LA-SUB--95-106

A Dynamic Explosive Model for MACH2

With Applications to Magnetic Flux Compression Generators

John J. Watrous and Michael H. Frese

March 2, 1993

Report 93-04

Los Alamos National Laboratory Contract 9-XT1-Q4340-1

NumerEx

1400 Central SE, Suite 2000
Albuquerque, New Mexico 87106-4811

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

GH

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

SecretMilitaryTechnology.com

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SecretMilitaryTechnology.com

Contents

List of Figures

1. Introduction

An explosive model has recently been designed and added to MACH2 to enable that code to be used as a tool for studying explosive magnetic flux compression generators. This report describes this model and gives examples of its use in both one- and two-dimensional simulations. Section 2 will provide a description of the model. One-dimensional simulations will be discussed in Section 3. Section 4 will show examples of two-dimensional simulations. Appendices contain input decks for the one- and two-dimensional simulations and a listing of the modifications made to MACH2 for this purpose.

2. The Dynamic Explosive Model

Models useful for computational modeling of detonations have been under continuous use and development for decades. They have achieved a high degree of sophistication, and are capable of predicting experimental results with quite adequate accuracy. However, such models have not yet found wide use in multidimensional MHD codes. As the behaviors of primary interest in an explosive magnetic flux compression generator are those of the detonating driver and of the compressing magnetic field, an MHD code with an explosive model is potentially of great value for designing such devices.

The intent of this effort was not to produce a new explosive model, nor to endow MACH2 with the most sophisticated state-of-the-art model available. Rather, the intent was to outfit MACH2 in the most expedient means possible with a simple model that would allow it to simulate the basic features of a detonating explosive. Thus, the primary concerns guiding the design reported here were (1) that the model be capable of reproducing the essential qualitative behavior of an explosive and provide some degree of control over the quantitative behavior, and (2) that the model be compatible with the existing structure of MACH2. The qualitative features which were considered essential were that the explosive exhibit a threshold behavior *i.e.* that it not detonate unless sufficiently perturbed, that the process of detonation release a set amount of energy which would then be available to drive on the detonation, and that once material had detonated, it would not be able to detonate again. Quantitative features deemed important to match were the pressure in the newly detonated material and the propagation speed of the detonation wave. With respect to the structure of MACH2, the goal was to produce a modular package that could fit into the existing multiblock, arbitrary Langragian-Eulerian MHD algorithms without requiring that they be modified extensively.

The model used here is an adaptation of a technique proffered by Dr. G. McCall of Los Alamos National Laboratory. The essence of the technique is to define a set of conditions under which undetonated material will detonate, then to

monitor the conditions in each undetonated cell until the chosen conditions are met. Once that occurs, the value of the internal energy of that cell is incremented by a certain amount, and the material is marked as detonated. The detonation conditions suggested by McCall are (1) the artificial viscosity in the given cell is decreasing and (2) the internal energy in the cell exceed a critical value. These conditions will now be described in greater detail.

Artificial viscosity, denoted here as q, is a numerical technique used predominantly in Lagrangian calculations to enhance the numerical stability of the hydrodynamics algorithm in the presence of a shock wave. Without artificial viscosity, hydrodynamic quantities tend to oscillate wildly behind a shock transition. The stabilizing nature of q is effected by modifying the momentum equation to read

$$\frac{d}{dt}\vec{v} = -\frac{1}{\rho}\nabla(p+q), \tag{1}$$

where the hydrodynamic pressure has been augmented by the inclusion of the artificial viscosity, q. Recipes for calculating q abound, but most share the feature that q is zero unless the cell is under compression, in which case q is set to a positive value tied to the local gradient of the fluid velocity. MACH2 uses the following recipe:

$$q = \begin{cases} 0 & \text{if} \quad \nabla\cdot\vec{v} \le 0 \\ \mu\rho\, dl^2(\nabla\cdot\vec{v})^2 & \text{if} \quad \nabla\cdot\vec{v} > 0 \end{cases} \tag{2}$$

where μ is a dimensionless input constant, typically near unity in value, ρ is the mass density, and dl is the cell size. Strong shocks are characterized by a value of q equal in magnitude to p. Numerically, the density ahead of a shock increases for several cells, telegraphing the arrival of the shock. The size of this precursor zone can be controlled by the parameter μ, but the point is that in this region, since it is undergoing some degree of compression, q has a nonzero value. In fact, q increases over several orders of magnitude in this transition region, until it is comparable to the peak pressure. Behind this point, that is, on the upstream side of the shock, q decreases over a few cells until the shock compression is over. Thus, the point at which q begins to decrease is a useful indicator for the peak of the shock, and motivates its use as part of the detonation criteria.

A decreasing value of q is not alone sufficient to provide a useful detonation criterion. Density gradients can diffuse numerically at the grid speed, $c_g = \frac{\Delta x}{\Delta t}$ leading to very small compressions and hydrodynamically insignificant nonzero values of q. Thus, keying on decreasing values of q alone would lead to a detonation wave whose speed would be determined by the local grid speed, and would occur in response to the most minute of perturbations. However, imposing the simultaneous requirement that the cell's internal energy exceed a critical value has the effect of filtering out these uninteresting and unimportant compressions. The precise value of the critical energy density depends on the explosive being modeled. Likewise, the value of the

internal energy to which to boost a detonating cell depends on the material being modeled. We found that a critical energy density of 5×10^5 J/m^3 and a detonation energy density of 5×10^6 J/m^3 allowed detonation speeds and pressures comparable to those reported for many high explosives to be obtained in the simulations.

Once the material in a cell has detonated, it is crucial to tag it in some way so that it cannot detonate again. MACH2 has a convenient method for doing this, but its use requires that the following restrictions be followed. A block containing the explosive must consist entirely of the explosive. Only one type of explosive is allowed. Each explosive block must be entirely Lagrangian, and the boundaries of such blocks must also be entirely Lagrangian. Finally, as the designation of explosive material as detonated or not detonated makes use of MACH2's multimaterial capability, this capability is effectively disabled in explosives calculations. Once the material in a cell has detonated, its value for the variable con2 is changed from zero to one. This instructs the equation of state routine to use a different equation of state for the detonated material than for the undetonated material. As the calculation is required to be entirely Lagrangian, the contents of any given cell do not mix with those of its neighbors. This insures that once a material has detonated, it will not flow into a region of undetonated material and contribute more than once to the ongoing explosion.

The equation of state used here for the detonated material is the Gruneisen equation of state. This is an analytic formulation wherein the pressure is determined by the expression

$$p = \Gamma\rho\varepsilon + \rho_o c_o^2 \frac{\left[1 - \frac{1}{2}(\Gamma-1)\left(1-\frac{\rho_o}{\rho}\right)\right]}{\left[\left(1-\frac{1}{2}(\Gamma+1)\right) + \frac{1}{2}(\Gamma+1)\frac{\rho_o}{\rho}\right]}\left(1-\frac{\rho_o}{\rho}\right) \tag{3}$$

where ε is the specific internal energy, ρ is the mass density, Γ is the Gruneisen coefficient, ρ_o is the reference density, and c_o is the reference sound speed. Numerical tests show that the parameters Γ and c_o have some degree of influence over the speed of the propagating detonation wave. In tamped 1-d tests, with a reference density of 1.894×10^3 kg/m^3, an initial internal energy density of 0.267×10^6 J/m^3, and a detonation energy release of 5.0×10^6 J/m^3, the speed of the detonation wave for c_o = 0.141 cm/μs was observed to increase from 0.527 cm/μs to 0.638 cm/μs as Γ increased from 2.0 to 2.37. For Γ fixed at 2.0, the detonation wave speed increased from 0.492 cm/μs to 0.550 cm/μs as c_o varied from 0.01 cm/μs to 0.282 cm/μs. Thus, these two parameters allow the detonation speed to be tuned, giving the user some degree of control over the detonation characteristics.

The effects of different detonation initiation schemes have not yet been studied here in any great detail. The initiation method used to date has been a simple, but somewhat artificial, method wherein a chosen group of computational cells is

assumed to be undergoing detonation just as the computation starts. This is done by giving to these cells just enough internal energy to insure that their internal energy density exceeds the critical value. Their artificial viscosity is also adjusted by setting the old values of q to 2 and the new values to 1. This insures that in these chosen cells, q appears to be decreasing, and thus allows the detonation criterion to be met.

3. One-Dimensional Demonstration

One-dimensional tests afford a quick and convenient means of surveying the characteristics of the detonation model. The 1d demonstration reported here used 20 cells to span a distance of 20 cm. The initial conditions were a uniform distribution of mass with density 1.894×10^3 kg/m^3, temperature of 0.025 eV (room temperature), and zero flow velocity. The Gruneisen parameters Γ and c_o had values 2.37 and 0.141 cm/μs, respectively. The detonation was initiated at the right and propagates to the left. Boundary conditions at both the right and left represented impenetrable, immobile walls so that this was a tamped detonation.

Figures 1, 2, 3, 4, and 5 show the mass density, the temperature, the pressure, the detonated material tag, and the flow speed, respectively at t = 10 μs. The detonated material tag has a value of 0 for undetonated material and a value of 1 for detonated material. Plots of these same quantities at t = 22.5 μs are shown in Figures 6, 7, 8, 9, and 10.

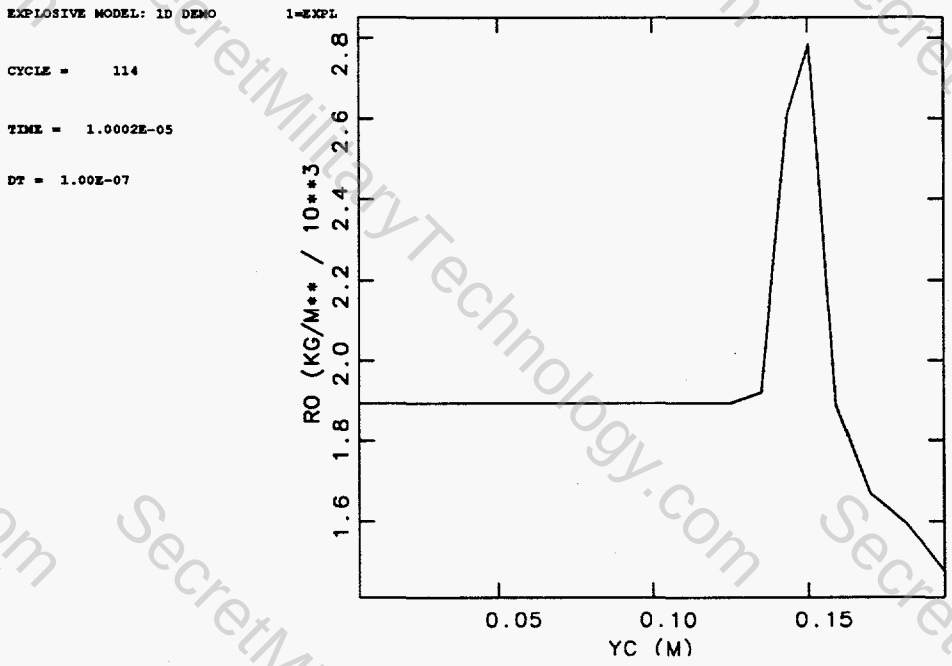

Figure 1 Mass density at 10 μs.

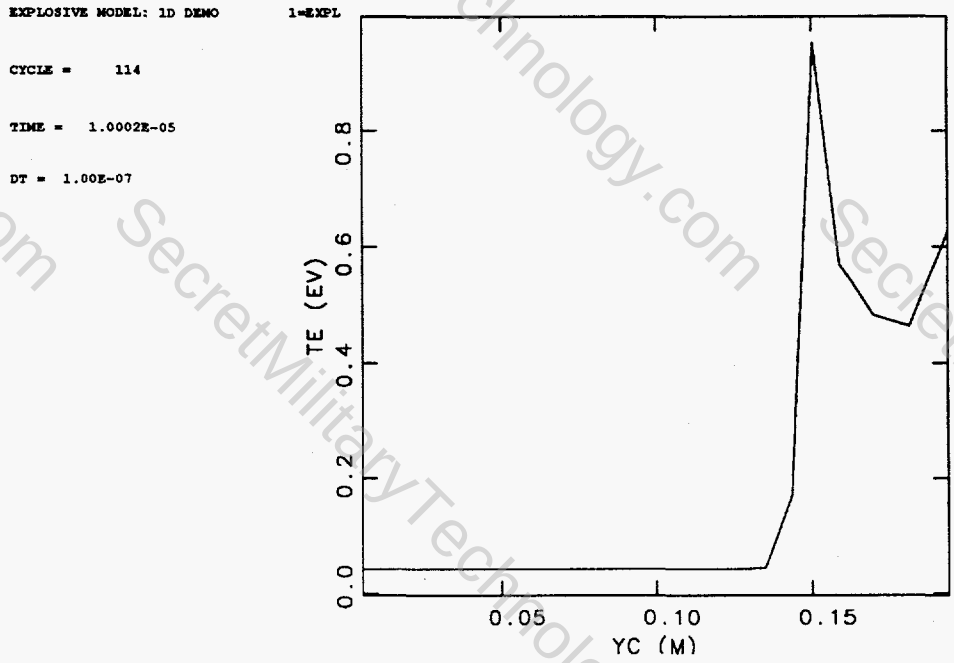

Figure 2 Temperature at 10 μs.

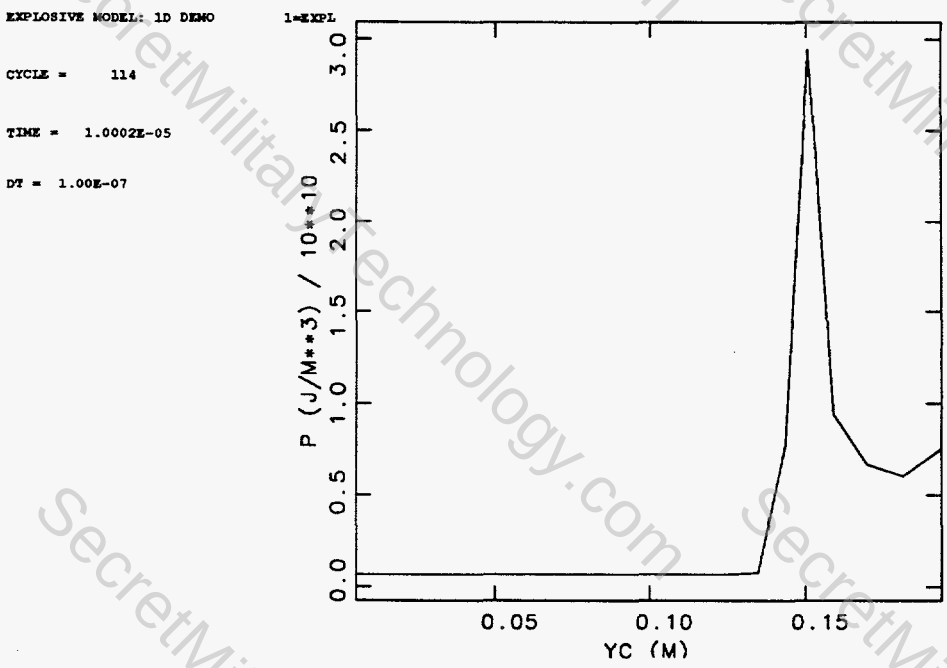

Figure 3 Pressure at 10 μs.

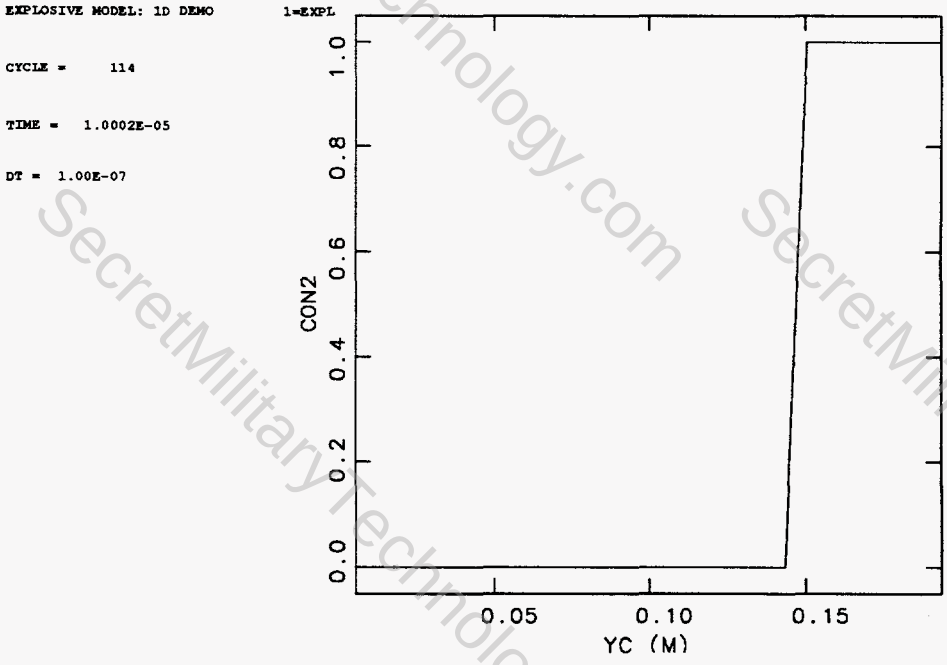

Figure 4 Detonated material tag at 10 μs.

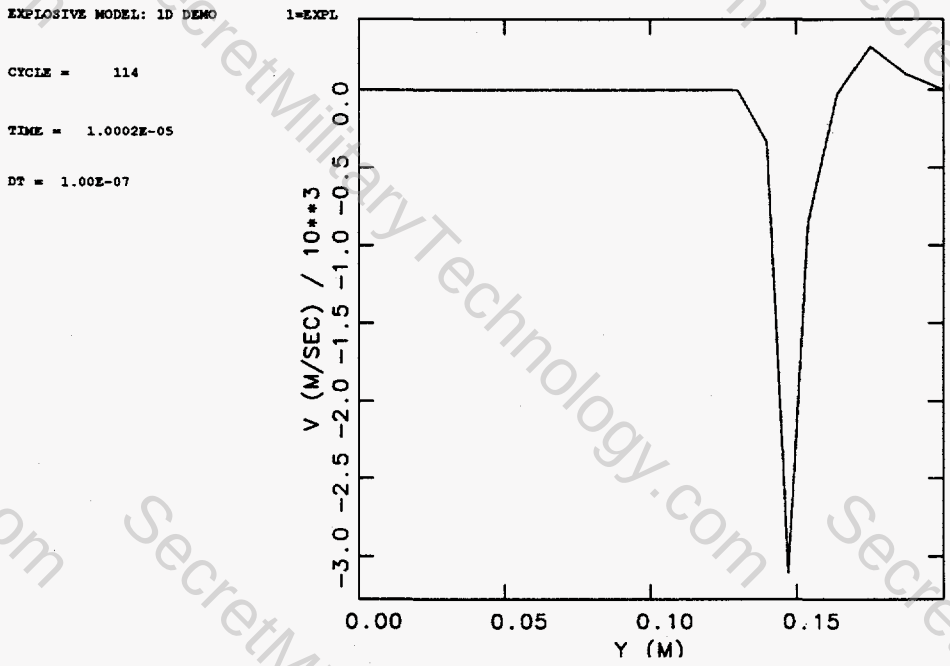

Figure 5 Flow speed at 10 μs.

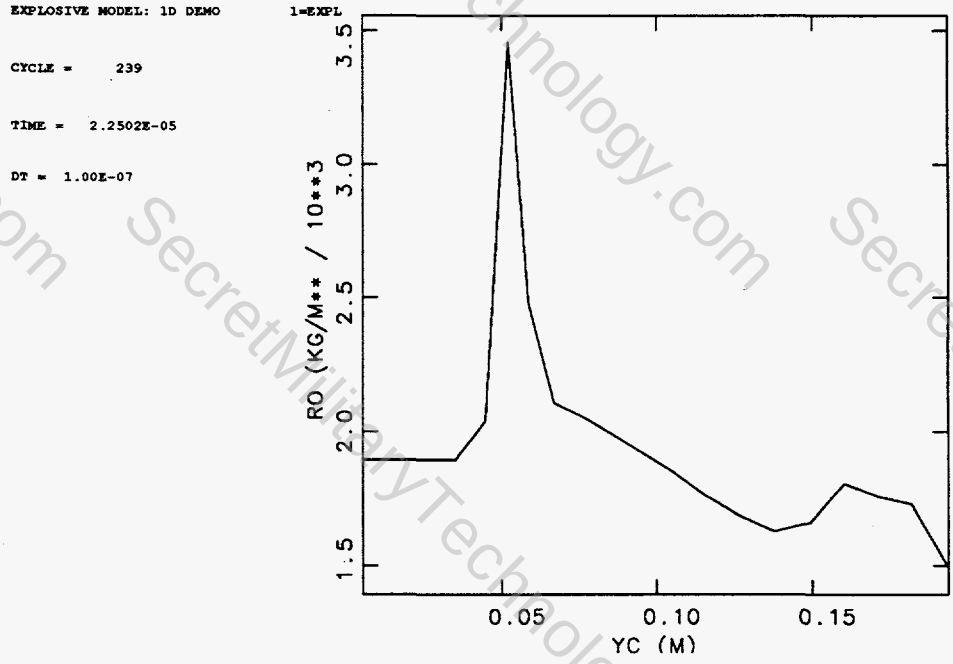

Figure 6 Mass density at 22.5 μs.

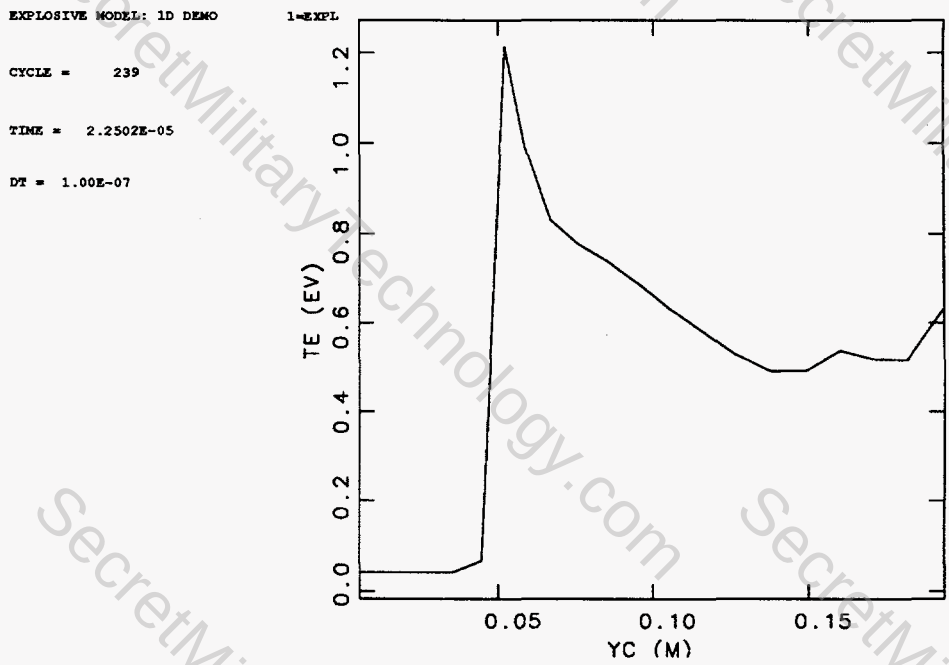

Figure 7 Temperature at 22.5 μs.

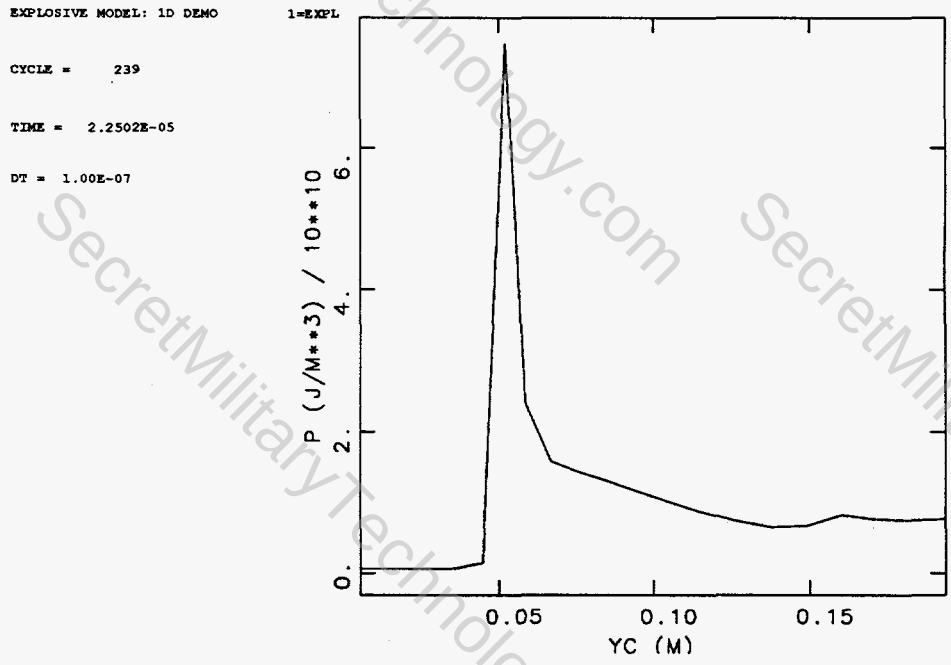

Figure 8 Pressure at 22.5 μs.

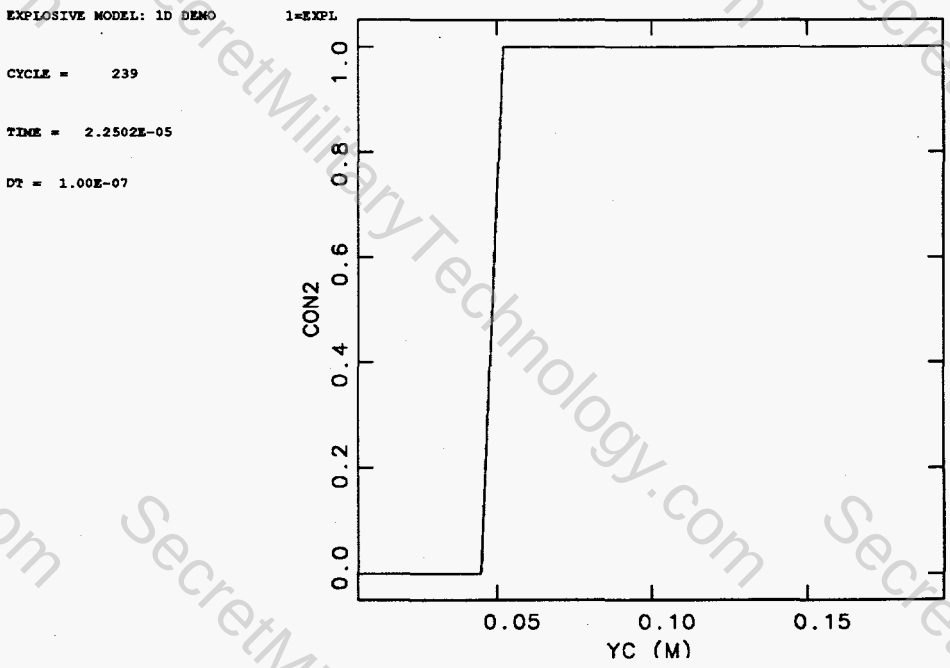

Figure 9 Detonated material tag at 22.5 μs.

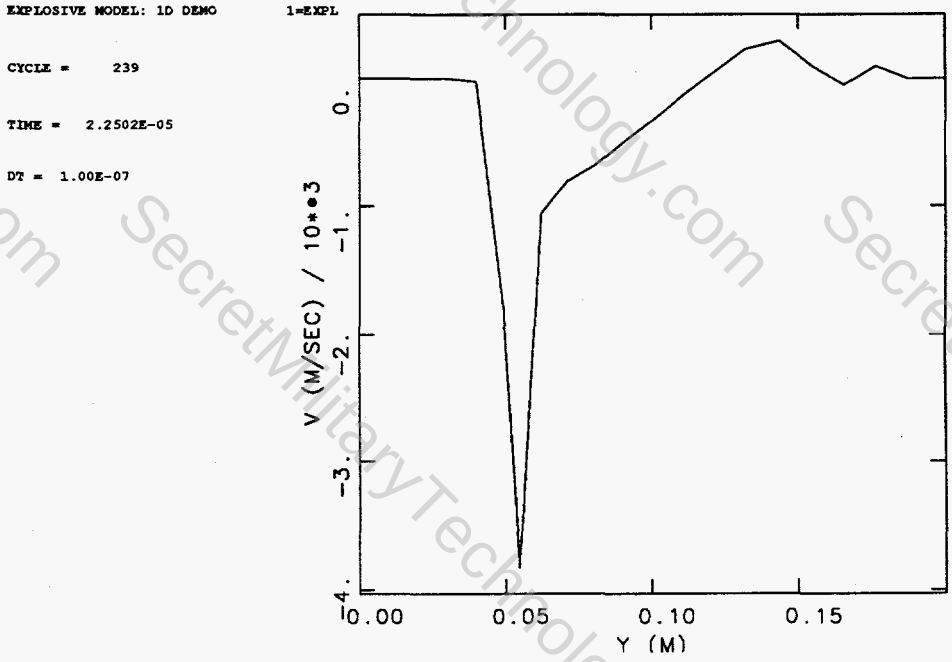

Figure 10 Flow speed at 22.5 μs.

4. Two-Dimensional Demonstration

Figure 11 shows the grid used for the two-dimensional demonstration. The geometry is cylindrical with radius increasing from left to right and axial distance increasing from bottom to top. The region from the left hand boundary at r = 10 cm to r = 15 cm is a void region. The thin layer of cells from r = 15 cm to r = 16 cm represents an aluminum armature. The SESAME tables are used for the aluminum equation of state. The region from r = 16 cm to r = 20 cm contains the explosive material. The parameters describing the explosive are identical to those used in the one-dimensional demonstration reported above. The detonation is initiated in the row of explosive cells at the top of the grid plot. An azimuthal magnetic field is present in the void region. The field has a maximum value of 1 T at the inner radius of the void region and decreases as 1/r. A return current flows on the inner surface of the armature; there is initially no magnetic field in either the armature nor the explosive.

Figures 12, 13, 14, and 15 show grid, fluid velocity, magnetic field, and internal energy density at t = 10 μs. At this point, the detonation is well under way; the portion of the armature nearest the initiation region has been accelerated to a speed of approximately 0.167 cm/μs. Figures 16, 17, 18, and 19 show the same quantities plotted at t = 20 μs, figures 20, 21, 22, and 23 at t = 30 μs. Averaged over the 30 μs it has taken the detonation wave to cross the simulation domain, the detonation speed is 0.67 cm/μs. On this time scale, the armature acts as a conductor, sweeping up the magnetic flux contained between it and the inner conductor. Some magnetic field does penetrate into the armature, but essentially none diffuses completely through it. The void region has been reduced in area by roughly a factor of two at t = 30 μs. This corresponds to the approximate doubling observed in the magnetic field intensity.

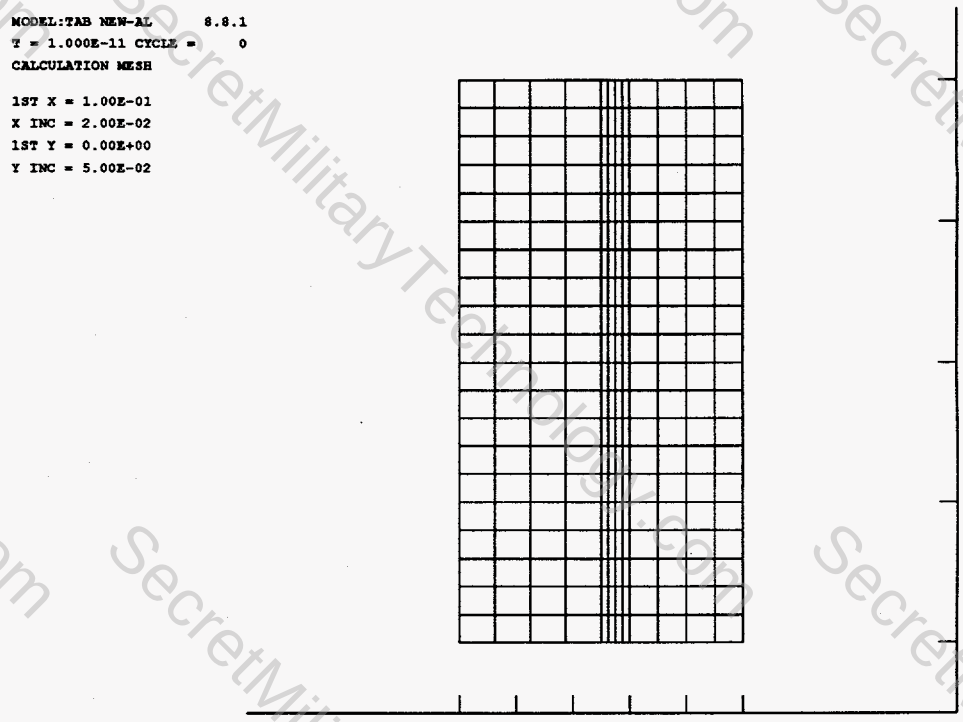

Figure 11 Initial simulation grid for two-dimensional demonstration.

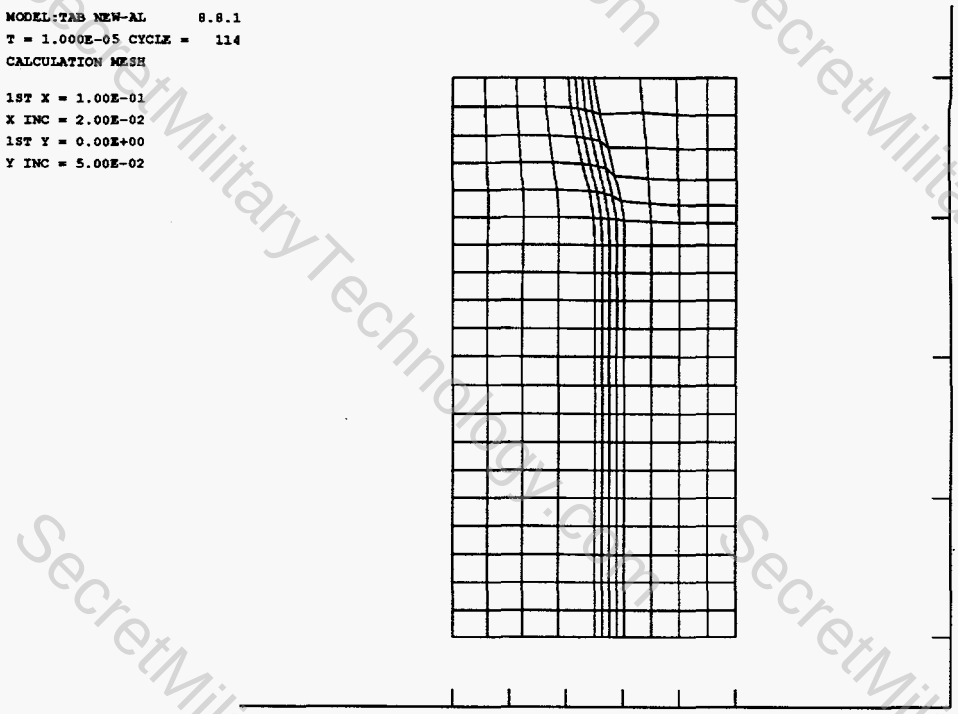

Figure 12 Simulation grid at 10 μs.

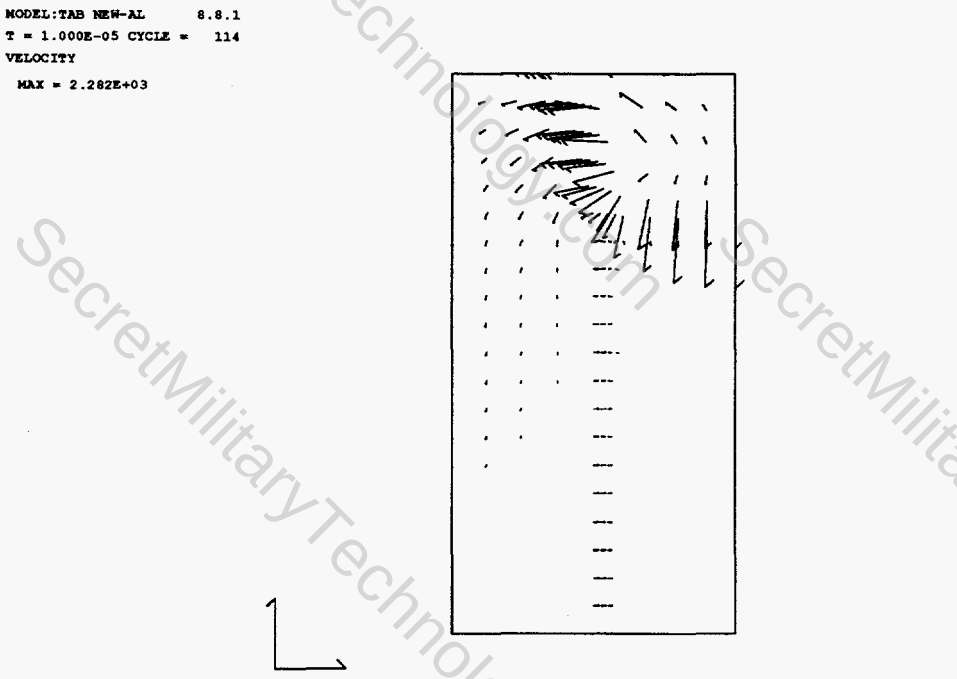

Figure 13 Fluid velocity at 10 μs.

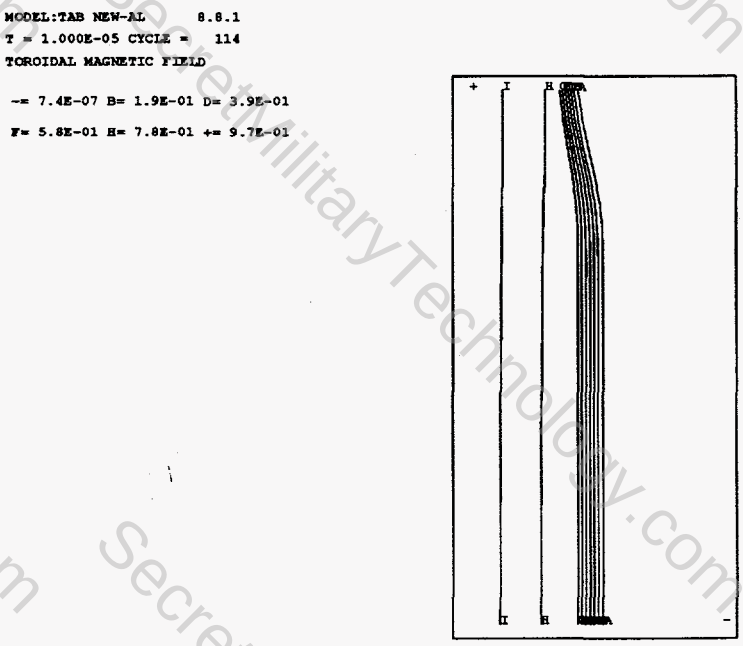

Figure 14 Azimuthal magnetic field at 10 μs.

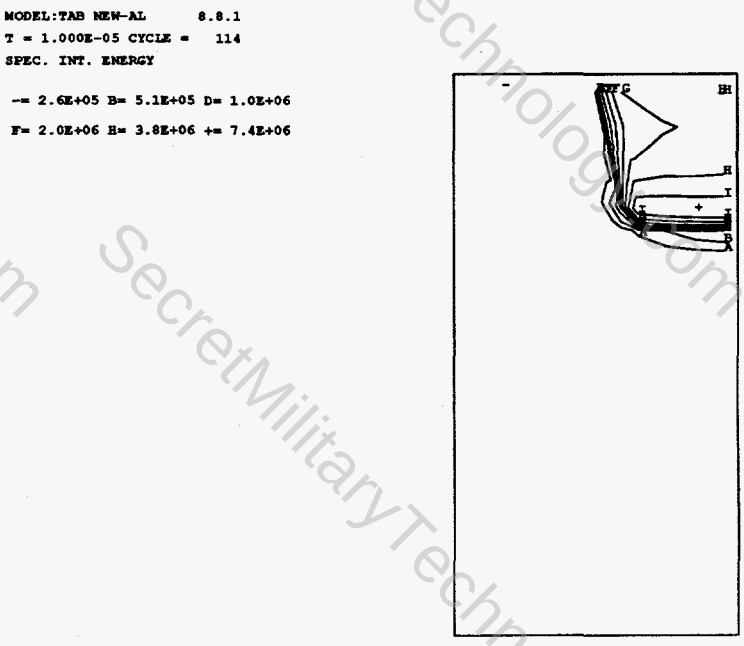

Figure 15 Internal energy density at 10 μs.

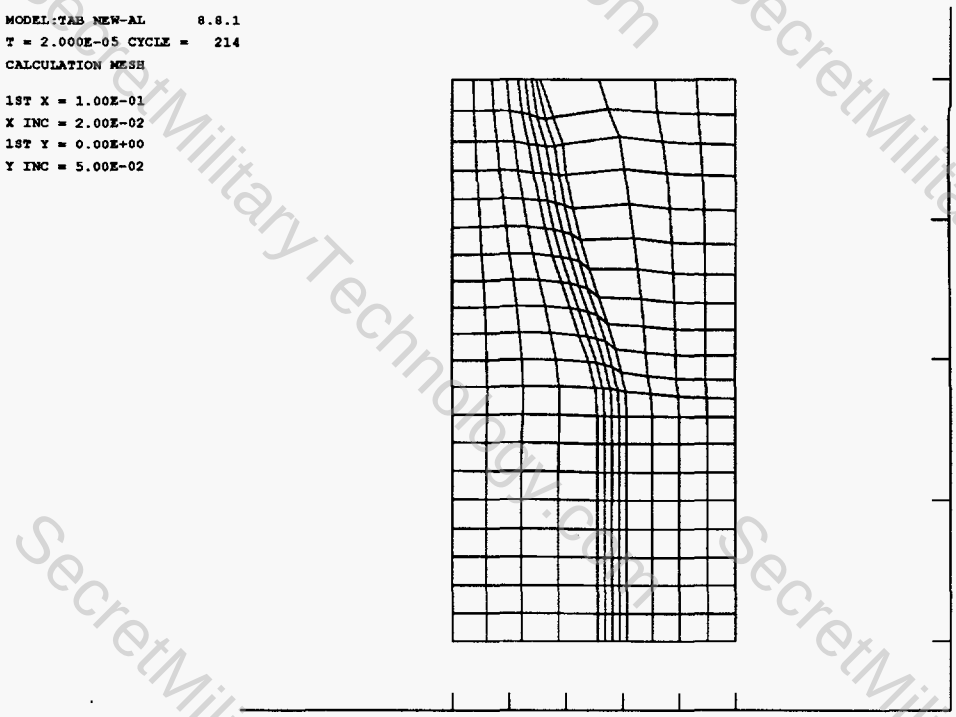

Figure 16 Simulation grid at 20 μs.

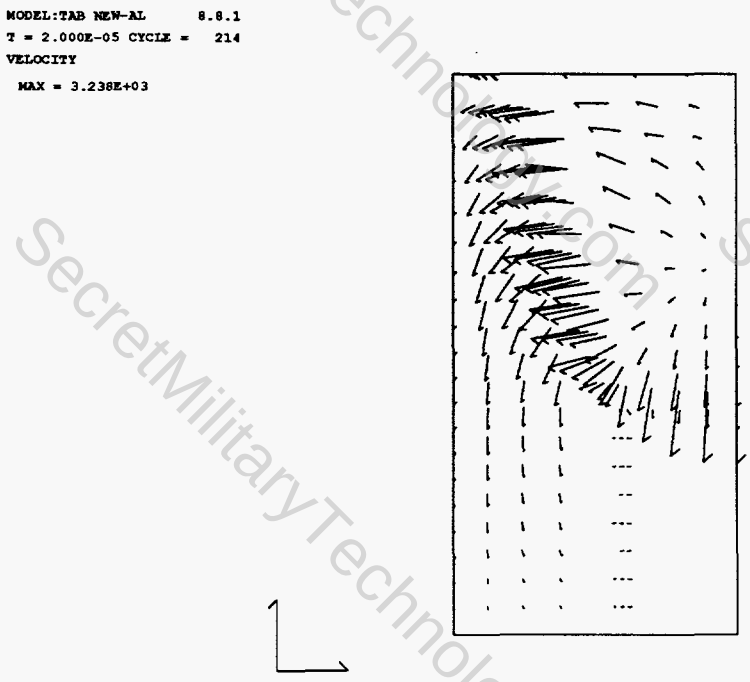

Figure 17 Fluid velocity at 20 μs.

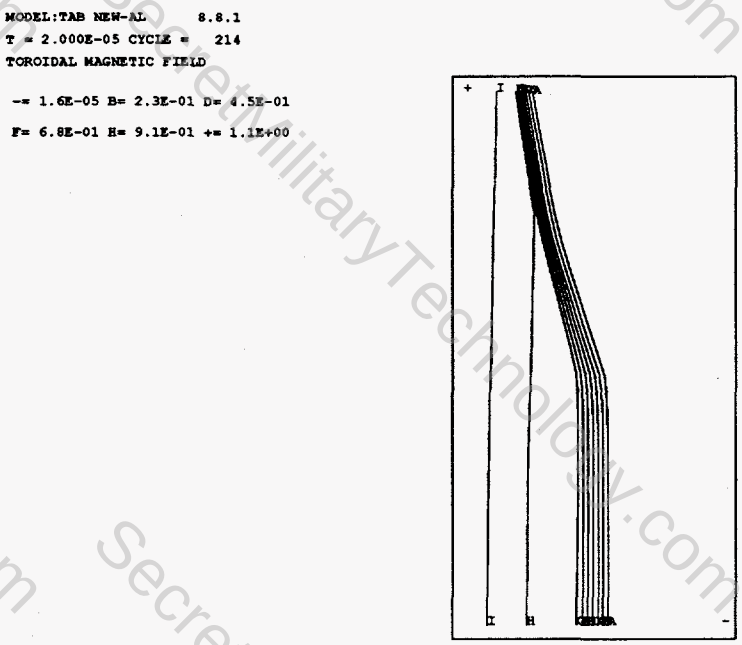

Figure 18 Azimuthal magnetic field at 20 μs.

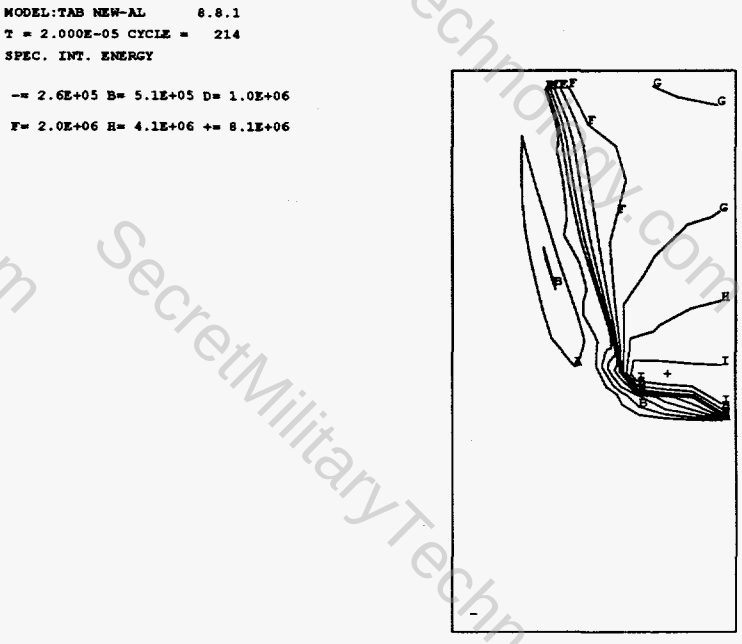

Figure 19 Internal energy density at 20 μs.

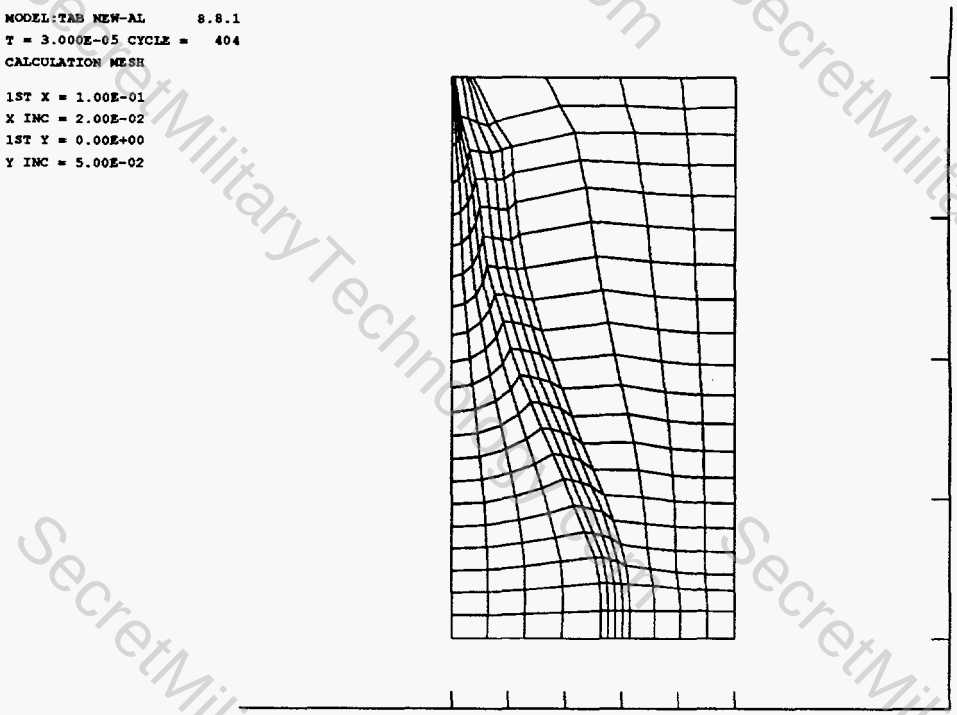

Figure 20 Simulation grid at 30 μs.

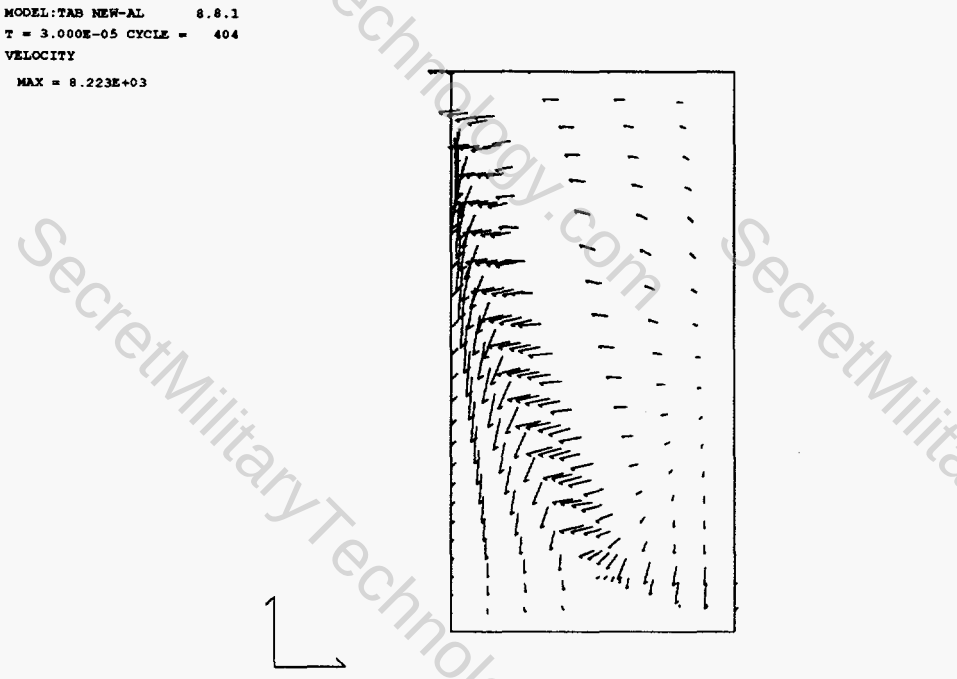

Figure 21 Fluid velocity at 30 μs.

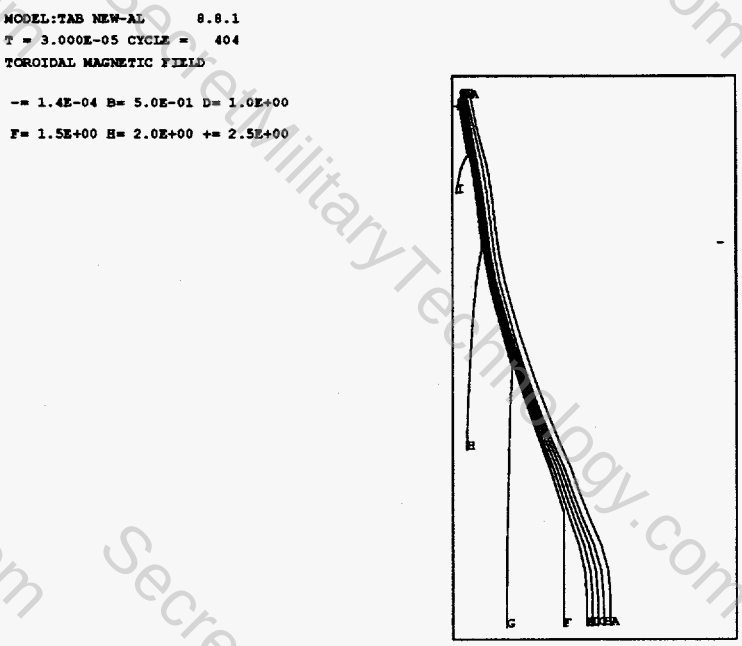

Figure 22 Azimuthal magnetic field at 30 μs.

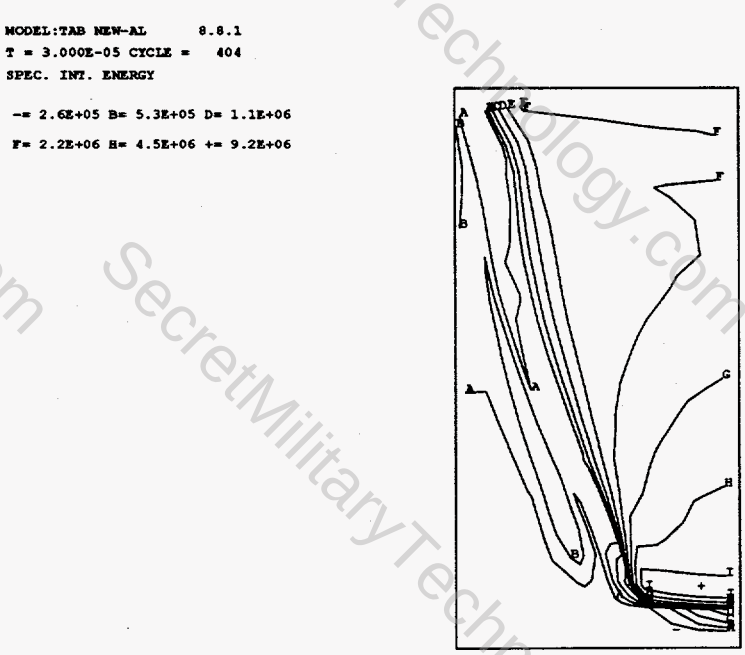

Figure 23 Internal energy density at 30 μs.

5. Conclusions

The dynamic explosive model described here appears to be a useful device for allowing MACH2 to simulate detonations. It has the positive features of being quite easy to institute, use, and modify, and of causing the detonation to act under many of the same physical effects that influence a real detonation. The On the negative side, it is most likely not as accurate as the more sophisticated detonation models that are available, nor does it provide the user with the abundance of controls that more advanced explosive equations-of-state provide.

The capability for designing explosive magnetic flux compression generators that this explosive model gives to MACH2 has been demonstrated by a simple two-dimensional simulation. There are still several areas of work that would make this a more useful and more easily used tool. The most significant of these areas concerns the impact of the armature on the inner conductor. In the simulation presented above, the cells in the void region become very highly compressed as the armature approaches the inner conductor. Research into the relevant physical processes and into an acceptable numerical treatment would be of great benefit in treating the collision. Another area of work that would lead to significant improvements in MACH2's capacity to design such devices is in the way the code handles problems containing several different materials. Its present method is adequate for the simple sort of problem present in the two-dimensional demonstration, but far more interesting problems could be treated if MACH2 were outfitted with an improved multimaterials capability.

Appendix A Listing of One-Dimensional Demonstration Input File

```
Explosive model: 1d sensitivity characterization
 $contrl

    t = 1.0e-11,
    twfn =  28.5e-6,
    imns = 60,
    dt = 1.0e-9,
    dtmax = 1.e-07,

    hydron = .true.,

    meshon = .true.,

    radiate = .false.,
!       radsplit = .false.,
!       radflxlt = .false.,

    thmldif = .false.,
        tdtol = 1.e-4,

    bdiff = .false.,
        rdtol = 1.e-4,
        aresfdg = 0.05,

    ciron = .false.,

    con2on = .true.,

    rofvac = 1.e-1,
    rofjoule = 1.e-1,
    rof = 1.e-5,
    rofsiecp = 1.e-4,
        siecap = 1.e9,

    nsmooth = 4,
    wrelax = 0.25,

!   make the thing slabindrical
    cyl = 0,

    volratm = 0.8,
    courmax = 1.0,
    rmvolrm = 0.2,
    itopt = 20,
    mu = 5.6,
```

```
   donormn = 1.,
   theb = 1.,
   eps = 1.e-3,
   conserv = 0.,

   mglmax = 1,

$end
$output

   dtm = 1.e10,
   dtrst = 10.0e9,
   dto = 1000.0e-9,
   dtp = 2.5e-6,

   dtslic = 2.5e-6,
   ibdyslic = 1,
   lblkslic = 1,
   ijslic = 2,

   intty = 'edits,10',

   intbound = .false.,
   kcon(1) = 11,
   contyp(1) = 'log'
   plot(8) = 'numvis',
   plot(9) = 'joulheat',
   plot(11) = 'sie',
   plot(12) = 'dirke   ',


$end
$ezgeom

   npnts = 4,

   pointx(1)  =  0.00e-2,
   pointy(1)  =  20.00e-2,

   pointx(2)  =  1.00e-2,
   pointy(2)  =  20.00e-2,

   pointx(3)  =  1.00e-2,
   pointy(3)  =  0.00e-2,

   pointx(4)  =  0.00e-2,
   pointy(4)  =  0.00e-2,


   nblk = 1,
   corners(1,1) =   1, 2, 3, 4,
```

```
 $end
 $ezphys

    icellsg = 2,
    jcellsg = 20,
    eosmodlg = "grun",
       ang = 6.,
       awg = 12.,
    roig = 1.894e3, densityg = 1.894e03, csq0g = 16.e6,
!   tempig = 2.5e-2,
    sieig = 2.67e05,
!   gdvlg = 0.95,
    gdvlg = 1.0,
    donorg = .true.,
    radmodlg = "none",

 $end
 $inmesh

     name(5) = '1=expl',
     nigen = 0,
     niter = 3,
     eqvol = 2500.,

     eosmodl(1) = "explosiv",
!    pretend the material is pbx-9502
       siei(1) = 0.2674e06,
       roi(1) = 1.894e03,
       an(1) = 5.598417, aw(1) = 10.980013,
!      tempi(1) = 0.025,
       gm10(1) = 1.37, csq0(1) = 2.e06,
       tfusi(1) = 5.e6, hfusi(1) = 1.e02,
       tvap(1)  = 6.e6, hvap(1) = 1.e02,
       tdiss(1) = 7.e6, hdiss(1) = 1.e02,
       tionize(1) = 8.e6, hion(1) = 1.e02,


 $end
!$modtim

!  tmod = 100.e-9,

!$end
!$inmesh

!   tflow(4,1) = 2.5e-02,
!   velbc(4,1) = 'freeslip',  probc(4,1) = 'wall',

!$end
```

Appendix B Listing of Two-Dimensional Demonstration Input File

```
Explosive model:TAB new-AL
 $contrl

    t = 1.e-11,
    twfn =  50.0e-6,
    imns = 60,
    dt = 1.0e-9,
    dtmax = 1.e-07,

    hydron = .true.,

    meshon = .true.,

    radiate = .false.,
!       radsplit = .false.,
!       radflxlt = .false.,

    thmldif = .false.,
        tdtol = 1.e-4,

    bdiff = .true.,
        rdtol = 1.e-4,
        aresfdg = 0.05,

    ciron = .false.,

    con2on = .true.,

    gdvlmod = .true.,

!   rofvac = 1.,
    rofjoule = 1.,
    rof = 1.e-5,
    rofsiecp = 1.e-4,
        siecap = 1.e9,

    nsmooth = 4,
    wrelax = 0.25,

!   make the thing cylindrical
    cyl = 1,
```

```
   volratm = 0.8,
   courmax = 1.0,
   rmvolrm = 0.2,
   itopt = 20,
   mu = 5.6,
   donormn = 1.,
   theb = 1.,
   eps = 1.e-3,
   conserv = 0.,

   mglmax = 1,

$end
$output

   dtm = 1.e10,
   dtrst = 10.0e9,
   dto = 1000.0e-9,
   dtp = 1.0e-06,
   dtslic = 1000.e-9,
   ibdyslic = 4,
   lblkslic = 3,
   ijslic = 2,

   intty = 'edits,10',

   intbound = .false.,
   kcon(1) = 11,
   contyp(1) = 'log'
   plot(8) = 'numvis',
   plot(9) = 'joulheat',
   plot(11) = 'sie',
   plot(12) = 'dirke   ',

   ncychist = 1,
    histnum = 1,  probtype(1) = 'bzdot',
    histx(1) = 1.05e-2,  histy(1) = 15.e-02,
    histnum = 2,  probtype(2) = 'rbzdot',
    histx(2) = 1.05e-2,  histy(2) = 15.e-02,


$end
$ezgeom

   npnts = 8,

   pointx(1)  =  20.00e-2,
   pointy(1)  =  20.00e-2,

   pointx(2)  =  20.00e-2,
   pointy(2)  =  0.00e-2,
```

```
      pointx(3)  =  16.00e-2,
      pointy(3)  =  0.00e-2,

      pointx(4)  =  16.00e-2,
      pointy(4)  =  20.00e-2,

      pointx(5)  =  15.00e-2,
      pointy(5)  =  20.00e-2,

      pointx(6)  =  15.00e-2,
      pointy(6)  =  0.00e-2,

      pointx(7)  =  10.00e-2,
      pointy(7)  =  0.00e-2,

      pointx(8)  =  10.00e-2,
      pointy(8)  =  20.00e-2,


      nblk = 3,
      corners(1,1) =  4,1,2,3,
      corners(1,2) =  5,4,3,6,
      corners(1,3) =  8,5,6,7,

 $end
 $ezphys

      icellsg = 4,
      jcellsg = 20,
!     tempig = 2.5e-2,
      sieig = 2.67e05,
!     gdvlg = 0.95,
      gdvlg = 1.0,
      donorg = .true.,
      radmodlg = "none",

 $end
 $inmesh

       name(5) = '8.8.3',
       nigen = 0,
       niter = 3,
       eqvol = 2500.,

       eosmodl(1) = "explosiv",
!      pretend the material is pbx-9502
          siei(1) = 0.2674e06,
          roi(1) = 1.894e03,
          an(1) = 5.598417, aw(1) = 10.980013,
!         tempi(1) = 0.025,
          gm10(1) = 1.37, csq0(1) = 2.e06,
```

```
        tfusi(1) = 5.e06, hfusi(1) = 1.e02,
        tvap(1)  = 6.e06, hvap(1) = 1.e02,
        tdiss(1) = 7.e06, hdiss(1) = 1.e02,
        tionize(1) = 8.e06, hion(1) = 1.e02,

!     eosmodl(2) = "grun",
!     the material is aluminum
        tempi(2) = 0.025,
        roi(2) = 2.7e03,
        matname(2) = 'al-new',
        resmodl(2) = 'tabular',
!       an(2) = 13., aw(2) = 26.9815,
!       gm10(2) = 1.136, csq0(2) = 29.e06,
!       tfusi(2) = 5.e06, hfusi(2) = 1.e02,
!       tvap(2)  = 6.e06, hvap(2) = 1.e02,
!       tdiss(2) = 7.e06, hdiss(2) = 1.e02,
!       tionize(2) = 8.e06, hion(2) = 1.e02,

        gdvlm(2) = 0.,
           rmingdv(2) = 0.104,
           delrgdv(2) = 0.001,
        gdvlm(3) = 0.,
           rmingdv(3) = 0.100,
           delrgdv(3) = 0.001,

!     eosmodl(3) = "idealgas",
      matname(3) = 'al-new',
      resmodl(3) = 'tabular',
!     pretend the material is voidium
        roi(3) = 1.e-3,
!       an(3) = 1., aw(3) = 1.,
        tempi(3) = 0.025,

        gdvl(3) = 0.95,
        gdvlb(2,3) = 1., gdvlb(4,2) = 1.,
        gridbc(2,3) = 'fixedgp',

!    fill in initial b-theta field
        binit(1) = 'nocurnt',
        bzi(1) = 0.e-02, rnomfld = 1.e-1,
        binit(2) = 'nocurnt',
!       anomres(2) = .false.,
!       anomres(3) = .false.,
        bzi(2) = 0.e-02,
        binit(3) = 'nocurnt',
        bzi(3) = 1.e0,
!
!

 $end
!$modtim
```

```
!  tmod = 100.e-9,

!$end
!$inmesh

!   tflow(4,1) = 2.5e-02,
!   velbc(4,1) = 'freeslip',  probc(4,1) = 'wall',

!$end
```

Appendix C Listing of MACH2 Modifications File

```
*id v9102
*define unicos
*d mach2.37
      modvers = 'x'
*d eos.14,22
         if(eosmodl(lblk).eq."explosiv")then
            call eosxpl
         else
            if(con2on)call eoslreg
            call eosawan
            if(eosmodl(lblk).eq."idealgas")then
               call eosideal
            else if(eosmodl(lblk).eq."grun")then
               call eosgrun
            else
               call eostable
            end if
         end if

*af,,eostrans
*dk eosxpl
      subroutine eosxpl


c--- explosive model
c--- tests whether material is detonated or undetonated material
c--- if undetonated, tests whether it should detonate, and if it
c--- should, adds the user-determined amount of internal energy
c--- to mimic detonation, then converts material to detonated state
c--- use gruneisen eos for undetonated and detonated material

cdir$ nolist
      include 'common.h'
      include 'inputcom.h'
      include 'pointer.h'
cdir$ list
      pointer(kp095, qold(0:ip2,0:jp2))
      common/explcom/xplcritn, xplsie, siecrit
      data xplcritn/0.1/, xplsie/5.e06/, siecrit/5.348e05/


c--- get new artificial viscosity
       call numvis(mu,cyl)
c--- it is used in the decision to detonate
```

```
      do 100 j = 1, jcels
         do 100 i = 1, icels
c------------detonation state is governed by con2
            if(con2(i,j).lt.0.5)then
c---------------cell contains undetonated explosive
               if(j.eq.jcels)then
c------------------artificial initiator
                  q(i,j) = 1.
                  qold(i,j) = 2.
                  sie(i,j) = siecrit
               end if
c---------------test for detonation
               if(q(i,j) .lt. qold(i,j) .and.
     %            sie(i,j).ge.siecrit)then
c------------------material in this cell is to detonate
                  sie(i,j) = sie(i,j) + xplsie
                  con2(i,j) = 1.
               end if
            end if
c------------- use gruneisen eos
                  lreg(i,j) = nreg(lblk)
                  lregc = lreg(i,j)
                  awc(i,j) = awanmlt(lregc) * aw(lregc)
                  anc(i,j) = awanmlt(lregc) * an(lregc)
c------------------gruneisen equation of state
                  eosfac = pm / qe
                  az = one - tsplit / ( tsplit + tiny )
                  fac = 0.5d0 * ( gm10(lregc) + 2.d0 )
                  dv = one / density(lregc) - one / ro(i,j)

c------------------cold curve
                  sieg = 0.5d0 * csq0(lregc) *
     %             (dv/(one/density(lregc)-fac*dv))**2
                  xyz = one - density(lregc)/ro(i,j)
                  pg = density(lregc)*csq0(lregc)*xyz/
     %                ((one-fac) + fac*density(lregc)/ro(i,j))**2 *
     %                (one - 0.5*gm10(lregc)*xyz)

c------------------bracket fusion
                  ef = sieg + tfusi(lregc) /
     %                ( awc(i,j) * eosfac * gm10(lregc) )
                  ef2 = ef + hfusi(lregc)

c------------------bracket vaporivation
                  ev = ef2 + ( tvap(lregc)  - tfusi(lregc) ) /
     %              ( awc(i,j) * eosfac * gm11(lregc)  )
                  ev2 = ev + hvap(lregc)

c------------------bracket disassociation
                  ed = ev2 + ( tdiss(lregc) - tvap(lregc) )  /
     %              ( awc(i,j) * eosfac * gm1(lregc)  )
```

```
                  ed2 = ed + hdiss(lregc)

c-----------------bracket ionization
                  ei = ed2 + ( tionize(lregc)  - tdiss(lregc) ) /
     %              ( awc(i,j) * eosfac * gm1(lregc)   )
                  ei2 = ei + hion(lregc)

c-----------------beyond this, assume the gas is ionized
c----------------- we really need something like a SAHA model here

c-----------------near ionization
                  if (sie(i,j) .gt. ei2) then
                     te(i,j) = tionize(lregc) +
     %                  eosfac*gm1(lregc)*(sie(i,j)-ei2)*awc(i,j) /
     %                  ( az + nfe(i,j) )
                     dtde(i,j) = eosfac*gm1(lregc)*awc(i,j) /
     %                   ( az + nfe(i,j) )
                     p(i,j) = pg+gm1(lregc)*ro(i,j)*sie(i,j)
                     csq(i,j) = (gm1(lregc)+one)*gm1(lregc)*sie(i,j)
                  endif
                  if(sie(i,j).gt.ei.and.sie(i,j).le.ei2)then
                     te(i,j) = tionize(lregc)
                     dtde(i,j) = tiny
                     p(i,j) = pg + gm1(lregc)*ro(i,j)*sie(i,j)
                     csq(i,j) = (gm1(lregc)+one)*gm1(lregc)*sie(i,j)
                  endif

c-----------------near disassociation
                  if(sie(i,j).gt.ed2.and.sie(i,j).le.ei)then
                     te(i,j) = tdiss(lregc) + az *
     %                  eosfac*gm1(lregc)*(sie(i,j)-ed2)*awc(i,j)
                     dtde(i,j) = eosfac * gm1(lregc) * awc(i,j)
                     p(i,j) = pg + gm1(lregc)*ro(i,j)*sie(i,j)
                     csq(i,j) = (gm1(lregc)+one)*gm1(lregc)*sie(i,j)
                  endif
                  if(sie(i,j).gt.ed.and.sie(i,j).le.ed2)then
                     te(i,j) = tdiss(lregc)
                     dtde(i,j) = tiny
                     p(i,j) = pg + gm1(lregc) * ro(i,j) * sie(i,j)
                     csq(i,j) = (gm1(lregc)+one)*gm1(lregc)*sie(i,j)
                  endif

c-----------------near vaporization
                  if(sie(i,j).gt.ev2.and.sie(i,j).le.ed)then
                     te(i,j) = tvap(lregc) + az *
     %                  eosfac*gm1(lregc)*(sie(i,j)-ev2)*awc(i,j)
                     dtde(i,j) = eosfac * gm1(lregc) * awc(i,j)
                     p(i,j) = pg + gm1(lregc) * ro(i,j) * sie(i,j)
                     csq(i,j) = (gm1(lregc)+one)*gm1(lregc)*sie(i,j)
                  endif
                  if(sie(i,j).gt.ev.and.sie(i,j).le.ev2)then
```

```
                    te(i,j) = tvap(lregc)
                    dtde(i,j) = tiny
                    p(i,j) = pg + gm1(lregc) * ro(i,j) * sie(i,j)
                    csq(i,j) = (gm1(lregc)+one)*gm1(lregc)*sie(i,j)
                  endif

c-----------------near fusion
                  if(sie(i,j).gt.ef2.and.sie(i,j).le.ev)then
                    te(i,j) = tfusi(lregc) + az *
     %                 eosfac*gm11(lregc)*(sie(i,j)-ef2)*awc(i,j)
                    dtde(i,j) = eosfac * gm11(lregc) * awc(i,j)
                    p(i,j) = pg+gm11(lregc)*ro(i,j)*sie(i,j)
                    csq(i,j) = csq1(lregc)
                  endif
                  if(sie(i,j).gt.ef.and.sie(i,j).le.ef2)then
                    te(i,j) = tfusi(lregc)
                    dtde(i,j) = tiny
                    p(i,j) = pg + gm11(lregc)*ro(i,j)*sie(i,j)
                    csq(i,j) = csq1(lregc)
                  endif

c-----------------solid
                  if(sie(i,j).ge.sieg.and.sie(i,j).le.ef)then
                    te(i,j) = eosfac * az *
     %                 gm10(lregc) * (sie(i,j)-sieg) * awc(i,j)
                    dtde(i,j) = eosfac * gm10(lregc) * awc(i,j)
                    p(i,j) = pg + gm10(lregc)*ro(i,j)*sie(i,j)
                    csq(i,j) = csq0(lregc)
                  endif

                  if(tsplit.eq.0)then
                    ti(i,j) = eosfac*gm1(lregc)*
     %                          (sieion(i,j)-sieg)*awc(i,j)
                    dtide(i,j) = eosfac * gm11(lregc) * awc(i,j)
                    pion(i,j) = pg
                  endif
  100   continue

c--- save old artificial viscosity
        do 50 j = 1, jcels
           do 50 i = 1, icels
              qold(i,j) = q(i,j)
   50   continue

        return
        end

*d eosmat.19,20
       if (eosmodl(lblk) .eq. 'idealgas'  .or.
     %     eosmodl(lblk) .eq. 'grun'      .or.
     %     eosmodl(lblk) .eq. 'explosiv'    ) then
```

```
*d eosmat.46,47
      elseif (eosmodl(lblk) .eq. 'idealgas'  .or.
     %        eosmodl(lblk) .eq. 'grun'       .or.
     %        eosmodl(lblk) .eq. 'explosiv'    ) then
*d eossie.34
      elseif (eosmodl(lblk) .eq. 'grun'  .or.
     %        eosmodl(lblk) .eq. 'explosiv') then
*d eossie.94
      elseif (eosmodl(lblk) .eq. 'grun'  .or.
     %        eosmodl(lblk) .eq. 'explosiv') then
```