ARMY RESEARCH LABORATORY

Airblast Loading Model for DYNA2D and DYNA3D

by Glenn Randers-Pehrson
and Kenneth A. Bannister

ARL-TR-1310 March 1997

Approved for public release; distribution is unlimited.

Abstract

We incorporated the CONWEP blast model into DYNA2D and DYNA3D. It works as expected and appears to be adequate for modeling problems such as vehicle response to land mines. The model accounts for the angle of incidence of the blast wave, but it does not account for shadowing by intervening objects or for confinement effects. This report provides FORTRAN listings and directions for incorporating the model in DYNA2D, DYNA3D, and associated preprocessors and postprocessors and suggests changes to the user manuals.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTENTIONALLY LEFT BLANK.

1. INTRODUCTION

DYNA2D [1] and DYNA3D [2] are general purpose finite-element programs that are capable of modeling large deformation of structures. In applications such as blast loading of the floor of a vehicle by a land mine, it is desirable to have a simple blast-loading model rather than having to explicitly simulate the progress of the shock wave from the high explosive through the air and its interaction with the structure.

CONWEP [3] is an implementation of the empirical blast models of Kingery and Bulmash [4, 5]. In the present work, the CONWEP model was incorporated into DYNA2D and DYNA3D and their interactive graphics preprocessors, MAZE [6] and INGRID [7].

The resulting model was tested by simulating a blast-loading experiment reported by Pytleski and Catherino [8] and some near-field blast-impulse experiments by Ewing and Huffington [9]. The results indicate that the DYNA2D/CONWEP and DYNA3D/CONWEP models are adequate for use in engineering studies of vehicle response to the blast from land mines.

2. OBJECTIVES

The objective of this work was to provide a convenient means of applying blast loading on structures in the DYNA2D and DYNA3D codes, without having to run the explosive blast problem explicitly. Analytic formulas fit to blast data were available from the work of Kingery and Bulmash [4, 5], and FORTRAN subroutines implementing these formulas were available in Hyde's CONWEP computer program [3].

3. IMPLEMENTATION

DYNA3D already has a blast-loading capability in which Brode functions can be applied to structures to simulate blast from a nuclear weapon. We extended the Brode loading function to incorporate the CONWEP algorithms. The meaning of the DYNA2D/DYNA3D "IBRODE" parameter was extended to be:

0: no blast load

1: Brode function

2: CONWEP loading function

The CONWEP blast-loading algorithms were extracted from the CONWEP package and modified where necessary to conform to FORTRAN 77 syntax. These FORTRAN subroutines are listed in Appendix A. These algorithms provide the time of arrival, peak reflected pressure, peak incident pressure, and duration and decay of the incident and reflected pressure.

Input data required by the CONWEP model are:

- weight: equivalent mass of TNT, in problem mass unit
- x0, y0, z0: coordinates of the point of explosion, in problem length units
- t0: delay time between when the DYNA problem starts and the instant of explosion, in problem time unit. Can be negative.
- nunit: units switch:

  1: pounds, feet, psi, seconds

  2: kilograms, meters, pascals, seconds

  3: dozens of slugs, inches, psi, seconds

  4: grams, centimeters, Megabars, microseconds

  5: conversion factors supplied by the user
- isurf: surface or air blast switch

  1: surface blast

  2: air blast

In addition, the DYNA2D/DYNA3D model requires a list of the surface segments that will experience the blast loading. This is done in the same manner as with the Brode model, except that the load curve number is -2 instead of -1.

The Brode model does not account for the angle of incidence of the blast wave on the surface of the structure. The CONWEP algorithms do account for angle of incidence by combining the reflected pressure (normal-incidence) value and the incident pressure (side-on incidence) value. Accordingly, we modified the DYNA2D/DYNA3D blast-loading model so that it can calculate the angle of incidence and then take the sum .

$$\text{PressureLoad} = \text{ReflectedPressure} \cdot \cos^2\theta$$
$$+ \text{IncidentPressure} \cdot (1 + \cos^2\theta - 2\cos\theta)$$

When $\cos\theta$ is negative (i.e., the surface is not facing the point of explosion), then

$$\text{PressureLoad} = \text{IncidentPressure},$$

but the arrival time and the incident pressure are not adjusted in any way to account for shadowing by the intervening structure.

The DYNA2D/DYNA3D blast-loading model does not take into account any confinement or tunnel effects and should not be used for analyzing such problems.

Except for the addition of a number of subroutines, all of whose names begin with "conwep_", only two DYNA2D subroutines and two DYNA3D subroutines were modified. These are DYNA2D's subroutine dynai and subroutine fe2a, and DYNA3D's subroutine dynai and subroutine load. These modifications are shown in Appendices B and C. We modified DYNA2D's graphics preprocessor, MAZE, to read and write CONWEP data. The subroutines that were modified are blkdat, meshol2, and meshol3, and we added subroutines brodin and ndbrod. The changes are shown in Appendix D. We also modified DYNA3D's graphics preprocessor, INGRID, similarly. The modified subroutines are dnopts, brodin, and ndbrod, which are shown in Appendix E. Revisions and additions to the DYNA2D, DYNA3D, MAZE, and INGRID manuals are provided in Appendices F, G, H, and I, respectively.

4. DISCUSSION

4.1 Modeling of Blast-Loading on a Heavy Plate. To test the implementation of the model in DYNA3D, we ran a problem in which a 2.32-kg charge was detonated as a surface blast at a standoff of 45.72 cm from a heavy plate. The plate was modeled as a very dense fluid (20,000 g/cm^3) so that its acceleration would be small and proportional to the applied pressure load and that its velocity would also be small and proportional to the impulse. Figure 1 shows the pressure contours on the plate at 1.1 ms after initiation (multiply the velocity by 1,000 to determine the impulse in Pascal-seconds). Figure 2 shows the pressure-time histories at 10-cm increments of radial distance from "ground zero," and Figure 3 shows the impulse-time histories at the same locations. The INGRID input file for this problem is given in Appendix J.

4.2 Comparison with Experiments.

4.2.1 Near-Field Blast Impulse on a Heavy Plate. We also ran a series of problems with DYNA2D, using experimental conditions described in Huffington and Ewing [9]. They measured the reflected impulse near spherical charges of Pentolite by measuring the velocity of a steel plug that was accelerated by the blast impulse.

In DYNA2D, the table was modeled as a very dense fluid (10^6 g/cm^3), such that the nodal acceleration would be proportional to the pressure load and the nodal velocity proportional to the impulse. The simulations were run until acceleration was complete and total impulse could be determined. A MAZE input file for one of the problems is given in Appendix K.

The initial runs revealed a problem with the calculation of positive phase duration when the scaled distance was less than the range of applicability claimed by Kingery and Bulmash [4, 5]. Accordingly, we modified the "conwep_tdur" subroutine to use a constant scaled duration in such cases.

The comparisons between the DYNA2D/CONWEP results and the experimental results are shown in Table 1 and Figure 4.

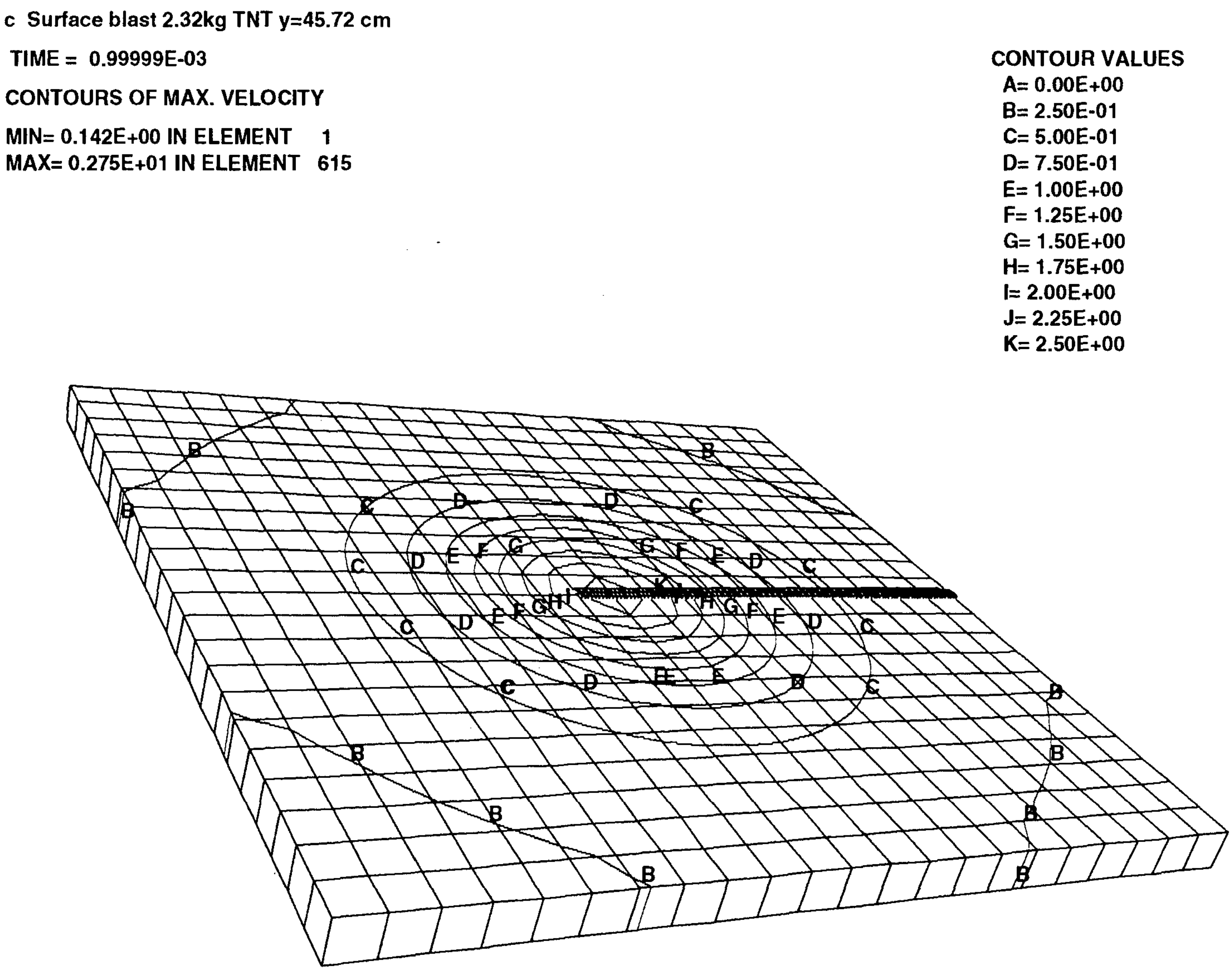

Figure 1. Contours of velocity (proportional to impulse) from sample problem.

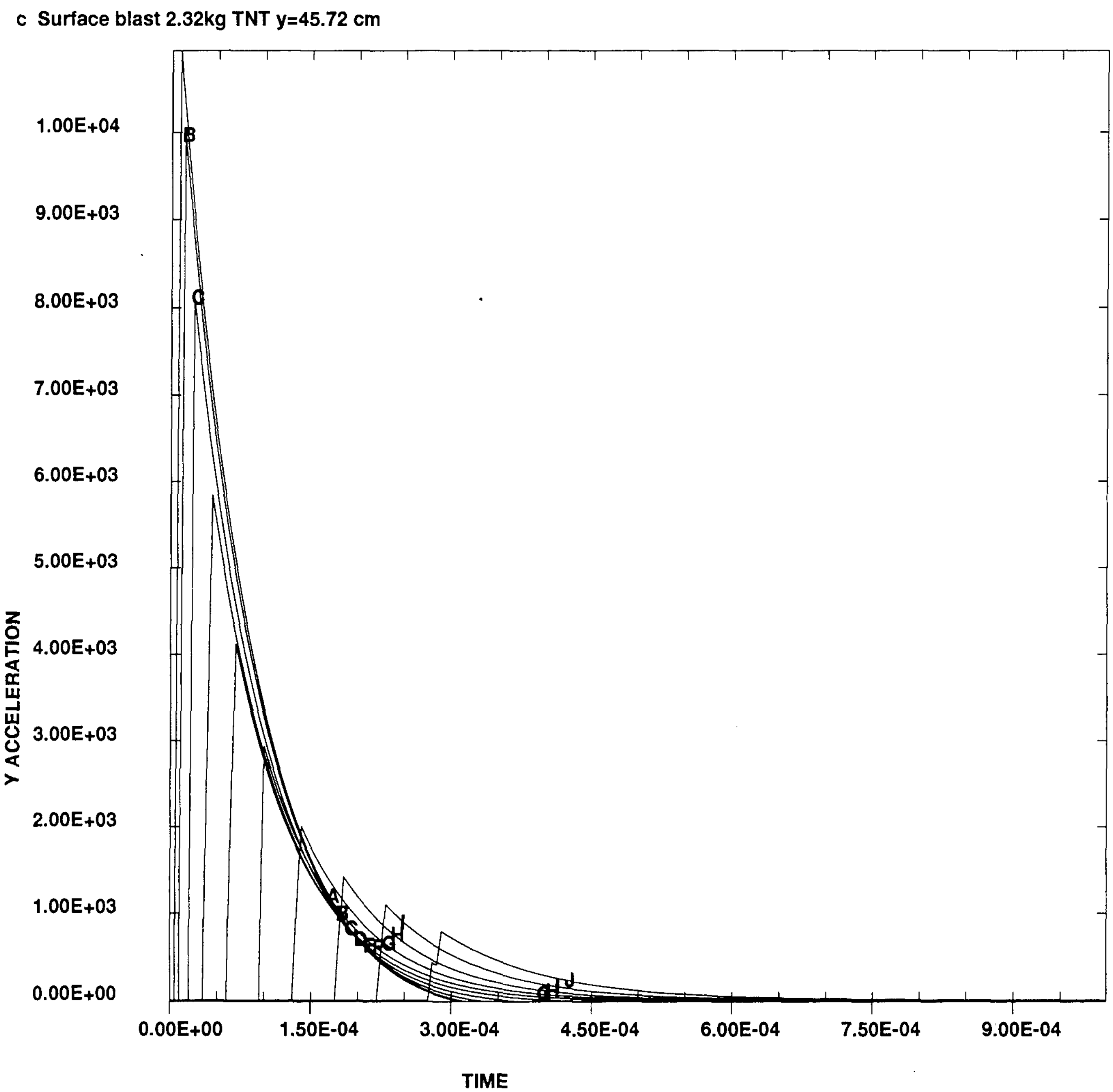

Figure 2. Pressure vs. time from sample problem.

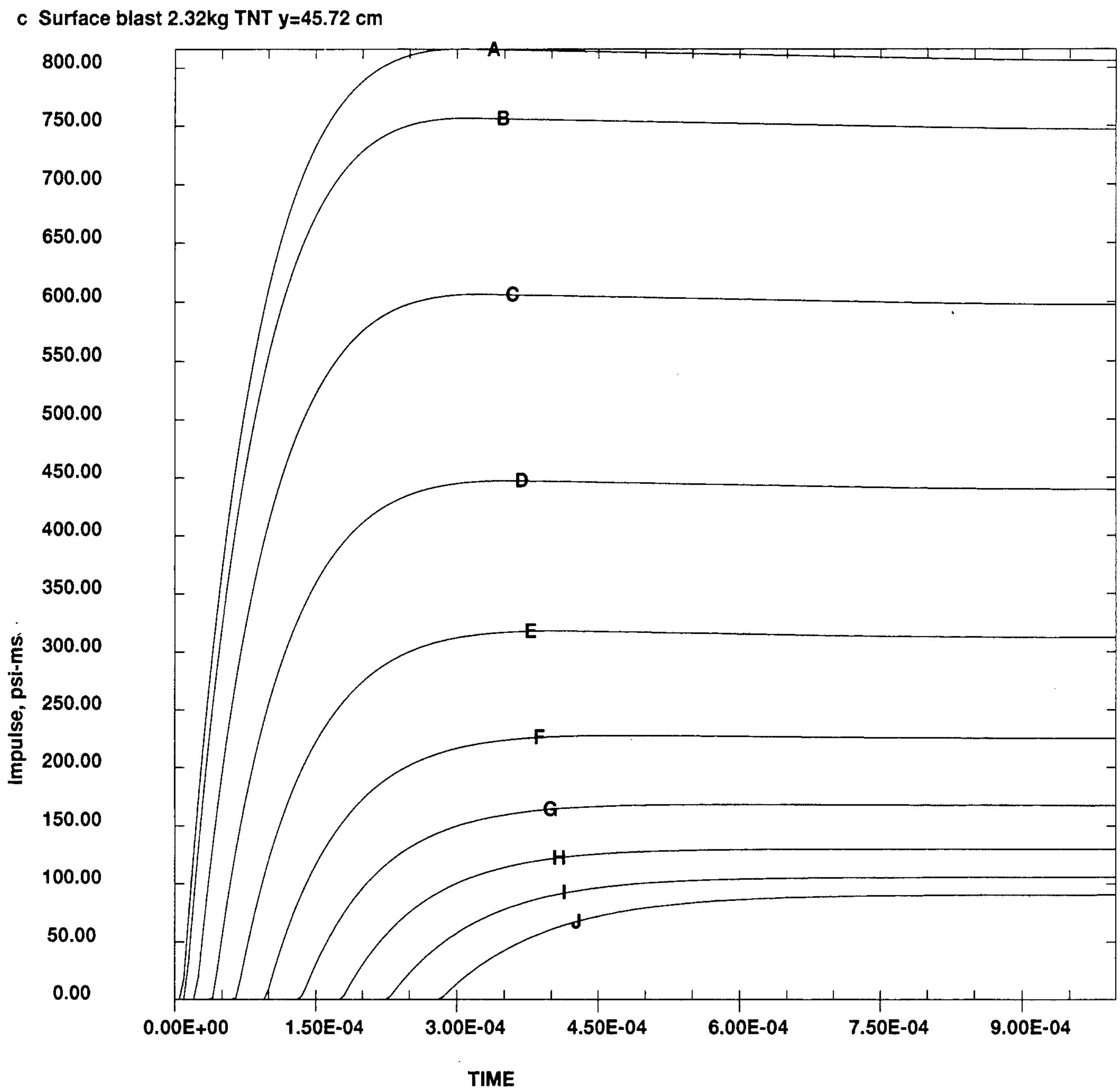

Figure 3. Impulse vs. time from sample problem.

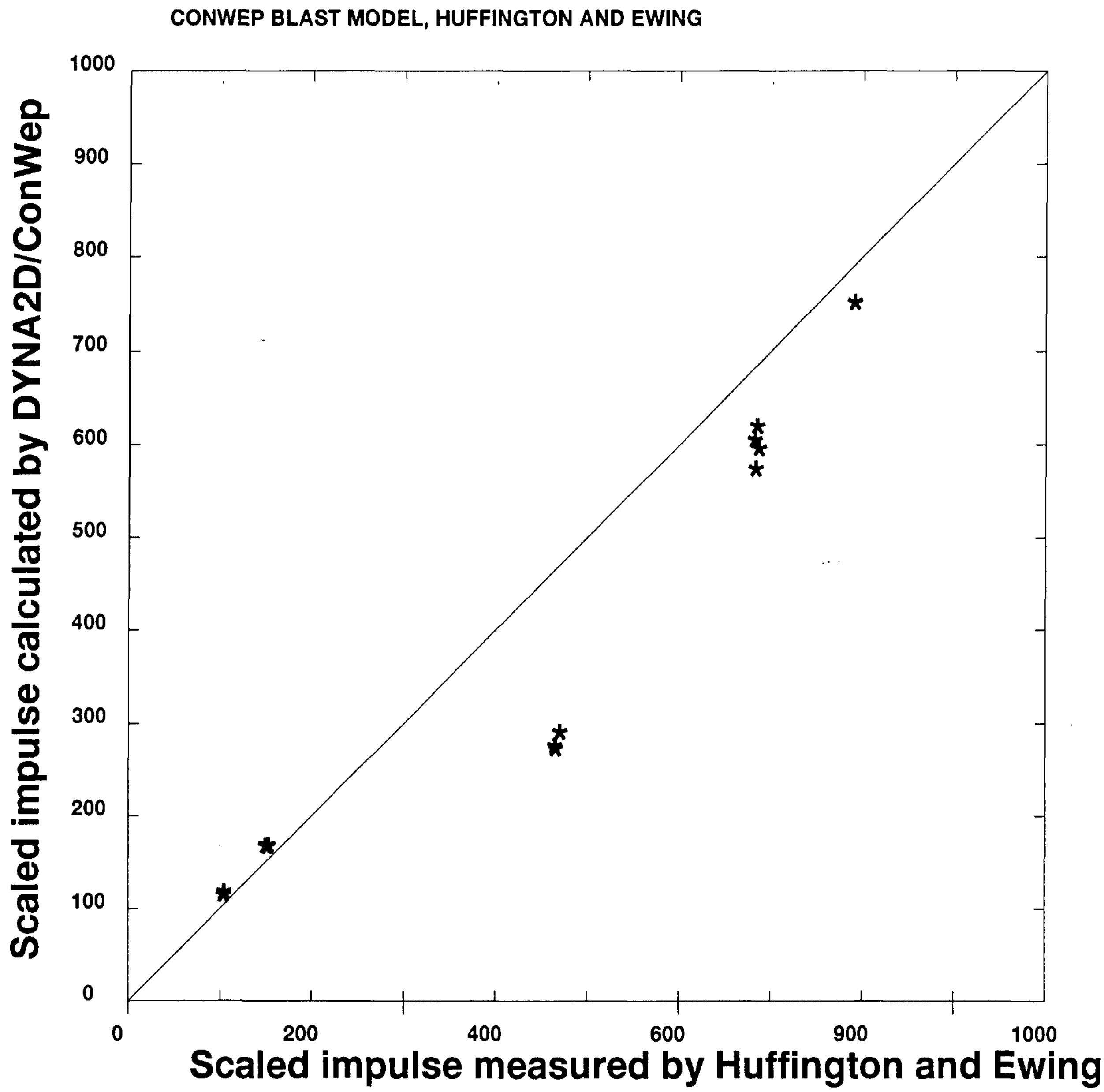

Figure 4. Comparison of DYNA2D/CONWEP model with experimental results by Huffington and Ewing [9].

Table 1. Comparison of DYNA2D/CONWEP Model with Experimental Results by Huffington and Ewing [9].

Test	Scaled Distance $ft/lb^{1/3}$	Scaled Impulse (DYNA2D/CONWEP)	Scaled Impulse [9]
T35	0.48	101.	101.
T66	0.48	100.	98.
T40	0.38	147.	151.
T72	0.38	149.	150.
T45	0.29	462.	256.
T100	0.29	149.	152.
T79	0.29	461.	259.
T107	0.28	467.	274.
T84	0.19	680.	557.
T95	0.19	679.	588.
T114	0.10	682.	603.
T111	0.19	683.	579.
T89	0.16	787.	734.

4.2.2 Dynamic Deflection of an Aluminum Plate. The DYNA3D/CONWEP model was "beta tested" by Ashutash Jhaveri and John Condon. They used it to simulate experimental data reported in 1993 by Pytleski and Catherino [8]. The DYNA3D/CONWEP predictions were in excellent agreement with the experimental results. Jhaveri and Condon's results were presented by Condon, Gniazdowski, and Gregory [10], who concluded that the DYNA3D/CONWEP model is sufficiently accurate for their study of the response of a vehicle floor panel to blast from a land mine.

5. CONCLUSIONS

The CONWEP blast model has been incorporated into DYNA2D and DYNA3D. It works as expected and appears to be adequate for modeling problems such as vehicle response to land mines. The model accounts for the angle of incidence of the blast wave, but it does not account for shadowing by intervening objects or for confinement effects.

6. REFERENCES

1. Whirley, R. G., B. E. Englemann, and J. O. Hallquist. "DYNA2D: A Nonlinear, Explicit, Two-Dimensional Finite Element Code for Solid Mechanics User Manual." UCRL-MA-110630, Lawrence Livermore National Laboratory, Livermore, CA, April 1992.

2. Whirley, R. G., B. E. Englemann, and J. O. Hallquist. "DYNA3D: A Nonlinear, Explicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics -- User Manual." UCRL-MA-107524 (rev. 1), Lawrence Livermore National Laboratory, Livermore, CA, November 1993.

3. Hyde, D. W. "User's Guide for Microcomputer Program CONWEP, Applications of TM 5-855-1, 'Fundamentals of Protective Design for Conventional Weapons'." SL-88-1, U.S. Army Corps of Engineers Waterways Experiment Station Instruction, Vicksburg, MS, April 1988, revised February 1993.

4. Kingery, C. N., and G. Bulmash. "Air-Blast Parameters from TNT Spherical Air Burst and Hemispherical Surface Burst." ARBRL-TR-02555, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, April 1984.

5. Department of the Army. "Fundamentals of Protective Design for Conventional Weapons." TM 5-855-1, Washington, D.C., November 1986.

6. Hallquist, J. O. "MAZE -- An Input Generator for DYNA2D and NIKE2D." UCID-19029 (rev. 2), Lawrence Livermore National Laboratory, Livermore, CA, June 1983.

7. Christon, M. A., D. Dovey, D. W. Stillman, J. O. Hallquist, and R. B. Rainsberger. "INGRID -- A 3-D Mesh Generator for Modeling Nonlinear Systems -- User Manual." UCRL-MA-109790 (draft), Lawrence Livermore National Laboratory, Livermore, CA, September 1992.

8. Pytleski, J. L., and H. Catherino. "Lightweight Hull Floor Program." ARL-CR-58 (prepared by General Dynamics Land Systems Division, Warren, MI), U.S. Army Research Laboratory, Watertown, MA, May 1993.

9. Huffington, N. J., Jr., and W. O. Ewing. "Reflected Impulse Near Spherical Charges." BRL-TR-2678, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, September 1985.

10. Condon, J. A., N. Gniazdowski, and F. H. Gregory. "The Design, Testing, and Analysis of a Proposed Composite Hull Technology Mine-Blast-Resistant Vehicle Floor Panel." ARL-TR-796, U. S. Army Research Laboratory, Aberdeen Proving Ground, MD, July 1995.

INTENTIONALLY LEFT BLANK.

APPENDIX A:

CONWEP SUBROUTINES, CONVERTED TO FORTRAN 77

INTENTIONALLY LEFT BLANK.

```
      subroutine conwep_angle(x1,x2,x3,x4,y1,y2,y3,y4,
     &                z1,z2,z3,z4,s,x5,y5,z5,cosa)

c     Find the incidence angle from x0,y0,z0 to the face that
c     has corners (x1,y1,z1) ... (x4,y4,z4).
c     Glenn Randers-Pehrson, 22 April 1994

c     "s" is the load curve multiplication factor.  If negative,
c     assume that the nodes are numbered in reverse order

      common /conwep_input/wtnt,x0,y0,z0,t0,isurf

c     Find vectors connecting opposite pair of corners
c     Call the vectors "a" and  "b"

      ax = x4 - x2
      ay = y4 - y2
      az = z4 - z2

      bx = x3 - x1
      by = y3 - y1
      bz = z3 - z1

c     Take cross product
c     Call the result a vector "e"

      ex = by*az - ay*bz
      ey = bz*ax - az*bx
      ez = bx*ay - ax*by

c     Normalize.  The unit vector points out of the surface

      emag = SQRT(ex*ex + ey*ey + ez*ez)
      if(emag.eq.0.)emag=1.
      ex = ex/emag
      ey = ey/emag
      ez = ez/emag

c     Find centroid

      if(x3.eq.x4 .and. y3.eq.y4 .and. z3.eq.z4)then
       x5=(x1+x2+x3)/3.
       y5=(y1+y2+y3)/3.
       z5=(z1+z2+z3)/3.
      else
```

```
        x5=(x1+x2+x3+x4)/4.
        y5=(y1+y2+y3+y4)/4.
        z5=(z1+z2+z3+z4)/4.
      endif

c     Find vector from centroid of face to center of blast
c     Call the result a vector "f"

      fx = x0 - x5
      fy = y0 - y5
      fz = z0 - z5

c     Normalize.  The unit vector points toward the center of blast

      fmag = SQRT(fx*fx + fy*fy + fz*fz)
      if(fmag.eq.0.)fmag=1.
      fx = fx/fmag
      fy = fy/fmag
      fz = fz/fmag

c     find angle between vectors "e" and "f"

      edotf = ex*fx + ey*fy + ez*fz
      cosa = MIN( MAX (edotf,-1.0), 1.0 )
      if (s.le.0.) cosa = -cosa

      end


      subroutine conwep_blast(x1,x2,x3,x4,y1,y2,y3,y4,
     &                        z1,z2,z3,z4,s,t,p)
c
      common /conwep_input/wtnt,x0,y0,z0,t0,isurf
c
      common /conwep_units/ iunit,uft,ulb,upsi,ums
      real*8 a, b

      logical warn
      save warn
      data warn/.true./
c
c   This subroutine invokes Conwep logic to compute the pressure
c   load P on a surface at a field point (x,y,z), due to a conventional
c   TNT charge undergoing a hemispherical surface burst (isurf=1) or
c   spherical air burst (isurf=2) located  at point (x0,y0,z0) and
```

```
c  time t0 relative to the start of a DYNA3D simulation.  P is
c  returned in psi, Pascals, or Megabars, depending on the value
c  of nunit.
c
c  The user must supply the following as input to DYNA3D, subroutine
c  conwep_blasti:
c
c  weight     - Equivalent weight or mass of TNT, in lbs, Kilograms,
c               dozens of slugs, or grams, respectively.
c  (x0,y0,z0) - Coordinates of point of explosion w.r.t. global
c               origin, in feet, meters, inches, or cm
c  t0         - Delay time, if any, between when problem
c               starts and instant of explosion, seconds or microseconds
c  nunit      - Units switch
c                = 1, engineering units (feet, pounds, psi, seconds)
c                = 2, S.I. units (meters, kilograms, pascals, seconds)
c                = 3, English units (inches, dozens of slugs, psi, seconds)
c                = 4, metric units (cm, grams, Megabars, microseconds)
c                = 5, conversion factors are supplied by the user
c
c  Also, to activate this logic, the user must use NL = -2 when
c  defining the pressure load input to DYNA3D.
c
c  The equations from BRL Technical Report ARBRL-TR-02555 are used
c  to find the incident pressure P, time of arrival ta, time of pulse
c  duration to, impulse impo, peak incident overpressure pso,
c  reflected impulse impr, and peak reflected overpressure pro for
c  a field point at slant range R from the point of explosion
c  (x0,y0,z0) for a conventional charge of TNT.  The peak incident
c  (side-on) and peak reflected (head-on) pressures, together with
c  the computed incidence angle alpha, are used to determine the
c  pressure load on the face.
c
c  Description of variables (units given are as used internally):
c
c   a      - decay coefficient of incident pressure vs. time history,
c             where  p(t) = pso*(1-t/to)*exp(-a*t/to)
c            This parameter is computed in function decay.
c   b      - decay coefficient of reflected pressure vs. time history,
c             where  p(t) = pro*(1-t/to)*exp(-b*t/to)
c            This parameter is computed in function decay.
c   cosa    - cos(incidence angle) (1 = head on, 0 = side on)
c   isurf  - 1 = surface burst, 2 = air burst (set to 1 currently)
c   P      - incident pressure at field point (x,y,z,t) and
c            time t; computed in routine pressure, psi
```

```
c  pro    - peak reflected overpressure, psi
c          computed in routine params by calling function pref
c  pso    - peak incident overpressure, psi
c          computed in routine params by calling function pinc
c  R      - slant range, = sqrt((x-x0)**2+(y-y)**2+(z-z0)**2), ft
c  s      - multiplication factor.  If negative, assume nodes are
c          numbered in reverse order
c  t      - current DYNA3D time
c  ta     - arrival time, msec
c          Computed by routine params by use of function tarr.
c  to     - positive phase duration time, msec
c          Computed by routine params by use of function tdur.
c  ts     - shifted time, = t - t0
c  wtnt   - equivalent weight of TNT, lb
c  x,y,z  - coordinates of field point w.r.t global origin where
c          incident pressure is to be computed, default unit
c          is meters
c  zlant  - scaled slant range, ft/lb**1/3
c          Computed in routine params.
c  zlog   - logarithm of scaled slant range
c          Computed in routine params.
c
c Compute centroid of face and incidence angle
c
      call  conwep_angle(x1,x2,x3,x4,y1,y2,y3,y4,z1,z2,z3,z4,
     &                s,x,y,z,cosa)
c
c Compute slant range (distance between field point and the point of
c explosion).  Convert to appropriate units.
c
      R = sqrt((x-x0)**2 + (y-y0)**2 + (z-z0)**2)*uft
c
c Compute pso, ta, to, zlant, and zlog for this point.
c
      call conwep_params(isurf,R,ta,to,pso,pro,wtnt,zlant,zlog,a,b)
c
c Get shifted time ts and express it in terms of milliseconds for
c internal use.
c
      ts = (t - t0)*ums

      if( (ta .lt. -t0*ums) .and. warn) then
        write(*,*)' Caution: arrival time < 0 in conwep_blast'
        write(*,*)'        ta =',ta
```

```
      if (t.ne.0.)warn=.false.
     endif
c
c  Compute the incident pressure P(x,y,z,ts).
c
     call conwep_press(a,b,ts,P,ta,to,cosa,pro,pso)
c
c  Convert pressure to appropriate units before returning to
c  calling program.
c
     p = P/upsi

     return
     end




     SUBROUTINE conwep_blasti
C
C    READ AND PRINT BLAST FUNCTION DATA FOR USE IN SUBROUTINE
C    CONWEP_BLAST (CALLED BY SUBROUTINE FE2A)

c  weight     - Equivalent weight or mass of TNT, in lbs, Kilograms,
c               dozens of slugs, or grams, respectively.
c  (x0,y0,z0) - Coordinates of point of explosion w.r.t. global
c               origin, in feet, meters, inches, or cm
c  t0         - Delay time, if any, between when problem
c               starts and instant of explosion, seconds or microseconds
c  nunit      - Units switch (default 2)
c                = 1, engineering units (feet, pounds, psi, seconds)
c                = 2, S.I. units (meters, kilograms, pascals, seconds)
c                = 3, English units (inches, dozens of slugs, psi, seconds)
c                = 4, metric units (cm, grams, Megabars, microseconds)
c                = 5, conversion factors are supplied by the user
c  isurf      - Surface or Air Blast switch (default 2)
c                = 1, surface blast
c                = 2, air blast
c
     common /conwep_input/wtnt,x0,y0,z0,t0,isurf
     common /conwep_units/ iunit,uft,ulb,upsi,ums

     CHARACTER*80 TXTS,MSSG
C
     CALL GTTXSG (TXTS,LCOUNT)
```

```
      READ (UNIT=TXTS,FMT=100,ERR=90)
     1 WEIGHT,X0,Y0,Z0,T0,nunit,nsurf

      iunit = nunit
      if(iunit.eq.0)iunit=2

      isurf = nsurf
      if(isurf.eq.0)isurf=2

      if(iunit.eq.5)then
        CALL GTTXSG (TXTS,LCOUNT)
        READ (UNIT=TXTS,FMT=100,ERR=90)uft,ulb,upsi,ums
      else
        call conwep_stunit
      endif
c     convert weight to appropriate units for internal use.
      wtnt = weight*ulb
c

      WRITE(12,110) WEIGHT,WTNT,X0,Y0,Z0,T0,IUNIT,ISURF
      WRITE(12,120) ulb,uft,ums,upsi
C
      RETURN
C
C     TERMINATE DUE TO BADLY FORMATTED DATA
C
   90 MSSG=' BLASTI: ERROR READING BLAST INPUT DATA'
      CALL TERMIN (TXTS,MSSG,LCOUNT,1)
C
  100 FORMAT(5E10.0,2i5)
  110 FORMAT(//' C O N W E P   O V E R P R E S S U R E',
     1       ' C A L C U L A T I O N '//
     1 4X,'WEIGHT ....',1PE11.4,'.... WEIGHT IN LB..........',1PE11.4//
     3 4X,'XB0..........................................',1PE11.4//
     4 4X,'YB0..........................................',1PE11.4//
     5 4X,'ZB0..........................................',1PE11.4//
     6 4X,'TB0..........................................',1PE11.4//
     7 4X,'IUNIT........................................',I11//
     7 4X,'ISURF........................................',I11//)
  120 format(
     1 4X,'LB/MASS UNIT..................................',1PE11.4//
     2 4X,'FT/LENGTH UNIT................................',1PE11.4//
     3 4X,'MILLISEC/TIME UNIT............................',1PE11.4//
     4 4X,'PSI/PRESSURE UNIT.............................',1PE11.4//)
      END
```

```
      blockdata conwep_brl
      common /conwep_input/wtnt,x0,y0,z0,t0,isurf
      common /conwep_brlpso/ cpso(0:11,2)
      common /conwep_ambient/ c0, p0, g0, rho0
      common /conwep_units/ iunit,uft,ulb,upsi,ums
c
c Coefficients for incident pressure equations
c
      data((cpso(i,j),i=0,11),j=1,2)  /
     & 1.9422502013,   -1.6958988741,
     & -0.154159376846,  0.514060730593,
     & 0.0988534365274, -0.293912623038,
     & -0.0268112345019,  0.109097496421,
     & 0.00162846756311,-0.0214631030242,
     & 0.0001456723382,  0.00167847752266,
     & 1.77284970457,   -1.69012801396,
     & 0.00804973591951, 0.336743114941,
     & -0.00516226351334,-0.0809228619888,
     & -0.00478507266747, 0.00793030472242,
     & 0.0007684469735,  3*0.0 /

      data c0/1116./p0/14.696/g0/1.4/rho0/0.0765/
      end




      double precision function conwep_decay(p0,i0,td)
      real*4 p0,i0,td
      real*8 a,fa,fpa
c
c  find rate of decay for pressure assuming:
c
c    p(t) = p0 * [ 1 - (t-ta)/td] * exp[-a*(t-ta)/td]
c         (friedlander's equation)
c
c  where a is a decay coefficient.  integrating this equation
c  over time gives the impulse:
c
c    i0 = p0 * td * [ a + exp(-a) - 1 ] / a**2
c
c  find f(a) = a**2 - p0*td/i0*[ a + exp(-a) - 1] = 0
c  or:
c            a**2 / [ a + exp(-a) - 1] - p0*td/i0 = 0
```

```
c
c  for large a, exp(-a) approaches 0, and
c  a**2 / (a - 1) approaches a + 1
c
c    initial guess a = p0*td/i0 - 1
c
      ptoi = p0*td/i0
c   initial guess:
      a = ptoi - 1.
 1     fa  = a*a - ptoi*(a + exp(-a) - 1.)
      fpa = 2*a - ptoi*(1. - exp(-a))
      a = a - fa/fpa
      if(abs(fa) .gt. 1.e-6) go to 1
      conwep_decay = a
      return
      end



      subroutine conwep_params(ISURF,R,ta,to,pso,pro,WTNT,z,zlog,a,b)
c
c purpose: use the equations from brl technical report
c          arbrl-tr-02555 to find the incident pressure and time of
c          arrival for a point in space at a given slant range from
c          conventional explosion.
c
c  description of variables:
c    a         - decay coefficient of incident pressure
c                  vs. time history, where
c                   p(t) = pso*(1-t/to)*exp(-a*t/to)
c    b         - decay coefficient of reflected pressure
c                  vs. time history, where
c                   p(t) = pro*(1-t/to)*exp(-b*t/to)
c    pso      - peak incident overpressure, psi
c    pro      - peak reflected overpressure, psi
c    R         - slant range, ft
c    ISURF    - 1 = surface burst, 2 = air burst
c    ta        - arrival time, msec
c    to        - positive phase duration, msec
c    WTNT     - equivalent weight of tnt, lb
c    z         - scaled slant range, ft/lb**1/3
c    zlog     - logarithm of scaled slant range
c
      real*8 a, b, conwep_decay
      real  impo, impr
```

```
logical warn
save warn
data warn/.true./

w3 = WTNT**(1./3.)
z = R / w3
zlog = alog10(z)

if(ISURF.eq.1) then
  zlo = 0.45
else
  zlo = 0.37
endif

ta   = conwep_tarr(ISURF,ZLOG) * w3
to   = conwep_tdur(ISURF,ZLOG) * w3
impo = conwep_ximps(ISURF,ZLOG) * w3
impr = conwep_ximpr(isurf,zlog) * w3
pso  = conwep_pinc(ISURF,ZLOG)
pro  = conwep_pref(isurf,zlog)


if(z .ge. zlo) then
  if(pso*to/impo .le. 2.5 .or. pro*to/impr .le.2.5)then
    if(warn)then
      write(*,*) 'conwep_params: resetting to from ',to
      to = 2.5*max(impo/pso,impr/pro)
      write(*,*) '                              to ',to
      warn=.false.
    else
      to = 2.5*max(impo/pso,impr/pro)
    endif
  endif
  a = conwep_decay(pso,impo,to)
  b = conwep_decay(pro,impr,to)
else
  if(pso*to/impo .le. 2. .or. pro*to/impr .le.2.)then
    if(warn)then
      write(*,*) 'conwep_params: resetting to from ',to
      to = 2.0*max(impo/pso,impr/pro)
      write(*,*) '                              to ',to
      warn=.false.
    else
      to = 2.0*max(impo/pso,impr/pro)
    endif
```

```
      endif
      a = 0.
      b = 0.
    endif
c
    return
    end



    real function conwep_pinc(isurf,zlog)
c
c purpose: find the incident pressure due to the detonation
c       of a 1 lb equivalent tnt charge.  equations are from BRL
c       technical report ARBRL-TR-02555.  pressure is returned in
c       units of psi.
c
c  description of variables:
c   isurf   - 1 for surface burst, 2 for air burst
c   zlog    - logarithm (base 10) of scaled range
c
    common /conwep_brlpso/ cpso(0:11,2)
c
    u  = -0.756579301809 + 1.35034249993*zlog
    conwep_pinc = cpso(11,isurf)
    do 10 i = 10,0,-1
 10  conwep_pinc = conwep_pinc*u + cpso(i,isurf)
    conwep_pinc = 10.**conwep_pinc
    return
    end



    real function conwep_pref(isurf,zlog)
c
c purpose: find the normally reflected pressure due to the detonation
c       of a 1 lb equivalent tnt charge.  equations are from BRL
c       technical report ARBRL-TR-02555.  pressure is returned in
c       units of psi.
c
c  description of variables:
c   isurf   - 1 for surface burst, 2 for air burst
c   zlog    - logarithm (base 10) of scaled range
```

```
c
    parameter (ns=11, nf=9)
    real csurf(0:ns), cfree(0:nf)
    data csurf / 2.56431321138,  -2.21030870597,
   &          -0.218536586295,  0.895319589372,
   &           0.24989009775,  -0.569249436807,
   &          -0.11791682383,   0.224131161411,
   &           0.0245620259375, -0.0455116002694,
   &          -0.00190930738887, 0.00361471193389 /
    data cfree / 2.39106134946,  -2.21400538997,
   &           0.035119031446,  0.657599992109,
   &           0.0141818951887, -0.243076636231,
   &          -0.0158699803158,  0.0492741184234,
   &           0.00227639644004,-0.00397126276058 /
c
    if (isurf.eq.1) then
      u = -.789312405513 + 1.36637719229*zlog
      pref = csurf(ns)
      do 10 i = ns-1,0,-1
 10     pref = pref*u + csurf(i)
    else
      u = -0.756579301809 + 1.35034249993*zlog
      pref = cfree(nf)
      do 20 i = nf-1,0,-1
 20     pref = pref*u + cfree(i)
    endif
    conwep_pref = 10.**pref
    return
    end



    subroutine conwep_press(a,b,ts,p,ta,to,cosa,pro,pso)
    real*8 a,b
    common /conwep_units/ iunit,uft,ulb,upsi,ums
c
c  Compute pressure at shifted time ts for this field point in space.
c
c  The shifted time, ts (in milliseconds), is defined by
c
c                    ts = (t - t0)*ums
c
c  where t  = current DYNA3D elapsed time, seconds
c       t0 = delay time for the explosion, seconds
```

```
c          t0 > 0.0 means explosion takes place t0 seconds after
c                 DYNA3D calculation starts
c          t0 = 0.0 means explosion coincides with start of
c                 DYNA3D calculation
c          t0 < 0.0 means explosion occurs t0 seconds before
c                 DYNA3D calculation starts.
c
c
c  If the pressure front has not arrived here yet ( ts .LT. ta ),
c  and this will include the case when ts < 0.0 (i.e. (t - t0) < 0.0 ),
c  then the pressure at this field point is zero.
c
c  Parameters a, ta, to, and pmax are functions of the
c  charge parameters and the slant range distance from this field
c  point to the point of explosion.  These parameters must have been
c  correctly computed before entering this routine.
c
      if(ts.ge.ta) then

        exa = exp(-a*(ts-ta)/to)
        exb = exp(-b*(ts-ta)/to)
        poscosa=MAX(cosa,0.0)
        p =(pso*exa*(1.+poscosa-2.*poscosa**2)+pro*exb*poscosa**2)
     &              *(1-(ts-ta)/to)
        p = max(p,-14.7)

      else

        p  = 0.0

      endif
c
      return
      end



      SUBROUTINE conwep_stunit
      common /conwep_units/ iunit,uft,ulb,upsi,ums
      IF(IUNIT .EQ. 0. .or. iunit .eq. 1) THEN
c       Engineering: ft-lb-sec-psi
        UFT  = 1.
        ULB  = 1.
        UPSI = 1.
```

```
        ums  = 1000.
       ELSEIF (IUNIT .eq. 2) THEN
c         SI: m-kg-sec-Pascal
        UFT  = 1.0/0.3048
        ULB  = 1.0/0.45359
        UPSI = 1./6894.80
        ums  = 1000.
       ELSEIF (IUNIT .eq. 3) THEN
c         English: inch-12slugs-seconds-psi
        UFT  = 1./12.
        ULB  = 386.09
        UPSI = 1.
        ums  = 1000.
       ELSEIF (IUNIT .eq. 4) THEN
c         metric: cm-g-microsec-Mbar
        UFT  = 1.0/30.48
        ULB  = .001/0.45359
        UPSI = 1./(6894.8*1.e-11)
        ums  = .001
       ENDIF
       RETURN
       END




       real function conwep_tarr(ISURF,ZLOG)
c
c purpose: find the scaled time of arrival for the detonation of a
c        1 lb equivalent tnt charge.  equations are from BRL
c        technical report ARBRL-TR-02555.  arrival time is returned
c        in msec/lb**(1/3).
c
c  description of variables:
c   ISURF   - 1 for surface burst, 2 for air burst
c   ZLOG    - logarithm (base 10) of scaled range
c
       parameter (ns=9, nf=7)
       real csurf(0:ns), cfree(0:nf)
       data csurf /-0.173607601251,   1.35706496258,
      &           0.052492798645,  -0.196563954086,
      &          -0.0601770052288,  0.0696360270891,
      &           0.0215297490092, -0.0161658930785,
      &          -0.00232531970294, 0.00147752067524 /
       data cfree /-0.0423733936826, 1.36456871214,
```

```
     &         -0.0570035692784,-0.182832224796,
     &          0.0118851436014, 0.0432648687627,
     &         -0.0007997367834,-0.00436073555033 /
c
      if (ISURF.eq.1) then
        u  = -0.755684472698 + 1.37784223635*ZLOG
        conwep_tarr = csurf(ns)
        do 10 i = ns-1,0,-1
 10     conwep_tarr = conwep_tarr*u + csurf(i)
      else
        u = -0.80501734056 + 1.37407043777*ZLOG
        conwep_tarr = cfree(nf)
        do 20 i = nf-1,0,-1
 20     conwep_tarr = conwep_tarr*u + cfree(i)
      endif
      conwep_tarr = 10.**conwep_tarr
      return
      end



      real function conwep_tdur(ISURF,ZLOG)
c
c purpose: find the scaled duration for the detonation of a 1 lb
c       equivalent tnt charge. equations are from BRL technical
c       report ARBRL-TR-02555. duration is returned in
c       msec/lb**(1/3).
c
c  description of variables:
c  ISURF   - 1 for surface burst, 2 for air burst
c  ZLOG    - logarithm (base 10) of scaled range
c
      parameter (ns1=5,ns2=8,ns3=5,nf1=8,nf2=8,nf3=7)
      real csurf1(0:ns1), csurf2(0:ns2), csurf3(0:ns3)
      real cfree1(0:nf1), cfree2(0:nf2), cfree3(0:nf3)
      data csurf1 / -0.728671776005,   0.130143717675,
     &              0.134872511954,   0.0391574276906,
     &             -0.00475933664702,-0.00428144598008 /
      data csurf2 /  0.20096507334,   -0.0297944268976,
     &              0.030632954288,   0.0183405574086,
     &             -0.0173964666211, -0.00106321963633,
     &              0.00562060030977, 0.0001618217499,
     &             -0.0006860188944 /
      data csurf3 /  0.572462469964,   0.0933035304009,
```

```
     &           -0.0005849420883, -0.00226884995013,
     &           -0.00295908591505, 0.00148029868929 /
      data cfree1 / -0.801052722864,  0.164953518069,
     &            0.127788499497,  0.00291430135946,
     &            0.00187957449227, 0.0173413962543,
     &            0.00269739758043,-0.00361976502798,
     &           -0.00100926577934 /
      data cfree2 / 0.115874238335, -0.0297944268969,
     &            0.0306329542941, 0.018340557407,
     &           -0.0173964666286, -0.00106321963576,
     &            0.0056206003128, 0.0001618217499,
     &           -0.0006860188944 /
      data cfree3 / 0.50659210403,  0.0967031995552,
     &           -0.00801302059667, 0.00482705779732,
     &            0.00187587272287,-0.00246738509321,
     &           -0.000841116668,  0.0006193291052 /
c
c     surface burst
      if (ISURF.eq.1) then
        if(ZLOG .lt. -.34 ) then
          conwep_tdur = -.725
        elseif(ZLOG .lt. 0.4048337) then
          u  = -0.1790217052 + 5.25099193925*ZLOG
          conwep_tdur = csurf1(ns1)
          do 11 i = ns1-1,0,-1
 11         conwep_tdur = conwep_tdur*u + csurf1(i)
        elseif(ZLOG .lt. 0.845098) then
          u  = -5.85909812338 + 9.2996288611*ZLOG
          conwep_tdur = csurf2(ns2)
          do 12 i = ns2-1,0,-1
 12         conwep_tdur = conwep_tdur*u + csurf2(i)
        else
          u  = -4.92699491141 + 3.46349745571*ZLOG
          conwep_tdur = csurf3(ns3)
          do 13 i = ns3-1,0,-1
 13         conwep_tdur = conwep_tdur*u + csurf3(i)
        endif
      else
c     air burst
        if(ZLOG .lt. -.34 ) then
          conwep_tdur = -.824
        elseif(ZLOG .lt. 0.350248) then
          u = 0.209440059933 + 5.11588554305*ZLOG
          conwep_tdur = cfree1(nf1)
          do 21 i = nf1-1,0,-1
```

```
21      conwep_tdur = conwep_tdur*u + cfree1(i)
      elseif(ZLOG .lt. 0.7596678) then
        u = -5.06778493835 + 9.2996288611*ZLOG
        conwep_tdur = cfree2(nf2)
        do 22 i = nf2-1,0,-1
22      conwep_tdur = conwep_tdur*u + cfree2(i)
      else
        u = -4.39590184126 + 3.1524725264*ZLOG
        conwep_tdur = cfree3(nf3)
        do 23 i = nf3-1,0,-1
23      conwep_tdur = conwep_tdur*u + cfree3(i)
      endif

    endif
    conwep_tdur = 10.**conwep_tdur
    return
    end




    real function conwep_ximpr(isurf,zlog)
c
c purpose: find the scaled normally reflected impulse from the
c       detonation of a 1 lb equivalent tnt charge.  equations
c       are from BRL technical report ARBRL-TR-02555.  impulse
c       is returned in units of psi*msec/lb**(1/3)
c
c  description of variables:
c   isurf   - 1 for surface burst, 2 for air burst
c   zlog    - logarithm (base 10) of scaled range
c
    parameter (ns=3, nf=3)
    real csurf(0:ns), cfree(0:nf)
    data csurf / 1.75291677799,  -0.949516092853,
   &          0.112136118689, -0.0250659183287 /
    data cfree / 1.60579280091,  -0.903118886091,
   &          0.101771877942, -0.0242139751146 /
c
    if (isurf.eq.1) then
      u = -0.781951689212 + 1.33422049854*zlog
      ximpr = csurf(ns)
      do 10 i = ns-1,0,-1
10    ximpr = ximpr*u + csurf(i)
    else
```

```
      u = -0.757659920369 + 1.37882996018*zlog
      ximpr = cfree(nf)
      do 20 i = nf-1,0,-1
20    ximpr = ximpr*u + cfree(i)
    endif
    conwep_ximpr = 10.**ximpr
    return
    end
```

```
      real function conwep_ximps(isurf,zlog)
c
c purpose: find the scaled incident impulse for the detonation of
c         a 1 lb equivalent tnt charge.  equations are from brl
c         technical report arbrl-tr-02555.  impulse is returned in
c         units of psi*msec/lb**(1/3).
c
c  description of variables:
c   isurf    - 1 for surface burst, 2 for air burst
c   zlog     - logarithm (base 10) of scaled range
c
      parameter (ns1=4,ns2=7,nf1=4,nf2=8)
      real csurf1(0:ns1),csurf2(0:ns2),cfree1(0:nf1),cfree2(0:nf2)
      data csurf1 / 1.57159240621, -0.502992763686,
     &             0.171335645235, 0.0450176963051,
     &            -0.0118964626402 /
      data csurf2 / 0.719852655584,  -0.384519026965,
     &            -0.0280816706301,  0.00595798753822,
     &             0.014544526107,  -0.00663289334734,
     &            -0.00284189327204, 0.0013644816227 /
      data cfree1 / 1.43534136453, -0.443749377691,
     &             0.168825414684, 0.0348138030308,
     &            -0.010435192824 /
      data cfree2 / 0.599008468099,  -0.40463292088,
     &            -0.0142721946082,  0.00912366316617,
     &            -0.0006750681404, -0.00800863718901,
     &             0.00314819515931, 0.00152044783382,
     &            -0.0007470265899 /
c
      if (isurf.eq.1) then
       if(zlog .lt. 0.382017) then
         u = 0.832468843425 + 3.0760329666*zlog
         conwep_ximps = csurf1(ns1)
```

```
      do 11 i = ns1-1,0,-1
11      conwep_ximps = conwep_ximps*u + csurf1(i)
    else
      u = -2.91358616806 + 2.40697745406*zlog
      conwep_ximps = csurf2(ns2)
      do 12 i = ns2-1,0,-1
12      conwep_ximps = conwep_ximps*u + csurf2(i)
    endif
  else
    if(zlog .lt. 0.30103) then
      u = 1.04504877747 + 3.24299066475*zlog
      conwep_ximps = cfree1(nf1)
      do 21 i = nf1-1,0,-1
21      conwep_ximps = conwep_ximps*u + cfree1(i)
    else
      u = -2.67912519532 + 2.30629231803*zlog
      conwep_ximps = cfree2(nf2)
      do 22 i = nf2-1,0,-1
22      conwep_ximps = conwep_ximps*u + cfree2(i)
    endif
  endif
  conwep_ximps = 10.**conwep_ximps
  return
  end
```

APPENDIX B:

MODIFICATIONS TO THE DYNA2D CODE

INTENTIONALLY LEFT BLANK.

1. In subroutine dynai, change

```
    IF (IBRODE.NE.0) CALL BRODEI
```
to
```
    IF (IBRODE.EQ.1) CALL BRODEI
    IF (IBRODE.EQ.2) CALL conwep_BLASTI
```

2. In subroutine fe2a, change
```
    IF(LCC.GT.0) GO TO 15
    IF(LCC.EQ.0) THEN
        XCTR=.5*(R(IR)+R(IL))
        YCTR=0.
        ZCTR=.5*(Z(IR)+Z(IL))
        CALL SBRODE(XCTR,YCTR,ZCTR,TT,P,NPC,F)
        GO TO 30
    ENDIF
    ISHEAR=1
    LCC=-LCC
 15 IF(T(N).NE.0.0) GO TO 20
```
to
```
    IF(LCC.GT.0) GO TO 15
     if(ibrode.eq.0)then
      ISHEAR=1
      LCC=-LCC
     else
      IF(LCC.LE.0) THEN
       IF(LCC.EQ.-1) THEN
         XCTR=.5*(R(IR)+R(IL))
         YCTR=0.
         ZCTR=.5*(Z(IR)+Z(IL))
         CALL SBRODE(XCTR,YCTR,ZCTR,TT,P,NPC,F)
       ELSEIF (LCC.EQ.-2) THEN
        width=sqrt((r(ir)-r(il))**2 + (z(ir)-z(il))**2)
        call conwep_blast(  R(IR),  R(IL),  R(IL),  R(IR),
   &                        -width, -width,  width,  width,
   &                         Z(IR),  Z(IL),  Z(IL),  Z(IR),
   &                         PMULT(1,N),  TT, F)
      ENDIF
     endif
    GO TO 30
 15 IF(T(N).NE.0.0) GO TO 20
```

INTENTIONALLY LEFT BLANK.

APPENDIX C:

MODIFICATIONS TO THE DYNA3D CODE

INTENTIONALLY LEFT BLANK.

1. In subroutine dynai, change

```
IF (IBRODE.NE.0) CALL BRODEI
```

to

```
IF (IBRODE.EQ.1) CALL BRODEI
IF (IBRODE.EQ.2) CALL conwep_BLASTI
```

2. In subroutine load, change

```
      IF (LC(1,IP+I).LT.0) THEN
        CALL SBRODE(XCTR(I),YCTR(I),ZCTR(I),TT,P,NPC,F(I))
      ENDIF
```

to

```
      IF (LC(1,IP+I).EQ.-1) THEN
        CALL SBRODE(XCTR(I),YCTR(I),ZCTR(I),TT,P,NPC,F(I))
      ELSE IF (LC(1,IP+I).EQ.-2) THEN
        call conwep_blast( XX11(I),XX12(I),XX13(I),XX14(I),
     &                  XX21(I),XX22(I),XX23(I),XX24(I),
     &                  XX31(I),XX32(I),XX33(I),XX34(I),
     &                  pmult(1,ip+i),TT,F(I))

      ENDIF
```

INTENTIONALLY LEFT BLANK.

APPENDIX D:

MODIFICATIONS TO THE MAZE CODE

INTENTIONALLY LEFT BLANK.

1. Insert the following in block data:

```
COMMON /BRODCM/ brodcmv(33), ibrode
DATA ibrode/0/
```

2. Insert the following at the appropriate places in subroutine meshol2:

```
COMMON /BRODCM/ brodcmv(33), ibrode

IF (ICM1.EQ.'BROD'.or.icm1.eq.'BLAS') GO TO 815

815 call brodin
    go to 20
```

3. Add subroutine brodin:

```
      SUBROUTINE BRODIN
      COMMON /BRODCM/ YLD,H,XB0,YB0,ZB0,TB0,CL,CT,CP,IYOPT,RANGE(5),
     . COEFF(8),GFUNC(7),IUNIT,ISURF,CM,ibrode
      CHARACTER*4 ICMD
      CHARACTER*1 IPMT
      DATA IPMT/'.'/
      IBRODE=1
   10 CONTINUE
      CALL GETSYM(1,IPMT,ICMD,4)
      IF(ICMD.NE.'MODE')GO TO 15
      CALL GETNUM(1,IPMT,CRUD,IBRODE)
      GO TO 900
   15 CONTINUE
      IF(ICMD.NE.'YLD'.and.icmd.ne.'WEIG')GO TO 20
      CALL GETNUM(1,IPMT,YLD,ICRUD)
      GO TO 900
   20 CONTINUE
      IF(ICMD.NE.'HEIG')GO TO 30
      CALL GETNUM(1,IPMT,H,ICRUD)
      GO TO 900
   30 CONTINUE
      IF(ICMD.EQ.'XBO')ICMD='XB0'
      IF(ICMD.EQ.'YBO')ICMD='YB0'
      IF(ICMD.EQ.'ZBO')ICMD='ZB0'
      IF(ICMD.EQ.'TBO')ICMD='TB0'
      IF(ICMD.NE.'XB0')GO TO 40
```

```
      CALL GETNUM(1,IPMT,XB0,ICRUD)
      GO TO 900
   40 CONTINUE
      IF(ICMD.NE.'YB0')GO TO 45
      CALL GETNUM(1,IPMT,YB0,ICRUD)
      GO TO 900
   45 CONTINUE
      IF(ICMD.NE.'ZB0')GO TO 50
      CALL GETNUM(1,IPMT,ZB0,ICRUD)
      GO TO 900
   50 CONTINUE
      IF(ICMD.NE.'TB0')GO TO 60
      CALL GETNUM(1,IPMT,TB0,ICRUD)
      GO TO 900
   60 CONTINUE
      IF(ICMD.NE.'CM')GO TO 65
      CALL GETNUM(1,IPMT,CM,ICRUD)
      GO TO 900
   65 IF(ICMD.NE.'CL')GO TO 70
      CALL GETNUM(1,IPMT,CL,ICRUD)
      GO TO 900
   70 CONTINUE
      IF(ICMD.NE.'CT')GO TO 80
      CALL GETNUM(1,IPMT,CT,ICRUD)
      GO TO 900
   80 CONTINUE
      IF(ICMD.NE.'CP')GO TO 90
      CALL GETNUM(1,IPMT,CP,ICRUD)
      GO TO 900
   90 CONTINUE
      IF(ICMD.NE.'RANG')GO TO 110
      DO 100 I=1,5
      CALL GETNUM(1,IPMT,RANGE(I),ICRUD)
  100 CONTINUE
      IYOPT=1
      GO TO 900
  110 CONTINUE
      IF(ICMD.NE.'COEF')GO TO 130
      DO 120 I=1,8
      CALL GETNUM(1,IPMT,COEFF(I),ICRUD)
  120 CONTINUE
      IYOPT=1
      GO TO 900
  130 CONTINUE
      IF(ICMD.NE.'GFUN')GO TO 150
```

```
      DO 140 I=1,7
      CALL GETNUM(1,IPMT,GFUNC(I),ICRUD)
  140 CONTINUE
      IYOPT=1
      GO TO 900
  150 CONTINUE
      IF(ICMD.NE.'IUNI')GO TO 152
      CALL GETNUM(1,IPMT,CRUD,IUNIT)
      GO TO 900
  152 CONTINUE
      IF(ICMD.NE.'ISUR')GO TO 154
      CALL GETNUM(1,IPMT,CRUD,ISURF)
      GO TO 900
  158 IF(ICMD.NE.';')GO TO 160
      RETURN
  160 CONTINUE
      WRITE(*,2010)
  900 CONTINUE
      GO TO 10
 2010 FORMAT(' *** ERROR *** - ILLEGAL COMMAND IN BRODE FUNCTION INPUT')
      END
```

4. In subroutine meshol3, after

```
      CALL WRTIVL (A(N4+N2-N1),A(N4),NUMNPO)
```

add

```
      call ndbrod
```

5. Add subroutine ndbrod:

```
      SUBROUTINE NDBROD
      COMMON /BRODCM/ YLD,H,XB0,YB0,ZB0,TB0,CL,CT,CP,IYOPT,RANGE(5),
     . COEFF(8),GFUNC(7),IUNIT,ISURF,cm,ibrode

      if(ibrode.eq.1)then
       write(12,'("$ BRODE DATA")')
       WRITE(12,2010)YLD,H,XB0,YB0,ZB0,TB0
       WRITE(12,2020)CL,CT,CP,IYOPT
       IF(IYOPT.EQ.0)RETURN
        WRITE(12,2030)RANGE
        WRITE(12,2030)COEFF
        WRITE(12,2030)GFUNC

      elseif(ibrode.eq.2)then
       write(12,'("$ CONWEP BLAST DATA")')
```

```
    WRITE(12,2040)YLD,XB0,YB0,ZB0,TB0,IUNIT,ISURF
    if(iunit.eq.5)then
     WRITE(12,2040)cm,cl,ct,cp
    endif

   endif

   RETURN
2010 FORMAT(6E10.3)
2020 FORMAT(3E10.3,I5)
2030 FORMAT(8E10.3)
2040 FORMAT(5E10.3,2I5)
   END
```

APPENDIX E:

MODIFICATIONS TO THE INGRID CODE

INTENTIONALLY LEFT BLANK.

1. In subroutine dnopts, change

```
IF(IDATA.NE.'BROD')GO TO 10
```

to

```
IF(IDATA.NE.'BROD' .and. idata.ne.'BLAS')GO TO 10
```

2. Replace subroutine brodin with

```
      SUBROUTINE BRODIN
      character *4 ianal,ibwm,iteo,nsmd
      common /junk00/ ianal,ibwm,iteo,nsmd
      COMMON /JUNK01/ GX,GY,GZ,NSTEP,DELT,TERM,DCTOL,ECTOL,
     . SHIFT,MAXMEM,NMODES,IPRT,IPLT,ANIP1,ANIP2,NLOADC,RX,RY,RZ,
     . NBSR,NBEI,NIBSR,MSRF,ISBRF,PRTI,PLTI,IBRODE,NDETP,
     . TSSF,NMOMDP,NVELBC,AITSS,IRIGID,NUMNPB,NUMEPB,NUM1PB,NUM2PB,
     . NUM3PB,NUM4PB,ISSTRN,IHYDRO,ISHLNRM,ISHLTCK,ISHLFRM
      COMMON /BRODCM/ YLD,H,XB0,YB0,ZB0,TB0,CL,CT,CP,IYOPT,RANGE(5),
     . COEFF(8),GFUNC(7),IUNIT,ISURF,CM,IJUNK
      CHARACTER*4 ICMD
      CHARACTER*1 IPMT
      DATA IPMT/'.'/
      IBRODE=1
   10 CONTINUE
      CALL GETSYM(1,IPMT,ICMD)
      CALL SETOPT(ICMD)
      IF(ICMD.NE.'MODE')GO TO 15
      CALL GETNUM(1,IPMT,CRUD,IBRODE)
      GO TO 900
   15 CONTINUE
      IF(ICMD.NE.'YLD'.and.icmd.ne.'WEIG')GO TO 20
      CALL GETNUM(1,IPMT,YLD,ICRUD)
      GO TO 900
   20 CONTINUE
      IF(ICMD.NE.'HEIG')GO TO 30
      CALL GETNUM(1,IPMT,H,ICRUD)
      GO TO 900
   30 CONTINUE
      IF(ICMD.EQ.'XBO')ICMD='XB0'
      IF(ICMD.EQ.'YBO')ICMD='YB0'
      IF(ICMD.EQ.'ZBO')ICMD='ZB0'
      IF(ICMD.EQ.'TBO')ICMD='TB0'
      IF(ICMD.NE.'XB0')GO TO 40
      CALL GETNUM(1,IPMT,XB0,ICRUD)
      GO TO 900
   40 CONTINUE
```

```
      IF(ICMD.NE.'YB0')GO TO 45
      CALL GETNUM(1,IPMT,YB0,ICRUD)
      GO TO 900
   45 CONTINUE
      IF(ICMD.NE.'ZB0')GO TO 50
      CALL GETNUM(1,IPMT,ZB0,ICRUD)
      GO TO 900
   50 CONTINUE
      IF(ICMD.NE.'TB0')GO TO 60
      CALL GETNUM(1,IPMT,TB0,ICRUD)
      GO TO 900
   60 CONTINUE
      IF(ICMD.NE.'CM')GO TO 65
      CALL GETNUM(1,IPMT,CM,ICRUD)
      GO TO 900
   65 IF(ICMD.NE.'CL')GO TO 70
      CALL GETNUM(1,IPMT,CL,ICRUD)
      GO TO 900
   70 CONTINUE
      IF(ICMD.NE.'CT')GO TO 80
      CALL GETNUM(1,IPMT,CT,ICRUD)
      GO TO 900
   80 CONTINUE
      IF(ICMD.NE.'CP')GO TO 90
      CALL GETNUM(1,IPMT,CP,ICRUD)
      GO TO 900
   90 CONTINUE
      IF(ICMD.NE.'RANG')GO TO 110
      DO 100 I=1,5
      CALL GETNUM(1,IPMT,RANGE(I),ICRUD)
  100 CONTINUE
      IYOPT=1
      GO TO 900
  110 CONTINUE
      IF(ICMD.NE.'COEF')GO TO 130
      DO 120 I=1,8
      CALL GETNUM(1,IPMT,COEFF(I),ICRUD)
  120 CONTINUE
      IYOPT=1
      GO TO 900
  130 CONTINUE
      IF(ICMD.NE.'GFUN')GO TO 150
      DO 140 I=1,7
      CALL GETNUM(1,IPMT,GFUNC(I),ICRUD)
  140 CONTINUE
```

```
      IYOPT=1
      GO TO 900
  150 CONTINUE
      IF(ICMD.NE.'IUNI')GO TO 152
      CALL GETNUM(1,IPMT,CRUD,IUNIT)
      GO TO 900
  152 CONTINUE
      IF(ICMD.NE.'ISUR')GO TO 154
      CALL GETNUM(1,IPMT,CRUD,ISURF)
      GO TO 900
      GO TO 900
  158 IF(ICMD.NE.';')GO TO 160
      CALL POPMG
      RETURN
  160 CONTINUE
      go to 900
  180 continue
      WRITE(*,2010)
      WRITE(3,2010)
      CALL TRACE(1)
  900 CONTINUE
      CALL POPMG
      GO TO 10
 2010 FORMAT(' *** ERROR *** - ILLEGAL COMMAND IN BRODE FUNCTION INPUT')
      END
```

3. Replace subroutine ndbrod with

```
      SUBROUTINE NDBROD(IBRODE)
      COMMON /BRODCM/ YLD,H,XB0,YB0,ZB0,TB0,CL,CT,CP,IYOPT,RANGE(5),
     . COEFF(8),GFUNC(7),IUNIT,ISURF,cm

      if(ibrode.eq.1)then
      write(13,'("$ BRODE DATA")')
        WRITE(13,2010)YLD,H,XB0,YB0,ZB0,TB0
        WRITE(13,2020)CL,CT,CP,IYOPT
        IF(IYOPT.EQ.0)RETURN
          WRITE(13,2030)RANGE
          WRITE(13,2030)COEFF
          WRITE(13,2030)GFUNC

      elseif(ibrode.eq.2)then
      write(13,'("$ BLAST DATA")')
        WRITE(13,2040)YLD,XB0,YB0,ZB0,TB0,IUNIT,ISURF
        if(iunit.eq.5)then
```

```
      WRITE(13,2040)cm,cl,ct,cp
     endif

    endif

    RETURN
2010 FORMAT(6E10.3)
2020 FORMAT(3E10.3,I5)
2030 FORMAT(8E10.3)
2040 FORMAT(5E10.3,3I5)
    END
```

APPENDIX F:

CHANGES TO THE DYNA2D MANUAL

INTENTIONALLY LEFT BLANK.

The following changes should be made to the DYNA2D manual, April 1992, when the ConWep blast model is installed.

Page 41, Card 7, field 6-10 (IBRODE)

add

EQ.2: ConWep blast model parameters are defined.

Page 157, Pressure and Shear Loads

change

LT.0: Brode function is used to determine pressure time history

to

-1: Brode function is used to determine pressure time history
-2: ConWep blast model is used to determine pressure time history

Page 178, Brode Functions

change

if columns 21-25 are blank

to

if columns 21-25 (IBRODE) are blank

change

Otherwise, enter two cards

to

Otherwise, if IBRODE EQ 1 enter two cards add pages 178.1 and 178.2:

If IBRODE EQ 2, enter one or two cards for the pertinent ConWep blast function data.

```
+--------+
| Card 1 |
+--------+
```

Columns	Quantity	Format
1-10	Equivalent weight of TNT	E10.0
11-20	X0 (x-coordinate of blast origin)	E10.0
21-30	Y0 (x-coordinate of blast origin)	E10.0
31-40	Z0 (x-coordinate of blast origin)	E10.0
51-60	T0 (time of explosion)	E10.0
61-65	IUNIT (problem units)	I5

EQ.0: iunit=2
EQ.1: feet, pounds, seconds, psi
EQ.2: meters, kilograms, seconds, Pascals
EQ.3: inch, dozens of slugs, seconds, psi
EQ.4: centimeters, grams, microseconds, Megabars
EQ.5: user conversions will be read from Card 2

Columns	Quantity	Format
66-70	ISURF (surface burst or air burst flag)	I5

EQ.0: isurf=2
EQ.1: surface burst
EQ.2: air burst

If IUNIT is not 5, skip the rest of this section. Otherwise, include Card 2:

```
+--------+
| Card 2 |
+--------+
```

Columns	Quantity	Format
1-10	ULB pounds per problem length unit	E10.0
11-20	UFT feet per problem length unit	E10.0
21-30	UMS milliseconds per problem length unit	E10.0
31-40	UPSI psi per problem pressure unit	E10.0

The ConWep functions are internally written in terms of feet, milliseconds, pounds, and psi. IUNIT supplies the conversion factors to DYNA2D units.

The development and limitations underlying this option are given in (Randers-Pehrson and Bannister, 1997), (Kingery and Bulmash, 1984) and (Hyde, 1993)

Page 241, References

add

30.1 Hyde, David W., "User's Guide for Microcomputer Program ConWep, Applications of TM 5-855-1, 'Fundamentals of Protective Design for Conventional Weapons'", U.S. Army Corps of Engineers Waterways Experiment Station Instruction Report SL-88-1, Vicksburg, MS, April 1988, revised February 1993.

and

38.1 Kingery, Charles N., and Gerald Bulmash, "Airblast Parameters from TNT Spherical Air Burst and Hemispherical Surface Burst," U.S. Army Ballistic Research Laboratory Technical Report ARBRL-TR-02555, Aberdeen Proving Ground, MD, April 1984.

Page 242, References

add

47.1 Randers-Pehrson, Glenn, and Kenneth A. Bannister, "Airblast Loading Model for DYNA2D and DYNA3D," U.S. Army Research Laboratory Technical Report ARL-TR-1310, Aberdeen Proving Ground, MD, March 1997.

APPENIX G:

CHANGES TO THE DYNA3D MANUAL

INTENTIONALLY LEFT BLANK.

The following changes should be made to the DYNA3D manual, November 1993, when the ConWep blast model is installed.

Page 44, Card 4, field 21-25 (IBRODE)

add

EQ.2: ConWep blast model parameters are defined.

Page 249, Pressure Loads

change

LT.O: Brode function is used to determine pressure

to

EQ.-1: Brode function is used to determine pressure
EQ.-2: ConWep blast model is used to determine pressure time history

Page 241, Brode Functions

change

if columns 21-25 are blank

to

if columns 21-25 (IBRODE) are blank

change

Otherwise, enter two cards

to

Otherwise, if IBRODE EQ 1 enter two cards add pages 241.1 and 241.2:

If IBRODE EQ 2, enter one or two cards for the pertinent ConWep blast function data.

```
+--------+
| Card 1 |
+--------+
```

Columns	Quantity	Format
1-10	Equivalent weight of TNT	E10.0
11-20	X0 (x-coordinate of blast origin)	E10.0
21-30	Y0 (x-coordinate of blast origin)	E10.0
31-40	Z0 (x-coordinate of blast origin)	E10.0
51-60	T0 (time of explosion)	E10.0
61-65	IUNIT (problem units)	I5

EQ.0: iunit=2
EQ.1: feet, pounds, seconds, psi
EQ.2: meters, kilograms, seconds, Pascals
EQ.3: inch, dozens of slugs, seconds, psi
EQ.4: centimeters, grams, microseconds, Megabars
EQ.5: user conversions will be read from Card 2

Columns	Quantity	Format
66-70	ISURF (surface burst or air burst flag)	I5

EQ.0: isurf=2
EQ.1: surface burst
EQ.2: air burst

If IUNIT is not 5, skip the rest of this section. Otherwise, include Card 2:

```
+--------+
| Card 2 |
+--------+
```

Columns	Quantity	Format
1-10	ULB pounds per problem length unit	E10.0
11-20	UFT feet per problem length unit	E10.0
21-30	UMS milliseconds per problem length unit	E10.0
31-40	UPSI psi per problem pressure unit	E10.0

The ConWep functions are internally written in terms of feet, milliseconds, pounds, and psi. IUNIT supplies the conversion factors to DYNA3D units.

The development and limitations underlying this option are given in (Randers-Pehrson and Bannister, 1997), (Kingery and Bulmash, 1984) and (Hyde, 1993)

Page 351, References

add

44.1 Hyde, David W., "User's Guide for Microcomputer Program ConWep, Applications of TM 5-855-1, 'Fundamentals of Protective Design for Conventional Weapons'", U.S. Army Corps of Engineers Waterways Experiment Station Instruction Report SL-88-1, Vicksburg, MS, April 1988, revised February 1993.

Page 352, References

add

51.1 Kingery, Charles N., and Gerald Bulmash, "Airblast Parameters from TNT Spherical Air Burst and Hemispherical Surface Burst," U.S. Army Ballistic Research Laboratory Technical Report ARBRL-TR-02555, Aberdeen Proving Ground, MD, April 1984.

Page 353, References

add

60.1 Randers-Pehrson, Glenn, and Kenneth A. Bannister, "Airblast Loading Model for DYNA2D and DYNA3D," U.S. Army Research Laboratory Technical Report ARL-TR-1310, Aberdeen Proving Ground, MD, March 1997.

APPENDIX H:

CHANGES TO THE MAZE MANUAL

INTENTIONALLY LEFT BLANK.

The following changes should be made to the MAZE manual, June 1983, when the ConWep blast model is installed.

Replace page 36 (This page intentionally left blank) with

BLAST options ; (DYNA2D only) Activate ConWep model

WEIGHT	weight	Equivalent weight of TNT
XB0	x	X-coordinate of point of explosion
YB0	y	Y-coordinate of point of explosion
ZB0	z	Z-coordinate of point of explosion
TB0	time	Time-zero of explosion
ISURF	isurf	type of burst EQ.1: surface burst EQ.2: air burst (default)
IUNIT	iunit	Units flag EQ.1: feet, pounds, seconds, psi EQ.2: meters, kilograms, seconds, Pascals (default) EQ.3: inch, dozens of slugs, seconds, psi EQ.4: centimeters, grams, microseconds, Megabars EQ.5: user conversions will be supplied
CM	ulb	Conversion factor, pounds per DYNA2D mass unit
CL	uft	Conversion factor, feet per DYNA2D length unit
CT	ums	Conversion factor, milliseconds per DYNA2D time unit
CP	upsi	Conversion factor, psi per DYNA2D pressure unit
;		end of BLAST option list

Page 27:

PBC

add

If k is greater than 1,000,000, then k=1,000,000-k. For example, to apply a ConWep blast load, use k=1,000,002; -2 will be written in the DYNA2D input file. Similarly, use k=1,000,001 to apply a Brode pressure load.

PBCS

add

If k is greater than 1,000,000, then k=1,000,000-k (see PBC, above).

INTENTIONALLY LEFT BLANK.

APPENDIX I:

CHANGES TO THE INGRID MANUAL

INTENTIONALLY LEFT BLANK.

The following changes should be made to the draft INGRID manual, September 1992, when the ConWep blast model is installed.

insert on page page 50

BLAST options ; Activate ConWep model with the following optional parameters. ConWep pressure loads are applied to surfaces using load curve number -2, the values for which are generated by the ConWep model.

WEIGHT weight Equivalent weight of TNT
XB0 x X-coordinate of point of explosion
YB0 y Y-coordinate of point of explosion
ZB0 z Z-coordinate of point of explosion
TB0 time Time-zero of explosion
ISURF isurf type of burst
EQ.1: surface burst
EQ.2: air burst (default)
IUNIT iunit Units flag
EQ.1: feet, pounds, seconds, psi
EQ.2: meters, kilograms, seconds, Pascals (default)
EQ.3: inch, dozens of slugs, seconds, psi
EQ.4: centimeters, grams, microseconds, Megabars
EQ.5: user conversions will be supplied
CM ulb Conversion factor, pounds per DYNA2D mass unit
CL uft Conversion factor, feet per DYNA2D length unit
CT ums Conversion factor, milliseconds per DYNA2D time unit
CP upsi Conversion factor, psi per DYNA2D pressure unit

INTENTIONALLY LEFT BLANK.

APPENDIX J:

SAMPLE INGRID INPUT

INTENTIONALLY LEFT BLANK.

```
c  Surface blast 2.32kg TNT y=45.72 cm
DN3D.
TERM 0.001_001 PRTI 0.000_005 PLTI 0.000_005
itss 0.000_000_005 tssf .000_010


blast model 2 weight 2.32448 xb0 0 yb0 .4572 zb0 0
     tb0 -.000_100 iunits 2 isurf 1 ;

C  IMPULSE GAGE MATERIAL, SI units
MAT 1 1 RO 20,000. E 0 PR 0 ENDMAT
c  if the impulse gage is .1m thick, rho=2*10**4.
c  mass/area = 1,000
c  pressure = acceleration * 1,000
c  impulse  = velocity * 1,000

mate 1
stab     120;     1;       2;   c impulse gage (10 cm strip)
    -.1 1.1  -.10 0   -.02 .02
   pr 0 2 0 0 2 0  -2  1  0 -1 0
    b 0 0 0 0 0 0  101000
end

stab 12; 12; 12                  c this part just for illustration
    -.1.1 1.1 -1.1 1.1 -.02 .02    c 2.2 m x 2.2 m plate
end

;
```

INTENTIONALLY LEFT BLANK.

APPENDIX K:

SAMPLE MAZE INPUT

INTENTIONALLY LEFT BLANK.

```
noplot tv 1625
c
ld 1 lp 2 0 -.1 0 .15  c axis
ld 4 lp 2 0 0 1.0 0  c z=0
ld 6 lp 2 0 -.02 1.0 -.02 c table bottom
ld 7 lp 2 0.01 -.1 0.01 0 c table edge
ld 8 lp 2 0 -.04 1.0 -.04 c table bottom
ld 9 lp 2 0 -.06 1.0 -.06 c table bottom
ld 10 lp 2 0 -.08 1.0 -.08 c table bottom
ld 11 lp 2 0 -.10 1.0 -.10 c table bottom

part 1 6 7 4 1    1  5  yes   c table/impulse gage
part 1 11 7 10 2  1  10 yes c extra nodes to get pbc/node ratio down to 1/5

c  if the impulse gage is .02m thick, rho=10**6.
c  mass/area = 10,000
c  pressure = acceleration * 10,000
c  impulse  = velocity * 10,000

assm

title
CONWEP BLAST MODEL, HUFFINGTON AND EWING TEST T89 W=896 Z=.06322

TERM .000_300 PRTI .000_0025 PLTI .000_0025

rcon 0

c put pressure boundary condition on top of table
p 1 b pbcs 3 1000002 1 1

c sphere weight 896 grams pentolite * .81/.7 === 1.036 kilograms TNT
  blast model 2 weight 1.036 xb0 0 yb0 0
  zb0  .06322 tb0 -.000_006 iunits 2 isurf 2 ;

c enforce reasonable timestep
itss 0.000_000_005 tssf .000_001

wbcd dyna2d

C  IMPULSE GAGE MATERIAL (TABLE)
MAT 1 1 RO 1,000,000. E 0 PR 0 ENDMAT

MAT 2 1 RO 1. E 0 PR 0 ENDMAT
t
```

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DFFNS TECHNICAL INFO CTR ATTN DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDQ ATTN DENNIS SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE DEPT OF MATHEMATICAL SCI ATTN MDN A MAJ DON ENGEN THAYER HALL WEST POINT NY 10996-1786
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CS AL TP 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CS AL TA 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1145
	ABERDEEN PROVING GROUND
2	DIR USARL ATTN AMSRL CI LP (305)

NO. OF COPIES	ORGANIZATION
6	CDR US ARMY ARDEC ATTN AMSTA AR AEE WW E BAKER C CHIN R FONG J PEARSON J WALSH TECH LIB PICATINNY ARSNL NJ 07806-5000
1	CDR US ARMY ARDEC ATTN AMSTA AR AET M TECH LIB PICATINNY ARSNL NJ 07806-5000
1	CDR US ARMY ARDEC ATTN AMSTA AR FS E ANDRICOPOULOS PICATINNY ARSNL NJ 07806-5000
3	CDR US ARMY MERDEC ATTN AMSME RD ST WF L CRAFT D LOVELACE M SCHEXNAYDER REDSTONE ARSNL AL 35898-5250
1	DIR US ARO WASH ATTN AMXRO W K A BANNISTER RM 8N31 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
3	DIR US ARO ATTN J CHANDRA K IYER TECH LIB PO BOX 12211 RSCH TRIANGLE PK NC 27709-2211
4	CDR US ARMY COE ATTN J BALSARA T BLEVINS P PAPIDOS R NAMBURU 3909 HALLS FERRY RD VICKSBURG MS 39180-6199

NO. OF COPIES	ORGANIZATION
3	DIR US ARMY DARPA ATTN J RICHARDSON TECH INFO B WILCOX 3701 N FAIRFAX DR ARLINGTON VA 22203-1714
2	CDR US ARMY TACOM ATTN AMSTA RSK J THOMPSON S GOODMAN WARREN MI 48397-5000
2	DIR NRL ATTN J A NEMES A E WILLIAMS CODE 6684 4555 OVERLOOK AVE SW WASH DC 20375
2	CDR NSWC ATTN W H HOLT CODE G22 W MOCK 17320 DAHLGREN RD DAHLGREN VA 22448-5000
8	CDR NSWC ATTN C S COFFEY R K GARRETT JR H MAIR R12 J MCKIRGAN B PARK D G TASKER TECH LIB F ZERILLI 10901 NEW HAMPSHIRE AVE SILVER SPRING MD 20903-5000
2	CDR NWC ATTN T GILL CODE 3261 TECH LIB CHINA LAKE CA 93555-6001
1	NAVAL POST GRAD SCHL ATTN J STERNBERG CODE 73 MONTEREY CA 93943
1	USAF PHILLIPS LAB ATTN PL WSCD F ALLAHDADI KIRTLAND AFB NM 87185
1	USAF WAL ATTN T NICHOLAS WRIGHT PAT AFB OH 45433

NO. OF COPIES	ORGANIZATION
4	USAF WL ATTN MNMW W COOK J FOSTER M NIXON TECH LIB EGLIN AFB FL 32542-5434
13	DIR LANL ATTN T F ADAMS F663 J BOLSTAD G787 J CHAPYAK R DAVIDSON K557 E FERM P FOLLANSBEE F663 G T GRAY III B 295 K HOLIAN B295 L HULL J JOHNSON F663 D A MANDELL F663 P MAUDLIN TECH LIB PO BOX 1663 LOS ALAMOS NM 87454
12	DIR LLNL ATTN B R BOWMAN L122 R COUCH L35 M FINGER L38 W H GOURDIN G GOUDREAU C HOOVER D LASSILA L342 P RABOIN J E REAUGH L290 M J MURPHY TECH LIB R E TIPTON L35 PO BOX 1663 LOS ALAMOS NM 87545
2	DIR SNL ATTN D BAMMANN M CHIESA LIVERMORE CA 94550

NO. OF COPIES	ORGANIZATION
9	DIR SNL ATTN R M BRANNON DIV 1432 L C CHHABILDAS MS 0821 M FORRESTAL DIV 1551 E S HERTEL JR MS 0819 M KIPP DIV 1533 J M MCGLAUN MS 0819 S A SILLING T TRUCANO MS 0819 P YARRINGTON DIV 1533 PO BOX 5800 ALBUQUERQUE NM 87185-0307
4	IAT UNIV OF TX AUSTIN ATTN S J BLESS H D FAIR T M KIEHNE M J NORMANDIA 4030 2 W BRAKER LN AUSTIN TX 78759-5329
4	SOUTHWEST RSCH INST DEPT OF MECH SCI ATTN C ANDERSON D LITTLEFIELD S MULLIN J WALKER 8500 CULEBRA RD PO DRAWER 28510 SAN ANTONIO TX 78284
1	AEROJET ELECTRO SYS CO ATTN WARHEAD SYSTEMS J CARLEONE PO BOX 296 AZUSA CA 91702
3	ALLIANT TECHSYS INC ATTN S BEISSEL T HOLMQUIST MN11 2720 R STRYK G R JOHNSON MN11 2925 7225 NORTHLAND DR BROOKLYN PK MN 55428
1	ALME AND ASSOC ATTN M L ALME 102 STEVENS FOREST PROF 9650 SANTIAGO RD COLUMBIA MD 21045

NO. OF COPIES	ORGANIZATION
1	APPLIED RSCH ASSOC INC ATTN J D YATTEAU 5941 S MIDDLEFIELD RD STE 100 LITTLETON CO 80123
2	APPLIED RSCH ASSOC INC ATTN T C CARNEY F MAESTAS 4300 SAN MATEO BLVD SE STE A220 ALBUQUERQUE NM 87110
1	BRIGS CO ATTN J E BACKOFEN 2668 PETERSBOROUGH ST HERNDON VA 22071-2443
3	DYNA EAST CORP ATTN P C CHOU R CICCARELLI W FLIS 3620 HORIZON DRIVE KING OF PRUSSIA PA 19406
2	GEN RSCH CORP ATTN A CHARTERS T MENNA 5383 HOLLISTER AVE SANTA BARBARA CA 93111
1	IRA INC ATTN D ORPHAL 4450 BLACK AVE STE E PLEASANTON CA 94566
1	KAMAN SCIENCES CORP ATTN J S WILBECK 7600 BLVD S STE 208 HUNTSVILLE AL 35802
1	KAMAN SCIENCES CORP ATTN N ARI PO BOX 7463 COLORADO SPRINGS CO 80933-7463
1	D R KENNEDY & ASSOC INC ATTN D KENNEDY PO BOX 4003 MOUNTAIN VIEW CA 94040

NO. OF COPIES	ORGANIZATION
1	KERLEY PUB SVC ATTN G I KERLEY PO BOX 13835 ALBUQUERQUE NM 87192-3835
3	LIVERMORE SOFTWARE TECH CORP ATTN J O HALLQUIST B MAKER D STILLMAN 2876 WAVERLY WAY LIVERMORE CA 94550
1	ORLANDO TECH INC ATTN D A MATUSKA PO BOX 855 SHALIMAR FL 32579
2	SRI INTERNATIONAL ATTN D CURRAN L SEAMAN 333 RAVENSWOOD AVE MENLO PARK CA 94025
1	ZERNOW TECH SVC INC ATTN L ZERNOW 425 W BONITA AVE STE 208 SAN DIMAS CA 91773
	ABERDEEN PROVING GROUND
1	DIR AMSAA ATTN R THOMPSON
49	DIR USARL ATTN AMSRL SL B, P DEITZ (328) AMSRL SL BC, J T KLOPCIC (328) AMSRL SL BV, R SAUCIER (247) R SHNIDMAN (247) J R STROBEL (247) AMSRL WM, D ECCLESHALL AMSRL WM MA S, J BEATTY S CHOU J DANDEKAR D J GROVE A RAJENDRAN T WEERASOORIYA

NO. OF
COPIES ORGANIZATION

AMSRL WM PD,
G GAZONAS
D HOPKINS
S WILKERSON
AMSRL WM T, W MORRISON
AMSRL WM TA,
W BRUCHEY JR
G FILBEY JR
W GILLICH
W GOOCH JR
Y HUANG
H MEYER JR
E RAPACKI JR
AMSRL WM TB,
R FREY
J STARKENBERG
AMSRL WM TC,
T BJERKE
R COATES
W S DE ROSSET
F GRACE
K KIMSEY
M LAMPSON
L MAGNESS
D SCHEFFLER
G SILSBY
W WALTERS
AMSRL WM TD,
R L BITTING
A M DIETRICH JR
T FARRAND
K FRANK
N GNIAZDOWSKI
F GREGORY
P W KINGMAN
M RAFTENBERG
G RANDERS-PEHRSON
M SCHEIDLER
S SCHOENFELD
S SEGLETES
J WALTER JR
T W WRIGHT

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1997	Final, February 1994 - September 1994

4. TITLE AND SUBTITLE

Airblast Loading Model for DYNA2D and DYNA3D

5. FUNDING NUMBERS

PR: 1L162618AH80

6. AUTHOR(S)

Glenn Randers-Pehrson and Kenneth A. Bannister

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-WM-TD
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-1310

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

We incorporated the CONWEP blast model into DYNA2D and DYNA3D. It works as expected and appears to be adequate for modeling problems, such as vehicle response to land mines. The model accounts for the angle of incidence of the blast wave, but it does not account for shadowing by intervening objects or for confinement effects. This report provides FORTRAN listings and directions for incorporating the model in DYNA2D, DYNA3D, and associated preprocessors and postprocessors and suggests changes to the user manuals.

14. SUBJECT TERMS

blast, DYNA2D, DYNA3D, CONWEP

15. NUMBER OF PAGES

73

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1310 (Randers-Pehrson) Date of Report March 1997

2. Date Report Received ____________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ____________________

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ____________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ____________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ____________________

CURRENT ADDRESS

____
Organization

____
Name E-mail Name

____
Street or P.O. Box No.

____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

____
Organization

____
Name

____
Street or P.O. Box No.

____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
US ARMY RESEARCH LABORATORY
ATTN AMSRL WM TD
ABERDEEN PROVING GROUND MD 21005-5066