Fermi National
Accelerator Laboratory

Summer Student Program

Final report

Simulation of material deformation due to multiple beam pulses on beryllium target

Author:
Matteo Malacarne

Supervisor:
Sujit Bidhar

October 2018

Abstract

The purpose of my Summer Student program at Fermilab has been to predict of the behaviour of target material subjected to multiple beam pulses using FEM softwares and compare these results with the experimental data obtained from the HiRadMat tests done at CERN.

In the first part of the program I continued the development of the model that had been created to simulate a single beam pulse and then, in the second part I implemented a user-material subroutine to incorporate Equation of State (EOS) to consider pressure and temperature dependent material properties.

Contents

Chapter 1

Introduction

1.1 HiRadMat facility at CERN

The High Radiation to Materials (HiRadMat) is a new test area designed to investigate the behavior of materials, in particular accelerator components of the Large Hadron Collider (LHC), when irradiated with high-energy and high-intensity pulsed beams extracted from the CERN Super Proton Synchrotron (SPS).

Beyond that, experiments in HiRadMat can contribute to the understanding of the behavior of materials under impact on high-energy protons, thus providing important feedback for the design of the new high-power accelerators and spallation target stations considered these days.

Experiments in HiRadMat under controlled beam loading conditions and with specially designed samples would provide data to check mechanical and multi-physics simulation codes and material strength model, thus enhancing their predictive power for future applications .

Table 1.1: Energy parameters of HiRadMat proton beams

Parameter	**Value**
Beam energy	440 GeV
Max. bunch intensity	1.2×10^{-11}
No. of bunches	$1 - 288$
Max. pulse intensity	3.5×10^{13} ppp
Max. pulse length	$7.2\,\mu$s
Gaussian beam size	1σ: $0.1 - 2$ mm

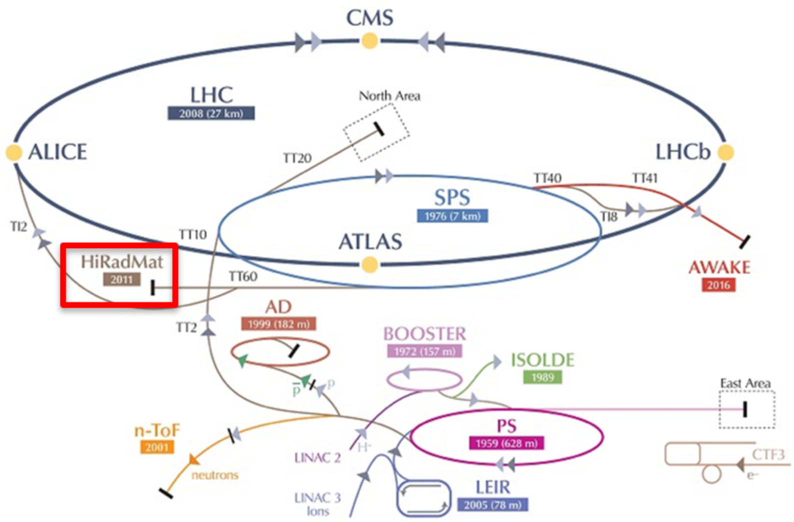


Figure 1.1: HiRadMat facility inside CERN accelerator complex.

1.2 Past studies

In the past, several simulation had been performed to predict the material response when subjected to single pulse beam and the experimental results was well reproduced by the simulations. Figure 1.2 represents a comparison between experimental data and simulation obtained for a circular beryllium target (⌀15 mm, 0.75 mm thickness): residual plastic deformation along disc axis in shown on the y axis while distance from that axis is shown on the x axis. Out of plane deformation is evaluated using profilometry.

A problem was encountered when multiple pulses would be simulated: the simulation worked well until the first loading but then failed to catch the next loadings. During the first part of my internship I worked to solve these problems regarding simulation settings in order to perform a six pulses simulation.

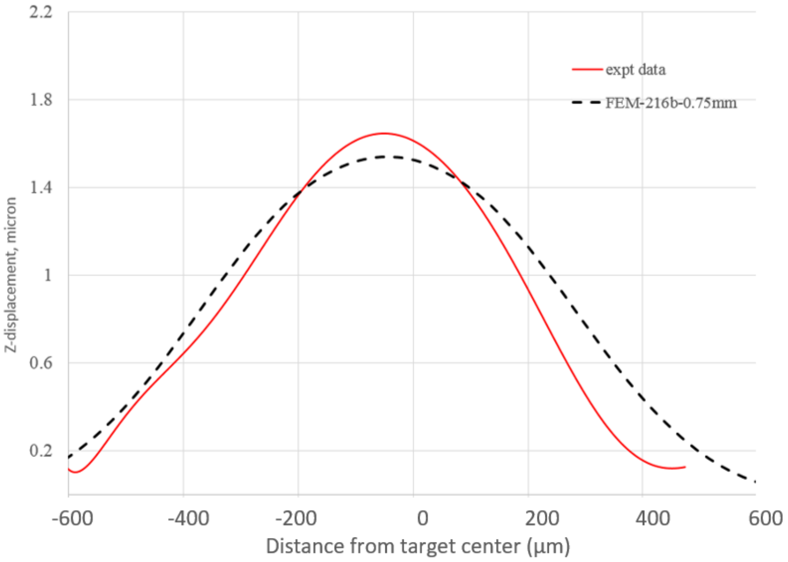


Figure 1.2: Comparison between experimental data and simulations from previous analyses on a 0.75 mm thick target.

Chapter 2

Problem description

This work focuses primarily on the structural analysis of the beryllium target, in particular a circular one with a diameter of 15 mm and 2 mm thick.

2.1 Mechanical properties

The structural analyses described in the following sections adopted the Johnson-Cook strength model: one of the simplest and most widely used temperature and strain rate dependent strength model. Equation below shows its general form:

$$\sigma_y = (A + B\varepsilon_p^n)(1 + C \ln \dot{\varepsilon}^*)(1 - T^{*m}) \tag{2.1}$$

where:

- A, B, C, n and m are empirically determined parameters;
- ε_p = effective plastic strain;
- $\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}$ effective plastic strain rate for $\dot{\varepsilon}_0$, in units of $\frac{1}{[\text{time}]}$;
- $T^* = \frac{T - T_m}{T_m - T_0}$, where T_0 and T_m are respectively the reference and melting temperatures.

Model parameters for Beryllium S-200FH were determined at Southwest Research Institute (SwRI) by means high strain rate and elevated temperature tests with Split-Hopkinson pressure bar. Strength model has been validated during HRMT24, as described in [1].

2.2 Model

During this HiRadMat tests, we must distinguish two phases. The *first phase* is when the proton beam hits the target and heats it, the length of this phase is a variable parameter of the different tests, in our case it last 3.6 μs. During the heating only a small amount of material in the center of the target is heated while the surrounding cold material prevents thermal expansion. This causes a rapid dynamic effect of pressure waves propagation throughout the specimen (the circular shape of the target allows the waves re-bouncing at edges without interference) that continues to decrease until 0.3 s (*second phase*) when the core temperature has returned to its initial value and another proton beam hits the target.

Chapter 3

Analysis description

3.1 Transient thermal analysis

The first simulation is the transient thermal analysis using ANSYS Workbench. We use the proton beam heat generation table obtained from a previous MARS simulation as an input to calculate the temperature profile in each nodes of the model.

An important thing to consider here is that we do not directly simulate the effect of the proton beam on the material but rather the effect of the heating caused by the beam-material interaction. Hence, to simulate multiple pulses we apply the heat generation for 3.6 μs which is the *pulse length* and then we remove it for the *cooling phase* which last until 0.3 s when another pulse is reapplied. This process repeats 6 times until the simulation is stopped.

At the end of the simulation the results are processed by a user defined sets of commands (`Solution` → `Commands (APDL)`) to extract the nodal temperature profiles and combine them in a .k file to use as a load file for the next LS-DYNA analysis. The command file is available in appendix B.1.

Figure 3.3 shows the temperature profile of a surface node in the center on the model. We can see how the temperature increases in the first 3.6 μs of the heating phase reaching a maximum value of about 500 °C and then how it decreases to its initial value until the next 0.3 s.

3.2 Structural analysis

The second simulation is the structural analysis using LS-DYNA. The model to use with LS-DYNA is the same used in the thermal analysis (figure 3.2), the only thing we have to do to get the .k file with nodes and elements is to create a Workbench LS-DYNA system that shares data with the Transient

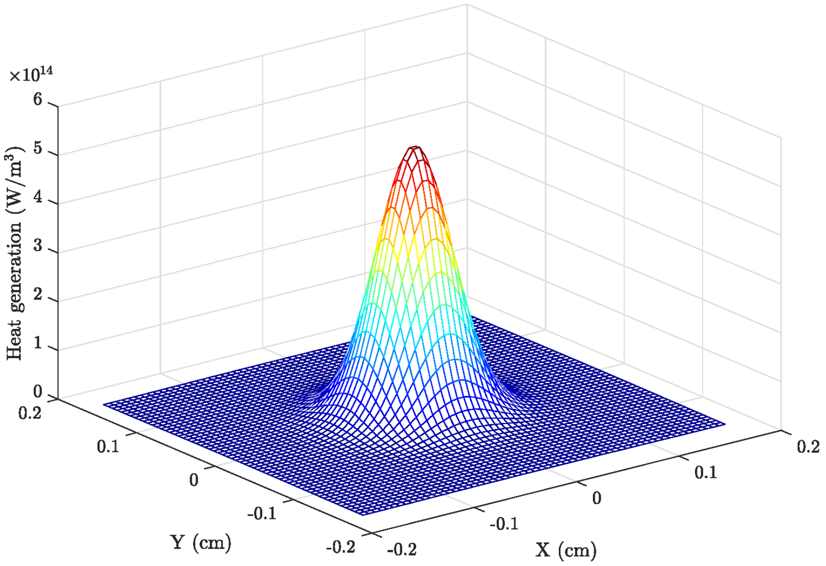


Figure 3.1: Heat generation profile from MARS simulation

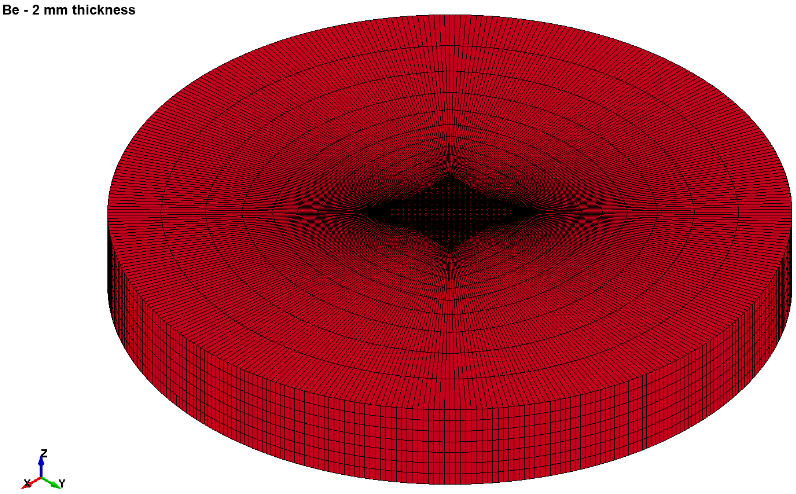


Figure 3.2: FEM model.

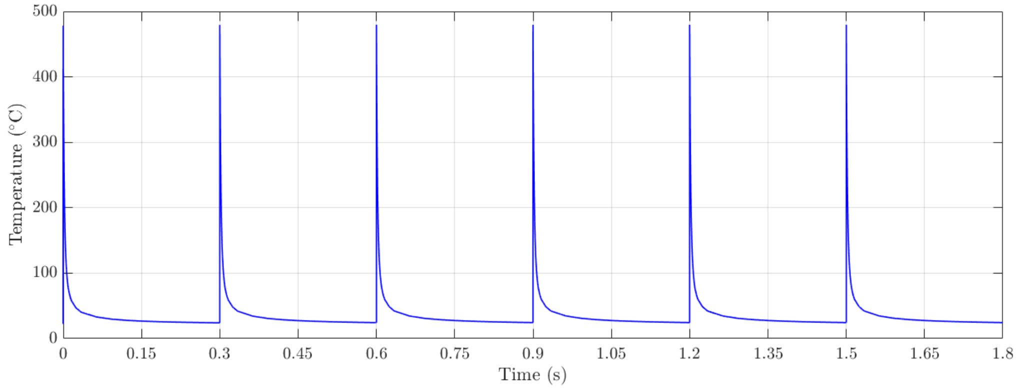


Figure 3.3: Temperature profile in the center of the target

Thermal system and set up a short simulation time. Once we have the .k file we can start working on the simulation parameters.

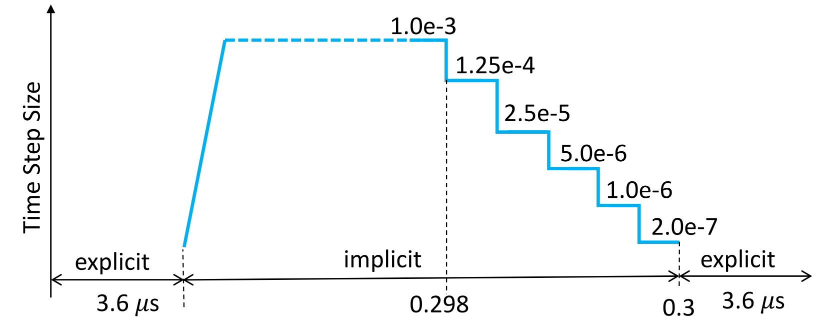
3.2.1 Method

In the structural analysis we want to use an explicit solver during the 3.6 μs of the heating phase when the dynamic effects are greater and then switch to an implicit solver during the cooling phase.

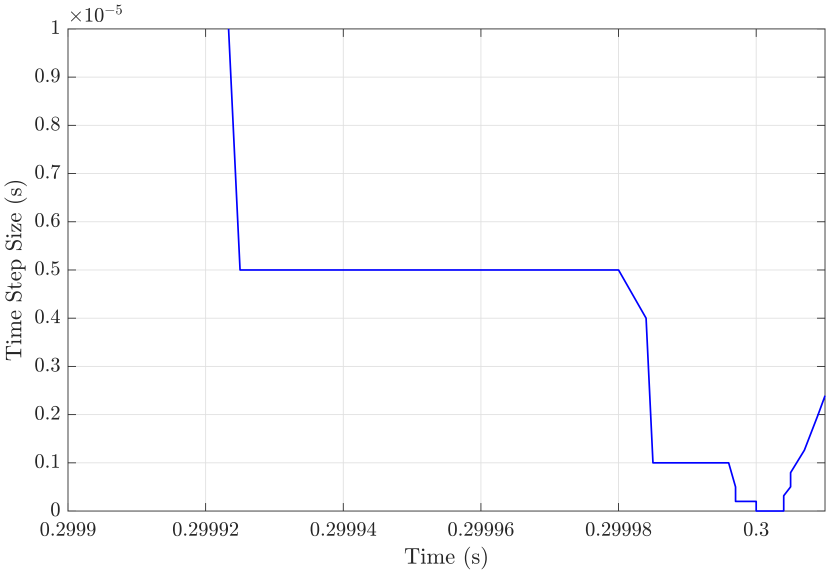
Past analyses to simulate multiple pulses had problem switching from implicit to explicit after the first cool down, this because we had no control over time step size during this phase and, using an automatic time step control with fixed maximum time step size, when we arrived near 0.3 s the last time step was always to big to go beyond the next 3.6 μs.

To solve this problem, the maximum time step size to be used by implicit solver has been defined as a function of time, in this way the time step size is adjusted automatically so that the value of each point in the curve is reached "exactly". In our curve, we decided to step by step decrease the maximum time step size when time approaches 0.3 s so that the last 2×10^{-7} s step size can't exceed the 3.6 μs of the heating phase.

In figure 3.4a we can see how the maximum time step has been defined using the curve option in the automatic time step control while the figure 3.4b represents the actual time step size from the simulation.



(a) Maximum time step size control



(b) Actual time step size from simulation

Figure 3.4: Time step size in implicit analysis (LS-DYNA)

Chapter 4

Results analysis

4.1 Simulation

From figure 4.1 we can see how the material behaves in the middle of the target: during the heating phase out of plane displacement reaches about $4\,\mu$m and then decreases up to a fairly stable value of $0.4\,\mu$m, this means that the specimen exhibited a permanent plastic deformation and that the value of the plastic deformation doesn't change so much between the end of each pulses as we can see from figure 4.2, this because of the hardening behavior of the material model used in the simulation.

Large deformations during each pulse are possible without cracking because of the high temperature $\sim 500\,°$C that softens the material.

Figure 4.3 shows the surface profile of the target after six pulses. A difference which is immediately evident with respect to the result of the 0.75 mm thick target in figure 1.2 is the shape of the deformed surface at

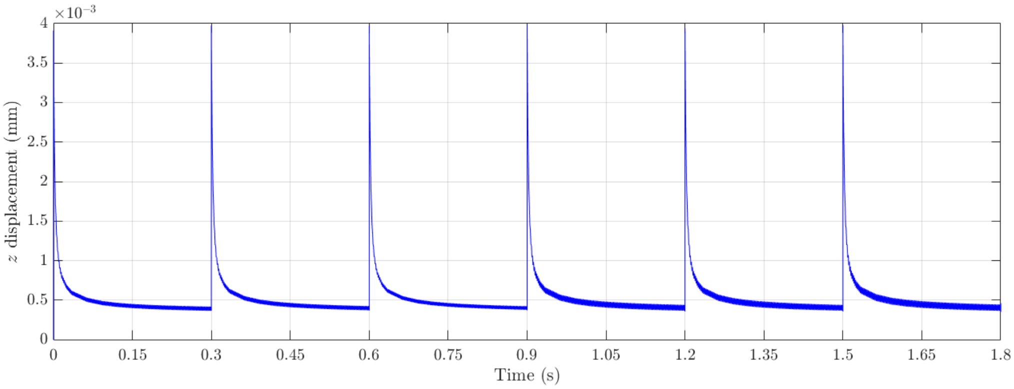


Figure 4.1: z displacement in the center of the target during the simulation.

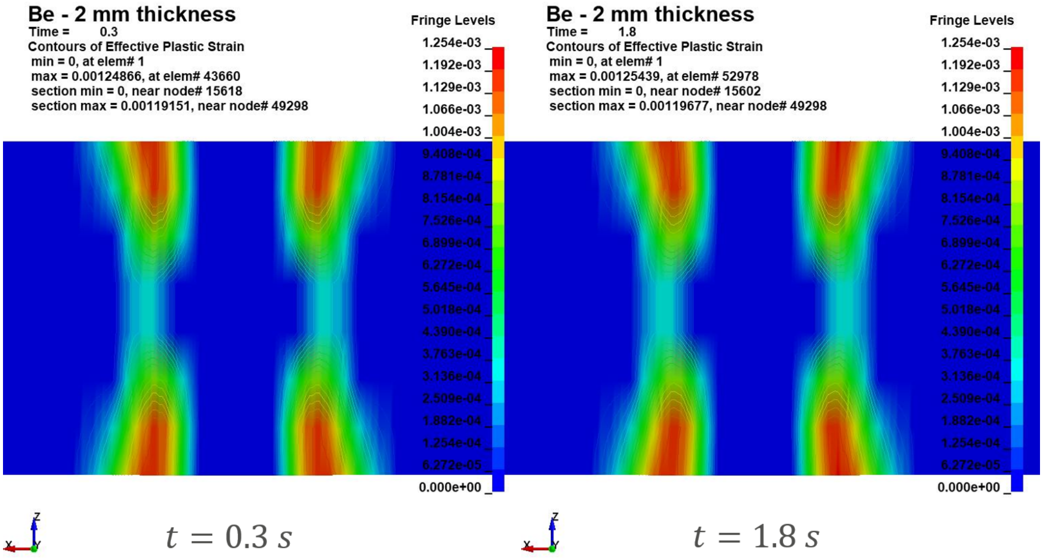


Figure 4.2: Comparison between the plastic strain at the end of the first and last cooling phases.

the center, the 2 mm thick specimen shows a double peak with a relative minimum. This behavior is due to J-C material model used for beryllium, in fact, the center of the target is the point with the highest temperature but it is also the one in which the strain rate is higher. These two parameters (temperature and strain rate) acts with opposite effects on yield stress: high temperature softens the material while high strain rate strengthens it. In figure 4.4 we can see that the higher strain rate is at the surface, right in the center.

4.2 Experimental data comparison

From figure 4.5 we can see that there's not a good matching between the results of our simulations and experimental data with significant differences up to 50% even if the slope of the curve section matches quite well.

This important difference may be due to multiple factors. One of which could be not having considered the influence of temperature and pressure on the material properties, in addition the non-symmetry of the experimental data could be due to a not perfect homogeneity of the beryllium with small defects inside. Finally another factor could perhaps be a small offset between each pulse with consequent heating not concentrated only in the center of the target.

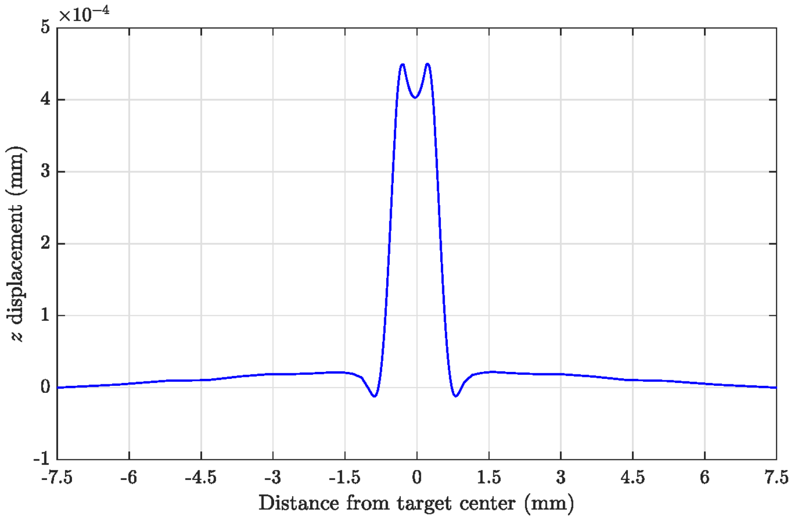


Figure 4.3: Deformed surface after six pulses (2 mm thick target).

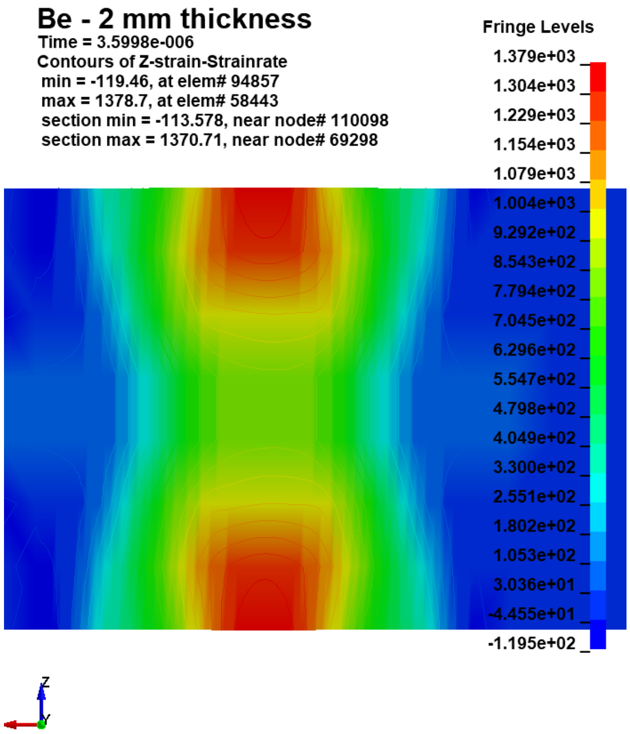


Figure 4.4: z strain rate at 3.6 μs .

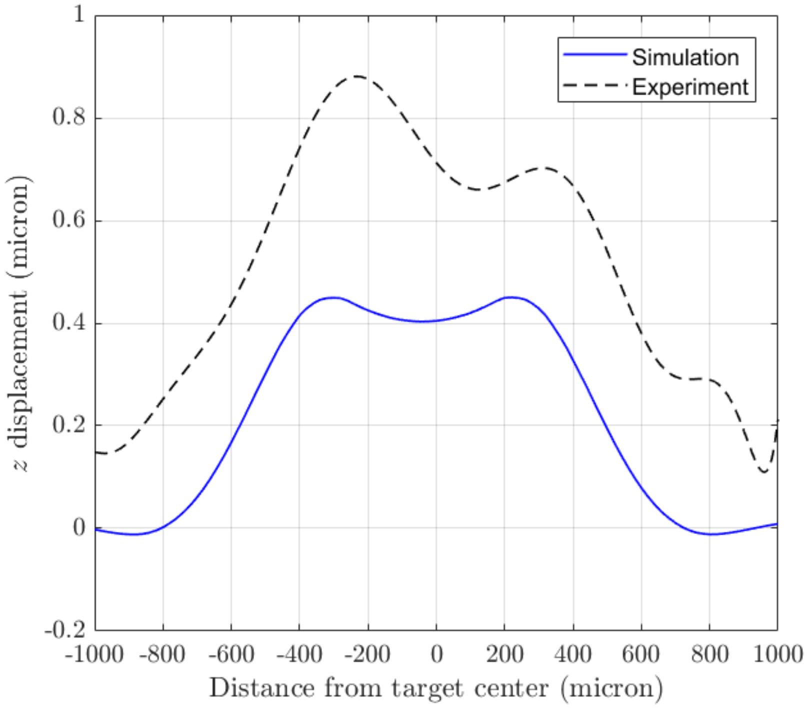


Figure 4.5: Comparison between experimental data and simulations (2 mm thick target).

Chapter 5

Equation of State

5.1 Introduction

Equation of State is a functional relationship among the thermodynamic variables of a system, in its general form, it can be written as

$$f(P, V, T) = 0 \tag{5.1}$$

in which P is the pressure, V is the specific volume (of $\rho = 1/V$, density) and T is the temperature (or E, internal energy). The simplest EOS that can be considered is the bulk modulus that expresses the relation between the volumetric deformation and pressure.

A knowledge of the EOS of matter is necessary to describe the dependence of thermodynamic properties on its microscopic internal structure and are also vital for a variety of applications, particularly numerical simulation of dynamic processes which involves high temperature and pressure with consequent effects on density.

In the previous analyses, the J-C model had fixed density. However, this setting is not representative of the actual model because of the large pressure due to beam heating (in our case, pressures up to 1 GPa, and therefore the hydrostatic components of the stress state). The use of a user defined EOS will take in account the density changes during the whole simulation.

5.2 Model

The model we implemented is the Birch-Murnaghan EOS (see [6] for more details) given by:

$$P(\rho) = \frac{3}{2K_{T0}}(\eta^{7/3} - \eta^{5/3})\big[1 + \frac{3}{4}(K_T{}' - 4)(\eta^{2/3} - 1)\big] \tag{5.2}$$

$$e_M(\rho) = \frac{9}{8K_{T0}\rho_0}(\eta^{2/3} - 1)^2\big[1 + \frac{1}{2}(K_T{}' - 4)(\eta^{2/3} - 1)\big] \tag{5.3}$$

where:

- e_M is the internal energy per unit mass;
- $\eta = \rho/\rho_0 = V_0/V$;
- ρ_0 is a reference density;
- K_{T0} is the initial bulk modulus;
- $K_T{}' = \frac{dK_T}{dP}$.

5.3 Implementation

The user subroutine is called twice for each integration point in each element. The first call requires the EOS to calculate the bulk modulus, and the second updates the pressure and internal energy (the subroutine do not use the parameter η but rather its reciprocal $v_R = 1/\eta$).

To calculate the bulk modulus, we use the definition:

$$K_T = -V\left(\frac{dP}{dV}\right)_T = -V\frac{dP}{dv_R}\frac{dv_R}{dV} = -\frac{1}{v_R}\frac{dP}{dv_R} \tag{5.4}$$

To update the pressure and internal energy we only need to rewrite them as a function of v_R being careful to convert the internal energy per *unit mass* in energy per *unit reference volume*:

$$E = e_M\rho_0 \tag{5.5}$$

The Fortran code which implement the user defined EOS is shown in appendix B.2. Unfortunately it was not possible to carry out simulations with the new EOS because of licensing problems that have not allowed to compile the new code to bet the custom executable file needed for the simulation.

Appendix A

FEM Overview

The finite element method (FEM), or finite element analysis (FEA), is based on the idea of building a complicated object with simple blocks, or, dividing a complicated object into smaller and manageable pieces.

What we do in FEM softwares is a simplification of the problem:

- we make a discretization of the continuous model by subdividing it into certain number of elements (mesh);
- for each element we can write simple equations and then we can assemble them into a larger system;
- the time interval in which the system behavior is to be studied is subdivided into successive time intervals (*steps*). Known the state of the system (displacements, velocities and accelerations) at time t_{n-1} we calculate the new state at the time t_n.

A.1 Solution algorithms

Depending on the nature of the problem to be analyzed and therefore of the equations that govern the phenomenon we can distinguish two kind of algorithms (or time integration schemes) that are used in FEM softwares to solve the differential equation system: Implicit and Explicit methods.

- Implicit algorithms give the configuration at time t+1 as an implicit function of the unknown as well as earlier configurations, e.g. $y_{n+1} = f(y_{n+1}, y_n)$.
- Explicit algorithms give the configuration at time t+1 as an explicit function of earlier configurations, e.g. $y_{n+1} = f(y_n)$.

In **Implicit** algorithms a step by step calculation method is used in which an appropriate *convergence criterion* allows the analysis to continue or not, possibly reducing the time step size, based on the accuracy of the results at the end of each step. In the Newmark time integration scheme, if the current step is n, the equation of motion estimate at time $n+1$ is:

$$Ma'_{n+1} + Cv'_{n+1} + Kd'_{n+1} = F^{\text{ext}}_{n+1} \tag{A.1}$$

where with the superscript $'$ we indicate the acceleration, velocity and displacement estimates an step $n+1$.

Displacements and velocity estimates are formulated starting from the values calculated in the previous step:

$$\begin{aligned} d'_{n+1} &= d^*_n + \beta a'_{n+1}\Delta t^2 \\ v'_{n+1} &= v^*_n + \gamma a'_{n+1}\Delta t \end{aligned} \tag{A.2}$$

Replacing (A.2) in (A.1) and simplifying we obtain:

$$M^* a'_{n+1} = F^{\text{residual}}_{n+1} \tag{A.3}$$

from which we obtain the acceleration estimate by inverting the mass matrix M^*. By gradually reducing the time increment, the solution tends to an asymptote of the value of the acceleration estimate at step n + 1.

In **Explicit** algorithms we do not have convergence problem because the analysis in not conditioned by a convergence criterion and the time step size is defined at the beginning and remains constant during the analysis. In explicit algorithm, the equation of motion is :

$$Ma_n + Cv_n + Kd_n = F^{\text{ext}}_n \tag{A.4}$$

where with the subscript n we indicate the values at step n of acceleration, velocity, displacement and external loads. Acceleration at step n is calculated by inverting the mass matrix M once in the simulation or if it's diagonal, it is possible to calculate the acceleration for each increment and for each degree of freedom by means a system of independent equations. Time increment is defined by the following expressions and it can not be established by the user (which instead it's possible with the use of implicit methods):

$$\begin{aligned} v_{n+1} &= v_{n-1} + a_n \frac{\Delta t_{n+1} + \Delta t_{n-1}}{2} \\ d_{n+1} &= d_n + v_{n+1}\Delta t_{n+1} \end{aligned}$$

Time step size is calculated as:

$$\Delta t = \min\left(\frac{L}{\sqrt{E/\rho}}\right) \tag{A.5}$$

where L is the characteristic length of the element and $\sqrt{E/\rho}$ is shock wave propagation speed in the material.

The advantage of using Explicit algorithms is due to the fact that, although the time increments are much smaller than those of Implicit algorithms (usually the increments used with the explicit method are from 100 to 1000 times lower than those used in the implicit method), there is a high saving of calculation time in analysis that involve high-speed shocks or problems in which there are large dissipations or transfers of energy and wave propagation impact (events occurring in very small time intervals).

Appendix B

Codes

B.1 Workbench command file

```
! Commands inserted into this file will be executed
!   immediately after the ANSYS /POST1 command.

! Active UNIT system in Workbench when this object
!   was created:  Metric (m, kg, N, s, V, A)
! NOTE:  Any data that requires units (such as mass)
!   is assumed to be in the consistent solver unit system.
! See Solving Units in the help system for more information.

! This updated script allows direct copying and pasting
!   into keyword files
! Changes made to previous script:
!        - no period output after value
!        - 20 character space for ordinate and abscissa


*GET,Noofnodes,NODE,0,COUNT

*CREATE,Temp_load,k
*keyword
*LOAD_THERMAL_VARIABLE_NODE
$#       nid       ts       tb       lcid
*end

*DIM,LC,ARRAY,Noofnodes,1,1, , ,
*DIM,N,ARRAY,Noofnodes,1,1, , ,
*DIM,ts,ARRAY,Noofnodes,1,1, , ,
*DIM,tb,ARRAY,Noofnodes,1,1, , ,
*VFILL,LC,RAMP,1,1,
*VFILL,n,RAMP,1,1,
*VFILL,ts,RAMP,1,0,
*VFILL,tb,RAMP,22,0,

*CFOPEN,Temp_load,k, ,APPEND
*VWRITE,N(1,1),ts(1,1),tb(1,1),LC(1,1)
(F11.0,TL1,F11.0,TL1,F11.0,TL1,F11.0,TL1,'␣')
*CFCLOS

FINISH
/POST26
FILE,'file','rth','.'
/UI,COLL,1
NUMVAR,200
SOLU,191,NCMIT
STORE,MERGE
FILLDATA,191,,,,1,1
REALVAR,191,191
)/GOP    ! Resume printing after UNDO process

*GET,size,VARI,,NSETS

*DIM,T,ARRAY,SIZE,1   !Noofnodes   ! Create array parameter

! Save time history variables to file time
*DEL,P26_EXPORT
*DIM,P26_EXPORT,TABLE,size,1  !Noofnodes
VGET,P26_EXPORT(1,0),1

*do,i,1,Noofnodes
        NSOL,2,i,temp
        STORE,MERGE        ! Store T at node i
        VGET,T(1,1),2      ! Store time history data
                           ! of variable i+1 into Ti
        VGET,P26_EXPORT(1,1),1
/OUTPUT
```

```
*dim, mystring, string,80
mystring(1) = '*DEFINE_CURVE'
*cfopen, Temp_load, k,   ,Append
*vwrite, mystring(1),  mystring(9),  mystring(17),  mystring(25),  mystring
(5A)
*cfclos

             /OUTPUT,
             *CFOPEN, Temp_load, k,   ,APPEND
             !*CFWRITE,          %i%
             *VWRITE, i,0,1,1,0,-22,0
(F11.0,TL1,F11.0,TL1,F11.0,TL1,F11.0,TL1,F11.0,TL1,F11.0,TL1,F11.0,TL1
             *VWRITE,0,22,  ,  ,  ,  ,  ,
(E20.12,E20.12)
VWRITE,P26_EXPORT(1,0),T(1,1),  ,  ,  ,  ,  ,
(E20.12,E20.12)
             *CFCLOS
VARDEL,2
*enddo

/OUTPUT,
mystring(1) = '*end'
*cfopen, Temp_load, k,   ,APPEND
*vwrite, mystring(1),mystring(9),mystring(17),mystring(25),mystring(33)
(5A)
*cfclos
```

B.2 User defined EOS subroutine

```
      subroutine ueos22s(iflag,cb,pnew,hist,rho0,eosp,
     & specen,df,dvol,v0,pc,dt,tt,crv,first)
c
c*************************************************************
c|  Livermore Software Technology Corporation  (LSTC) |
c|  ------------------------------------------------  |
c|  Copyright 1987-2008 Livermore Software Tech. Corp |
c|  All rights reserved                                |
c*************************************************************
c
c     scalar user implementation of the EOS
c
c***  variables
c          iflag ----- =0 calculate bulk modulus
c                      =1 update pressure and energy
c          cb -------- bulk modulus
c          pnew ------ new pressure
c          hist ------ history variables
c          rho0 ------ reference density
c          eosp ------ EOS constants
c          specen ---- energy/reference volume
c          df -------- volume ratio, v/v0 = rho0/rho
c          dvol ------ change in volume over time step
c          v0 -------- reference volume
c          pc -------- pressure cut-off
c          dt -------- time step size
c          tt -------- current time
c          crv ------- curve array
c          first ----- logical .true. for tt,crv,first time step
c                      (for initialization of the history variables)
c
      include 'nlqparm'
      logical first
c
      dimension hist(*),eosp(*),crv(lq1,2,*)
c
      Kt0 = eosp(1)
      Ktp = eosp(2)
```

```
c
c
c***  calculate the bulk modulus for the EOS contribution
c***  to the sound speed
      if (iflag.eq.0) then
        cb = (-4*(27-49*df**(2/3)+20*df**(4/3))+
                +3*(9-14*df**(2/3)+9*df(4/3))*Ktp)/(8*Kt0*df**(5))
c
c***  update the pressure and internal energy
      else

        pnew = ((1.5/Kt0)*(df**(-7/3)-
                +df**(-5/3))*(1+0.75*(df**(-2/3)-1)*(Ktp-4)))
        pnew = max(pnew,pc)
        specen = (9/(8*Kt0))*((df**(-2/3)-1)**(2))*
                +(0.5*(df**(-2/3)-1)*(Ktp-4)+1)
      endif
c
      return
      end
```

Bibliography

[1] K. Ammigan and P. Hurh. *High-intensity beam tests of thermal shock in irradiated materials*. TSD Topical Meeting. Fermilab, Sept. 2018.

[2] Leonardo Bertini. *Corso di Costruzione di macchine – Elementi di Dinamica Strutturale tramite FEM*. 2016.

[3] S. Evrard and M. Picard. *HiRadMat safety file - Descriptive part*. Safety documentation. CERN, Sept. 2011.

[4] B.K. Godwal, S.K. Sikka, and R. Chidambaram. *Equations of State Theories of Condensed Matter Up to about 10 TPa*. 1983.

[5] Fiona Harden. *HiRadMat Facility Review*. HiRadMAt USER DAY. CERN, Apr. 2018.

[6] Gerald I. Kerley. *Equations of State for BE, Ni, W, and Au*. SANDIA REPORT. Sandia National Laboratories, 2003.

[7] Livermore Software Technology Corporation (LSTC). *LS-DYNA Keyword User's Manual - Volume 1*. 2018.

[8] Livermore Software Technology Corporation (LSTC). *LS-DYNA Keyword User's Manual - Volume 2 Material Models*. 2018.

[9] Livermore Software Technology Corporation (LSTC). *LS-DYNA Theory Manual*. 2018.

[10] C.V. Nielsen et al. "Chapter 2". In: *Modeling of thermo-electro-mechanical manufacturing processes: applications in metal forming and resistance welding*. Springer Science & Business Media, 2012.