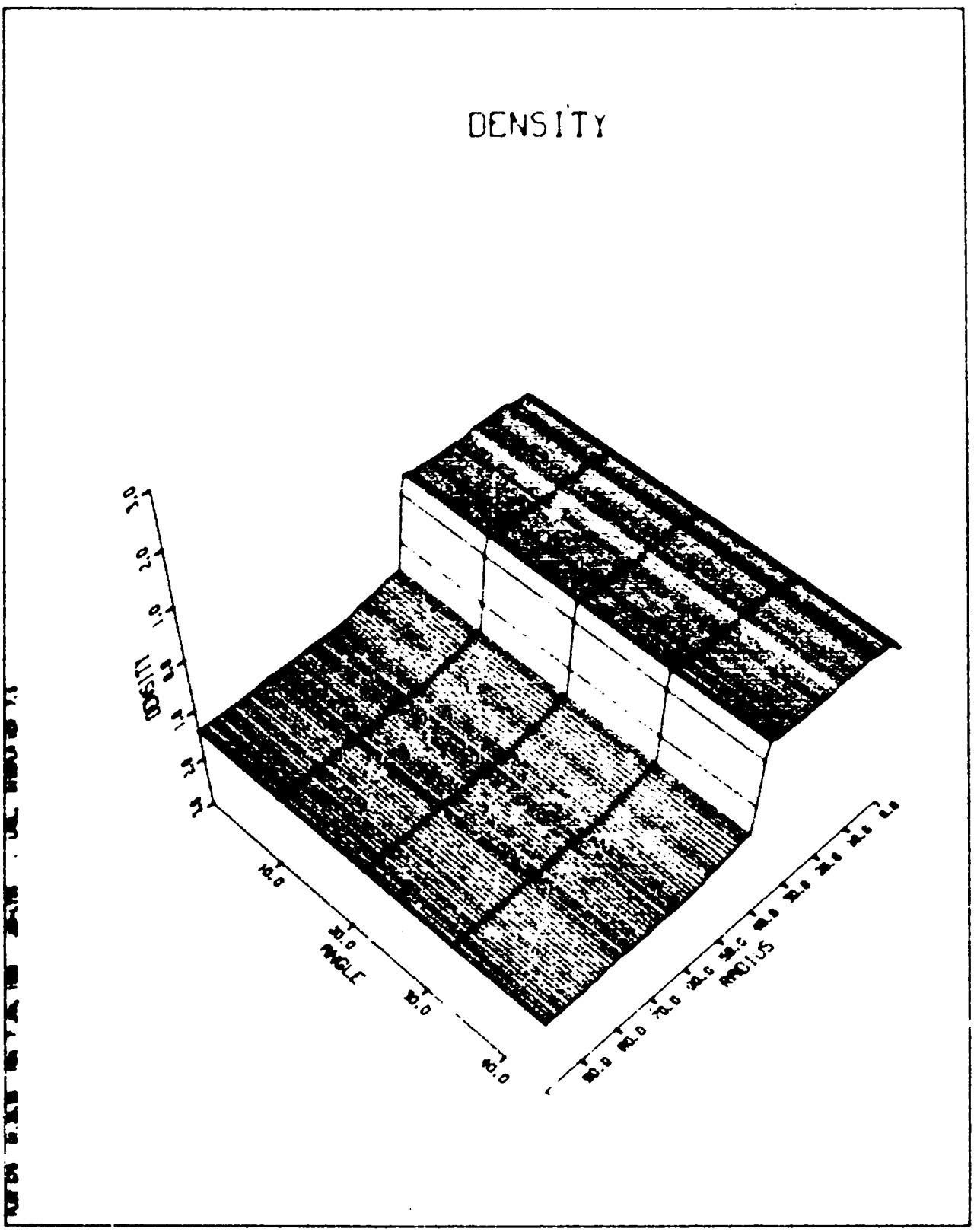

Fig. 7. Shell after rebound from the origin.

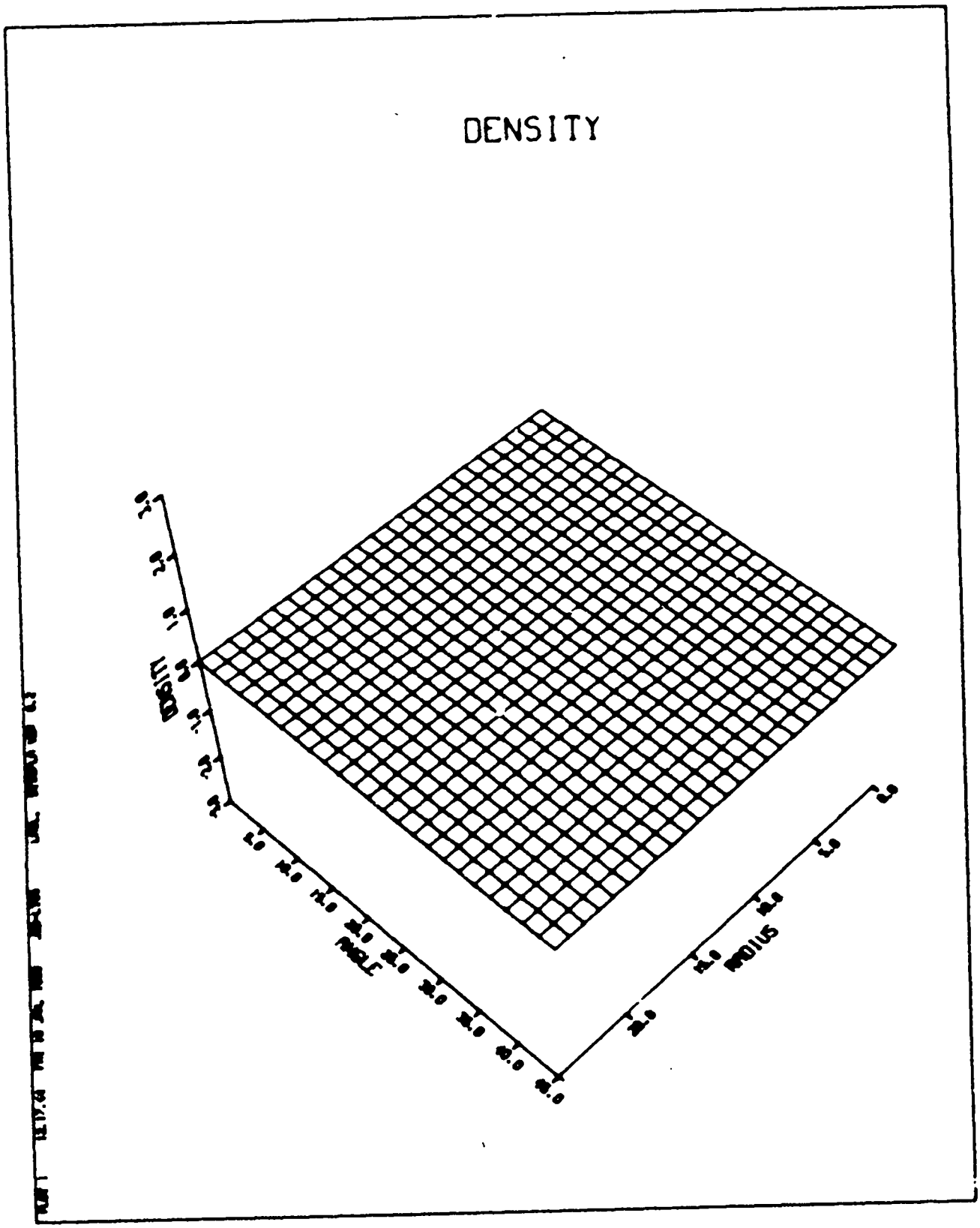

Fig. 8. Initial density configuration of the spitting-cone problem. See text for details.

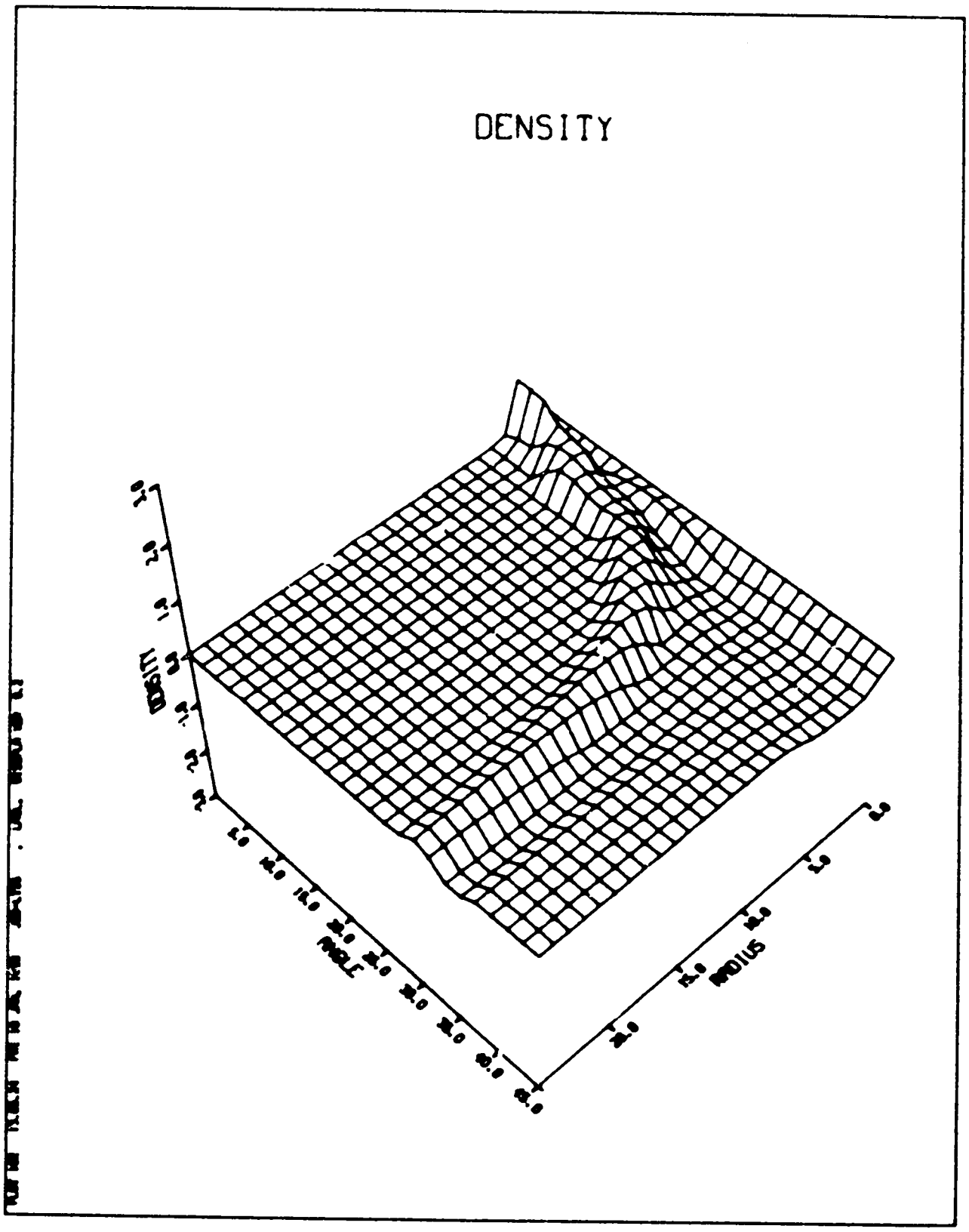

Fig. 9. Shock wave just reaching the axis of rotational symmetry near the origin of coordinates. Note the structure of the axially directed shock.

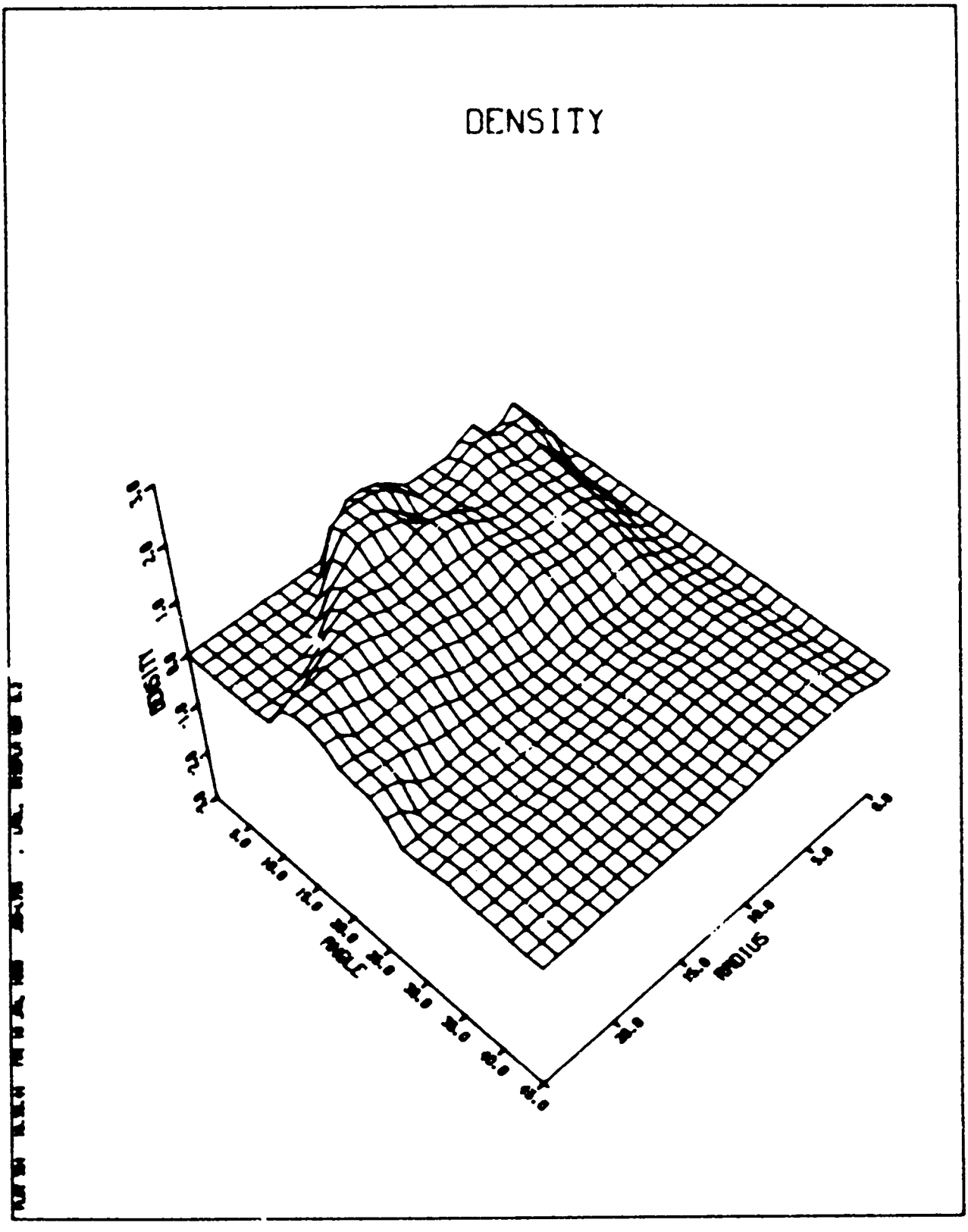

Fig. 10. Axial shock propagating along the axis of rotational symmetry.

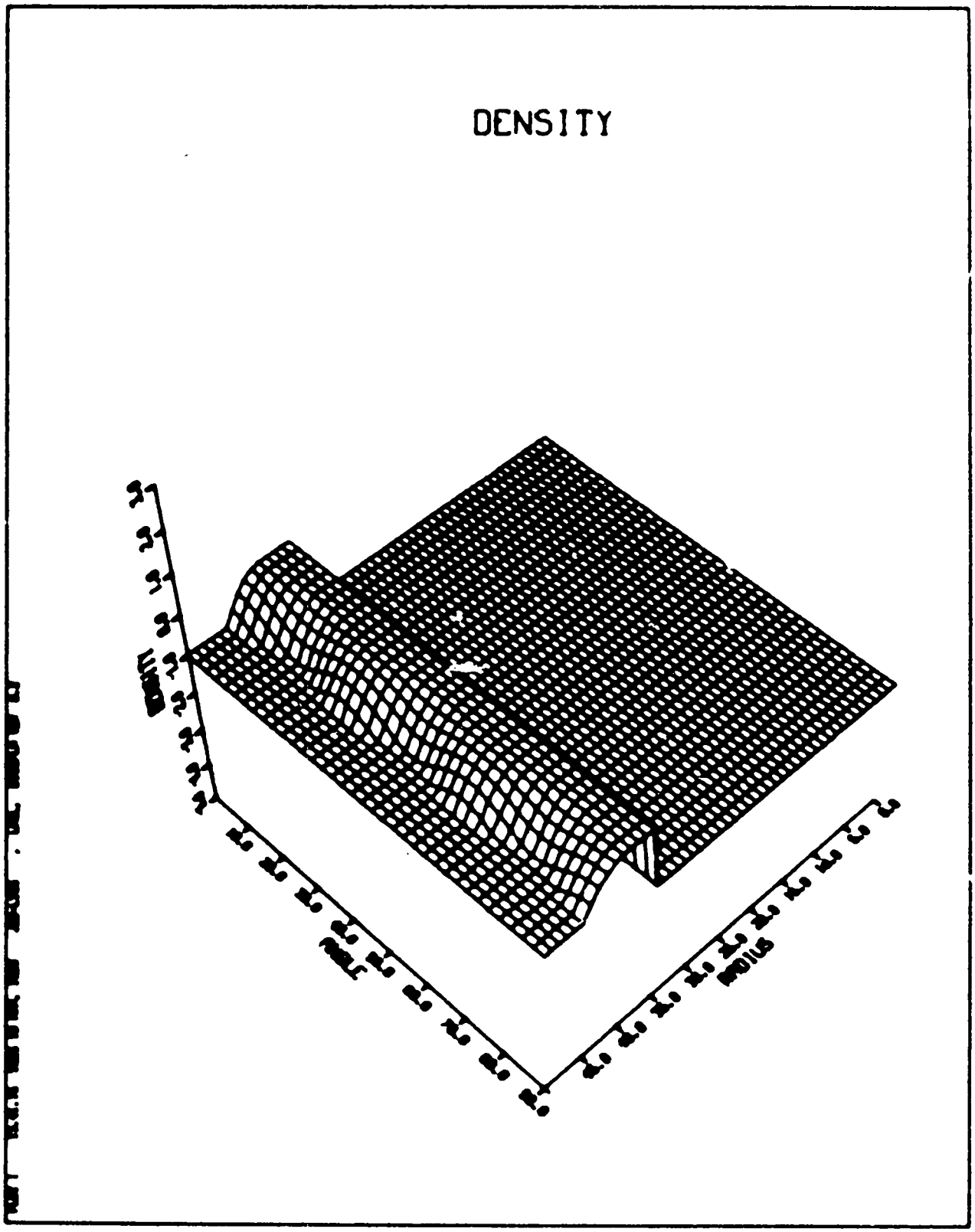

Fig. 11. Initial density configuration for imploding spherical shell with perturbations. See text for details.

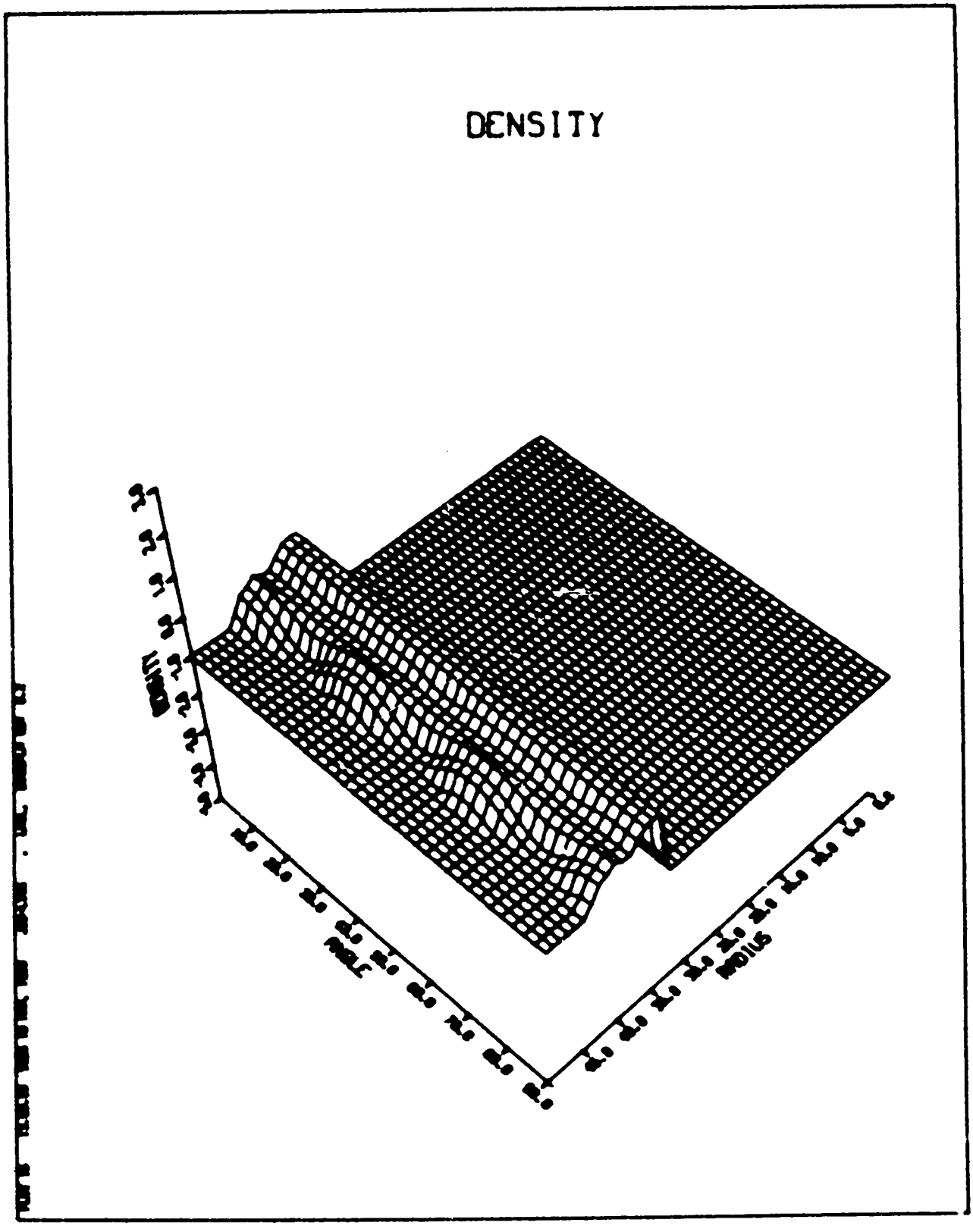

Fig. 12. Shell after passage of shock.

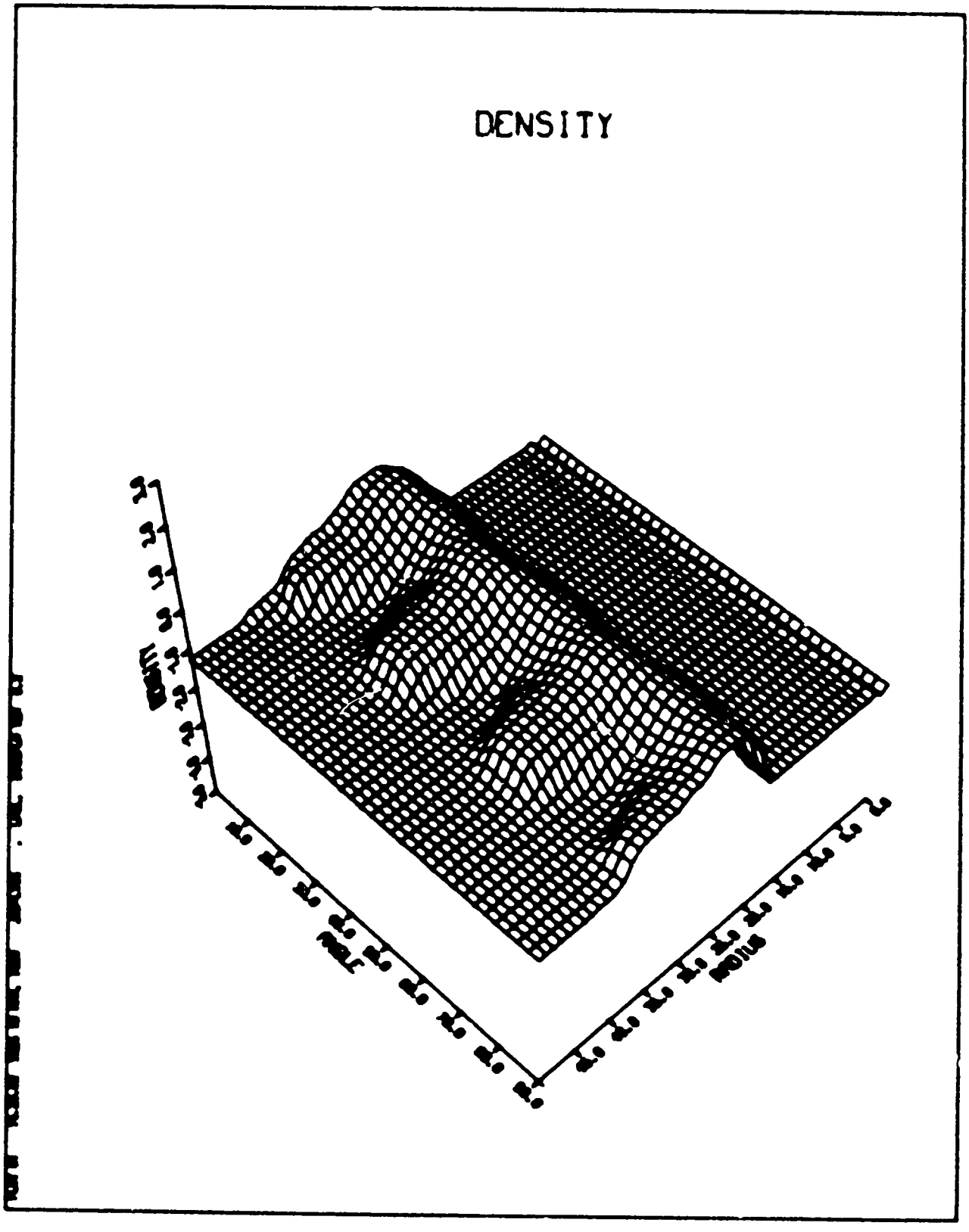

Fig. 13. Development of Taylor instability on outside of the shell.

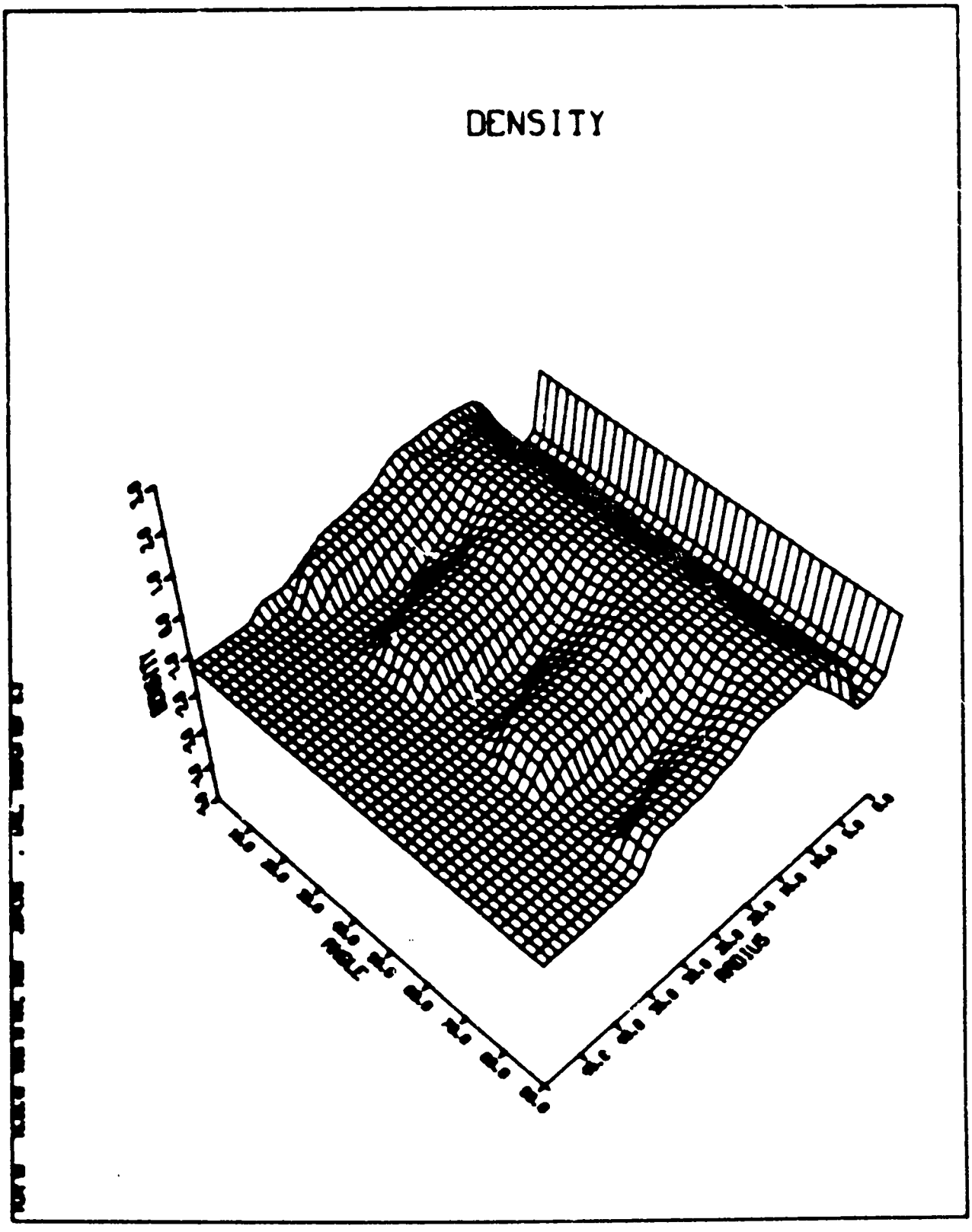

Fig. 14. Compare size of Taylor spikes at this later time with those in Fig. 13.

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche $3.50 (A01)

Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code
001-025	$ 5.00	A02	151-175	$11.00	A08	301-325	$17.00	A14	451-475	$23.00	A20
026-050	6.00	A03	176-200	12.00	A09	326-350	18.00	A15	476-500	24.00	A21
051-075	7.00	A04	201-225	13.00	A10	351-375	19.00	A16	501-525	25.00	A22
076-100	8.00	A05	226-250	14.00	A11	376-400	20.00	A17	526-550	26.00	A23
101-125	9.00	A06	251-275	15.00	A12	401-425	21.00	A18	551-575	27.00	A24
126-150	10.00	A07	276-300	16.00	A13	426-450	22.00	A19	576-600	28.00	A25
									601-up	†	A99

†Add $1.00 for each additional 25-page increment or portion thereof from 601 pages up.