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

Matteo Malacarne

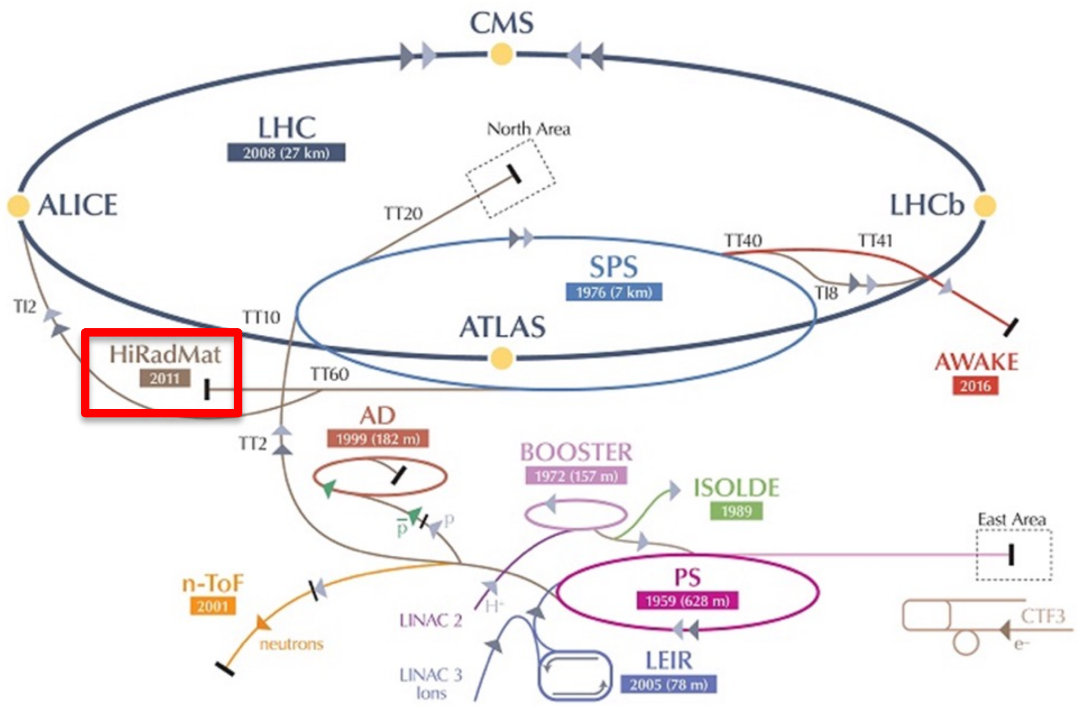
Summer Students at Fermilab - Final Report

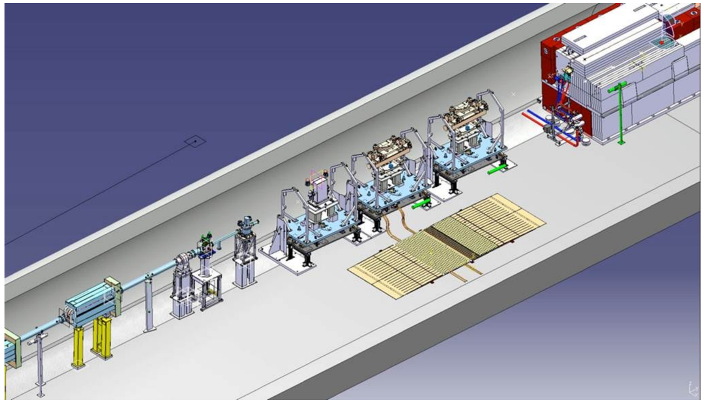
25th September 2018

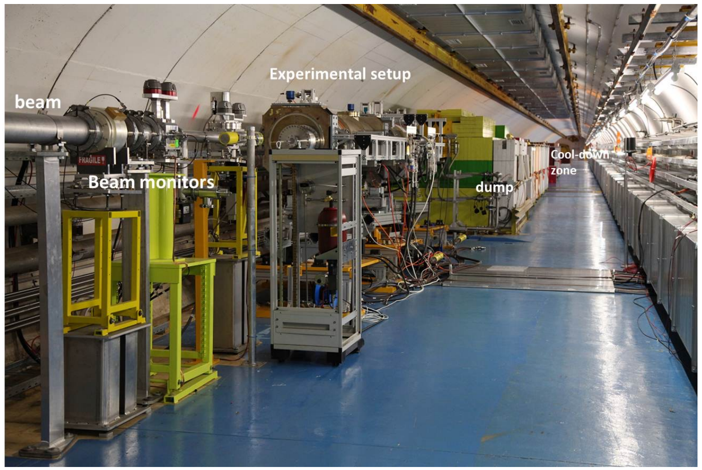
Supervisor:
Sujit Bidhar

Background - HighRADMat test at CERN

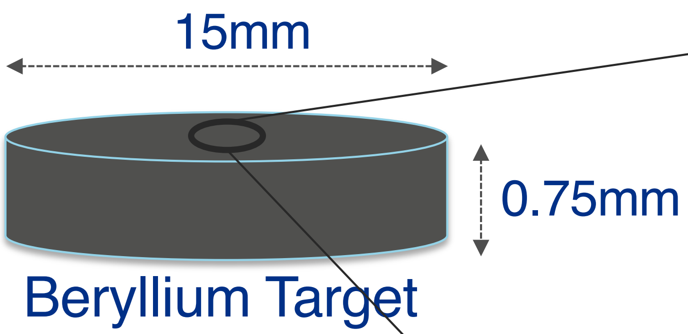
Evaluate the effect of high-intensity pulsed beams on materials or accelerator component assemblies in a controlled environment.



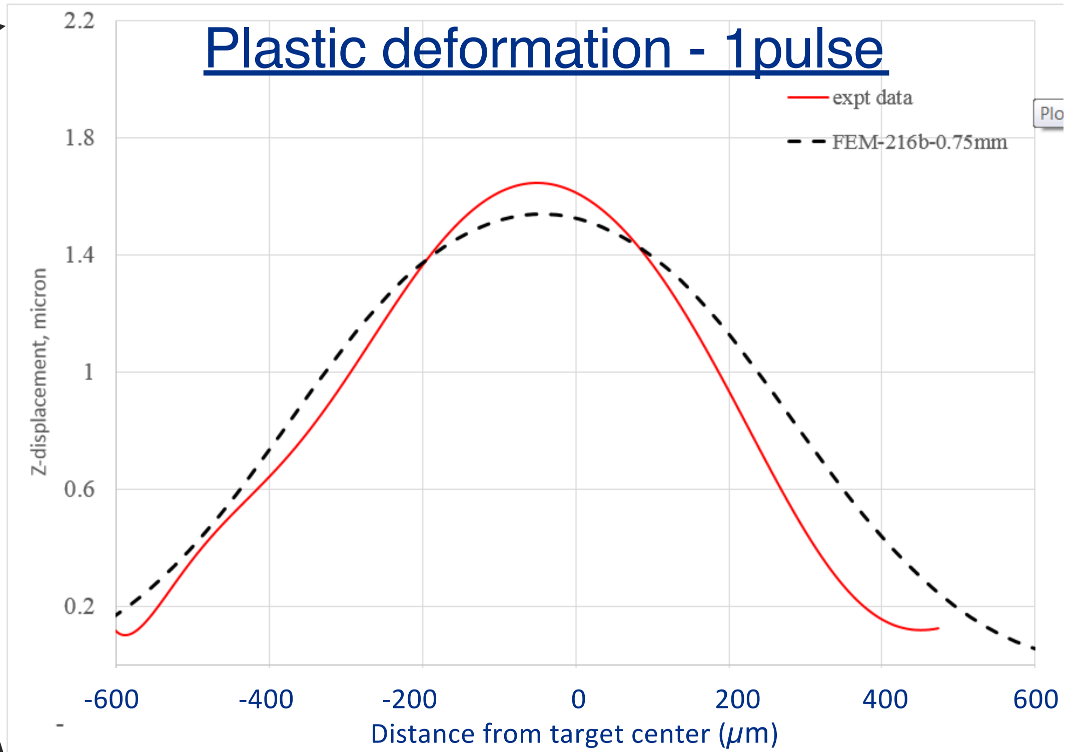




Background - HighRADMat test at CERN



Beam Parameters	
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 μs
Gaussian beam size	1σ: 0.1 – 2 mm



<u>FEM simulation</u> → matches with **<u>experimental data</u>** for single pulse

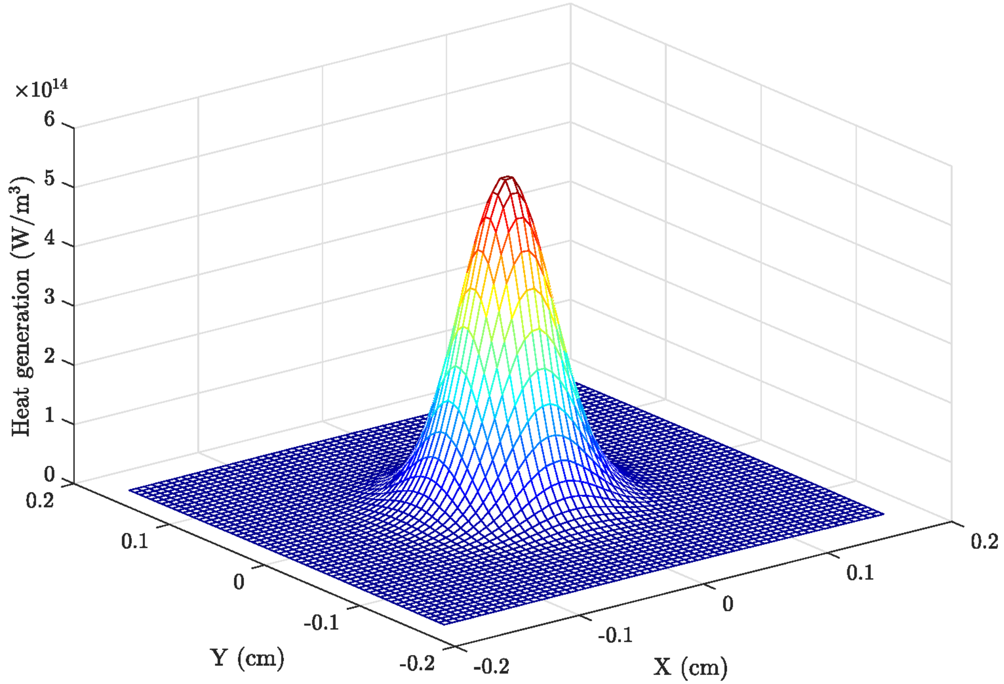
Objectives

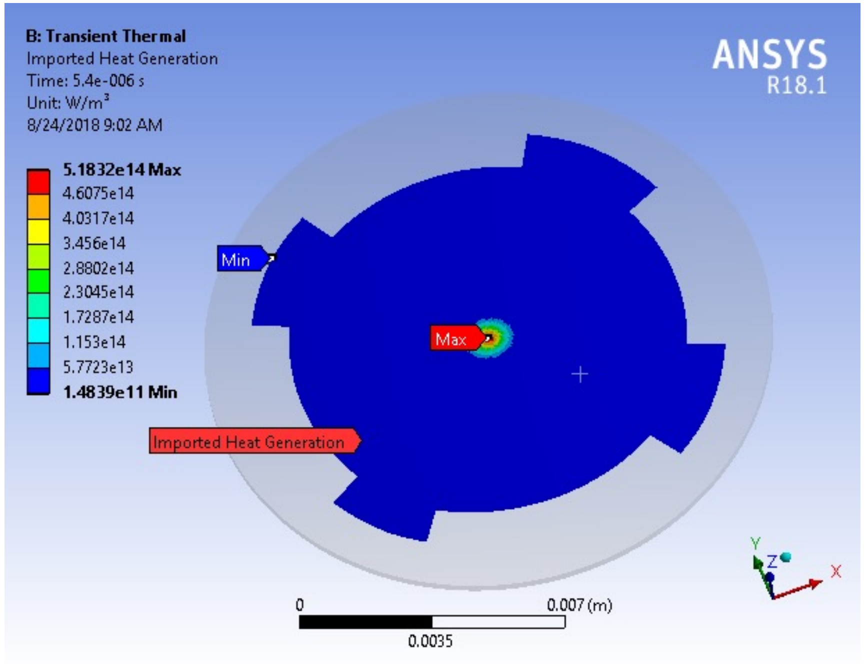
- Use commercial finite element software to simulate **material deformation** due to multiple pulsed beam on beryllium target.

- Implement **user material subroutine** for Equation of State of beryllium.

Analysis Steps

1. Transient Thermal analysis using ANSYS Workbench
   – Input: proton beam heat generation table from MARS simulation





2. Structural analysis using LS-DYNA
   – Import: nodal temperature profile (function of time) from thermal analysis

Johnson-Cook Model

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

Separate effects of:

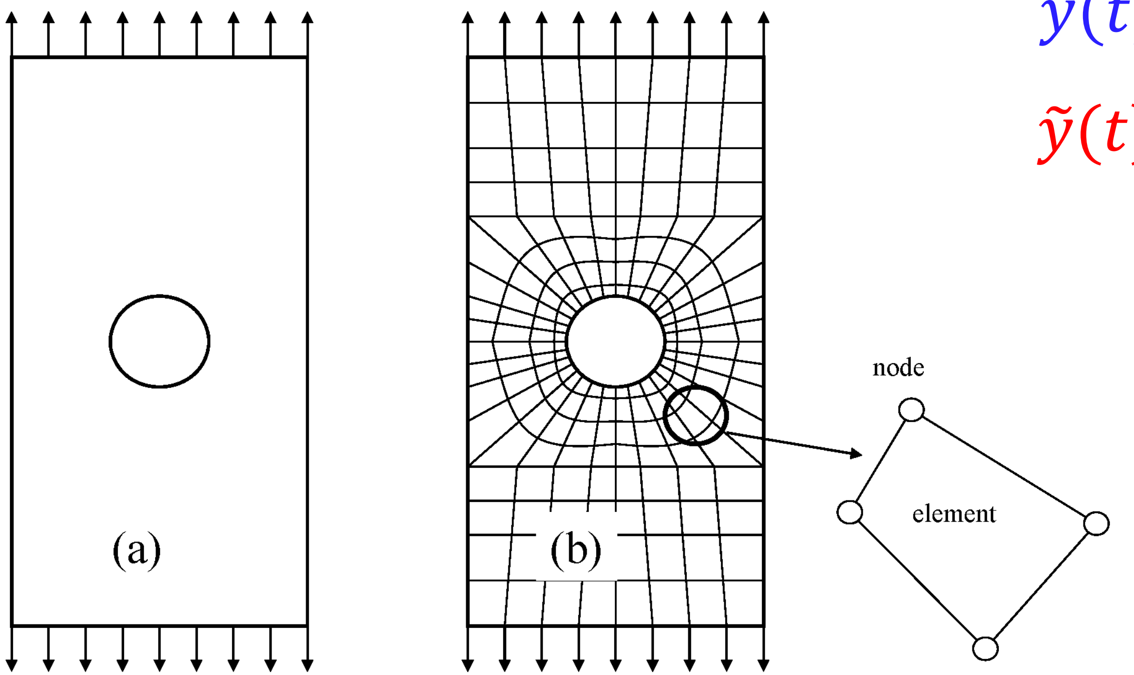
- Temperature
- Strain Rate
- Hardening

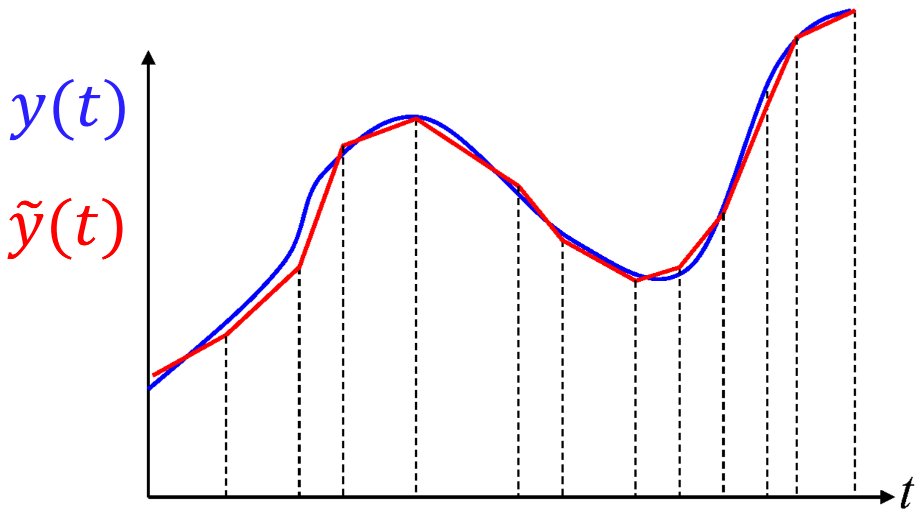
5 material constants

- ε_p = effective plastic strain
- $\dot{\varepsilon}^* = \frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_0}$ = effective plastic strain rate
- $T^* = \frac{T - T_0}{T_m - T_0}$

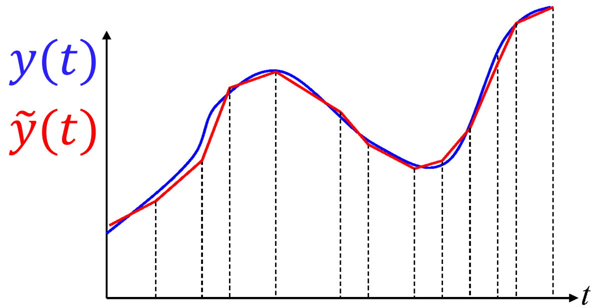
FEM overview (1)

- Subdivision of a large problem into smaller → Elements
- Simple equations for each elements assembled into a larger system
- Approximate values of the unknowns at discrete number of points over the domain





FEM overview (2)



- Implicit

$$y_{n+1} = f(y_{n+1}, y_n)$$

$$Ma'_{n+1} + Cv'_{n+1} + Kd'_{n+1} = F^{ext}_{n+1}$$
$$d'_{n+1} = d^*_n + \beta a'_{n+1}\Delta t^2$$
$$v'_{n+1} = v^*_n + \gamma a'_{n+1}\Delta \mathrm{t}$$

$$a'_{n+1} = {M^*}^{-1} F^{res}_{n+1}$$

- Explicit

$$y_{n+1} = f(y_n)$$

$$Ma_n + Cv_n + Kd_n = F^{ext}_n$$
$$Ma_n = F^{ext}_n - F^{int}_n$$
$$a_n = {M^*}^{-1} F^{res}_n$$
$$\left(a_{n_i} = {F^{res}_n}_i \big/ M^*_i \right)$$

$$v_{n+1} = v_{n-1} + a_n \frac{\Delta t_{n+1} + \Delta t_{n-1}}{2}$$
$$d_{n+1} = d_n + v_n \, \Delta t_n$$

Time Step overview

- Implicit:
  - Time step → Load step
  - Step size depends on convergence criteria

- Explicit:
  - Time step → actual time

$$\Delta t = \frac{L}{\sqrt{E/\rho}}$$

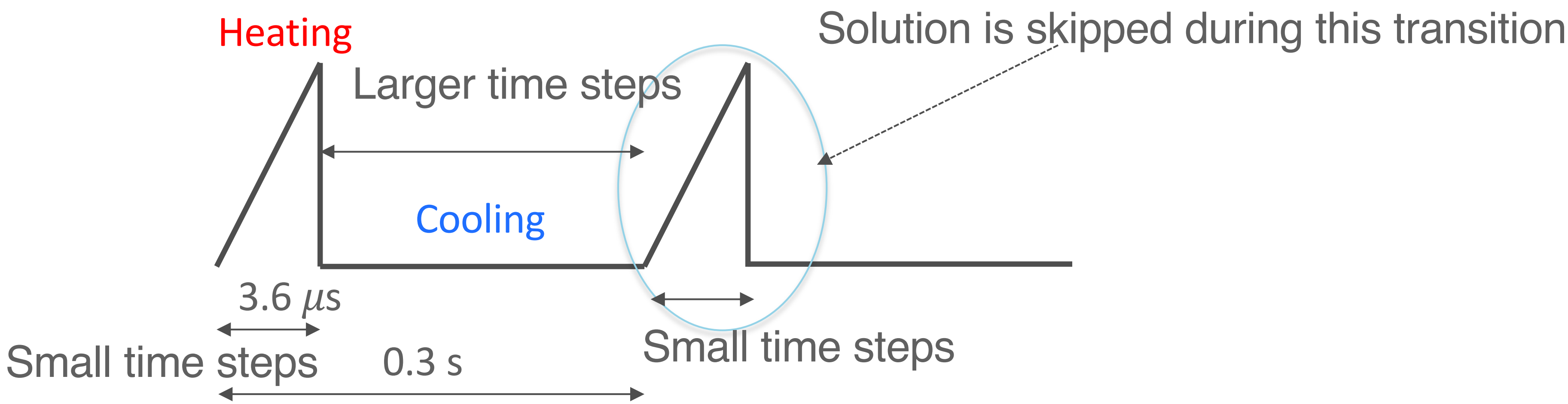
- Implicit
  - No limit to step size
  - Iterative
  - The computational time per load step is relatively high

- Explicit
  - Requires small time steps
  - Not-iterative
  - The computational time per load step is relatively low
  - More efficient
  - Good for dynamic problems (impact and shock problems)

LS-DYNA analysis

- Heating in the first 3.6 μs → explicit solver
- Cooling until 0.3 s → implicit solver

} 6 times

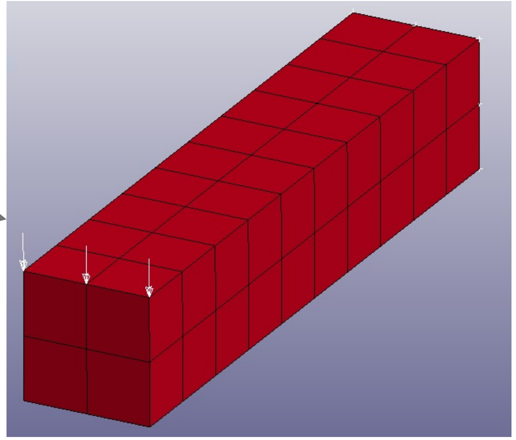


In multiple pulse simulation LS-DYNA had problems **switching from implicit to explicit analysis**

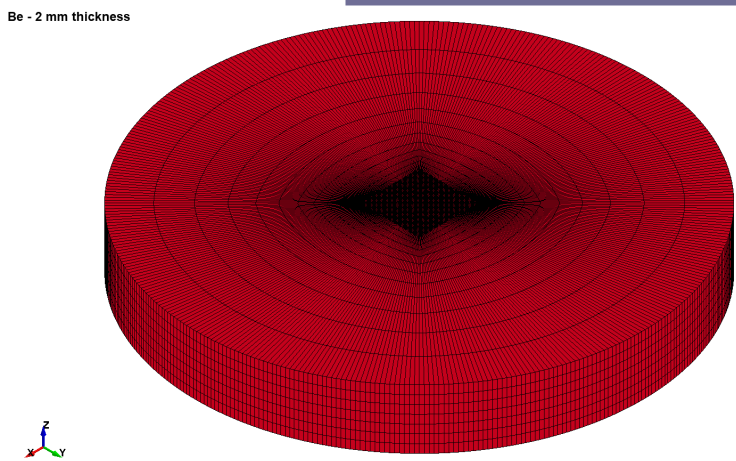
Arrived near 0.3 s, the last time step was always too big to go beyond the next 3.6 μs of the explicit analysis → <u>no control over time step size</u>

Status

- Familiarize with LS-DYNA and in particular its options regarding implicit and explicit analysis.
- Use simplified model to optimize time steps
  - same element size
  - same load curves
  - same solver options



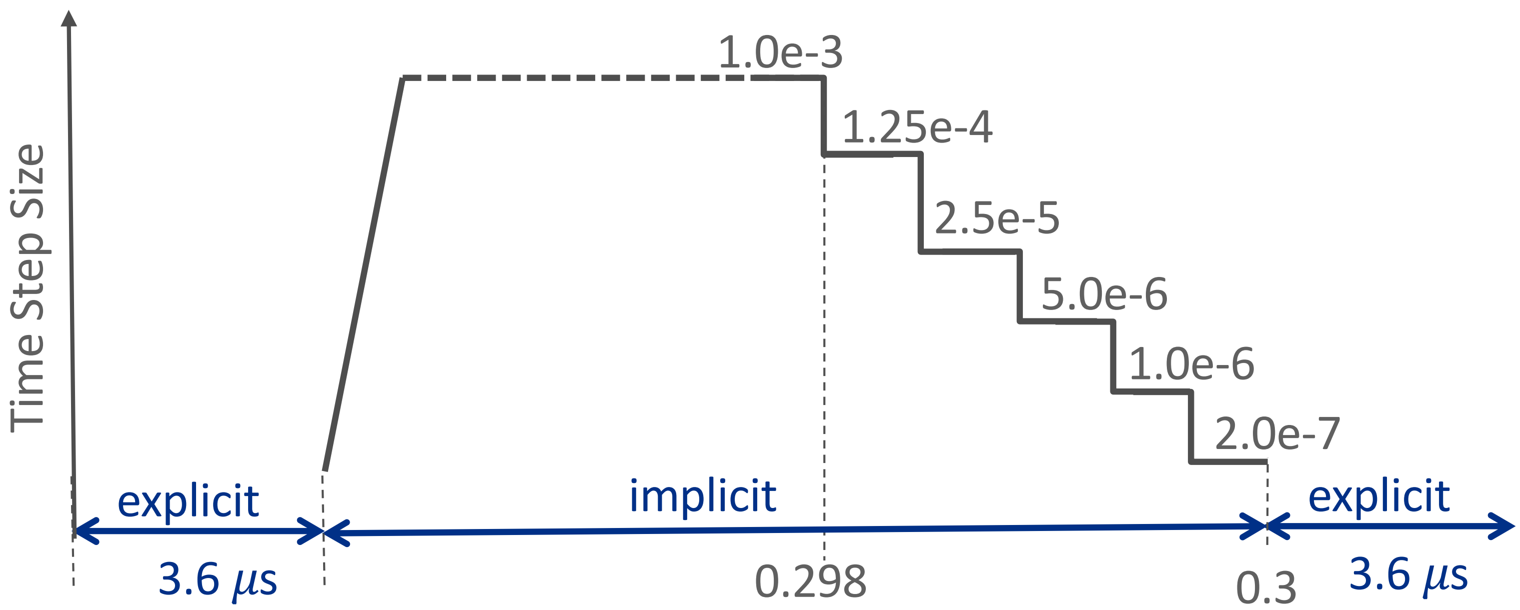
- Implement in actual model



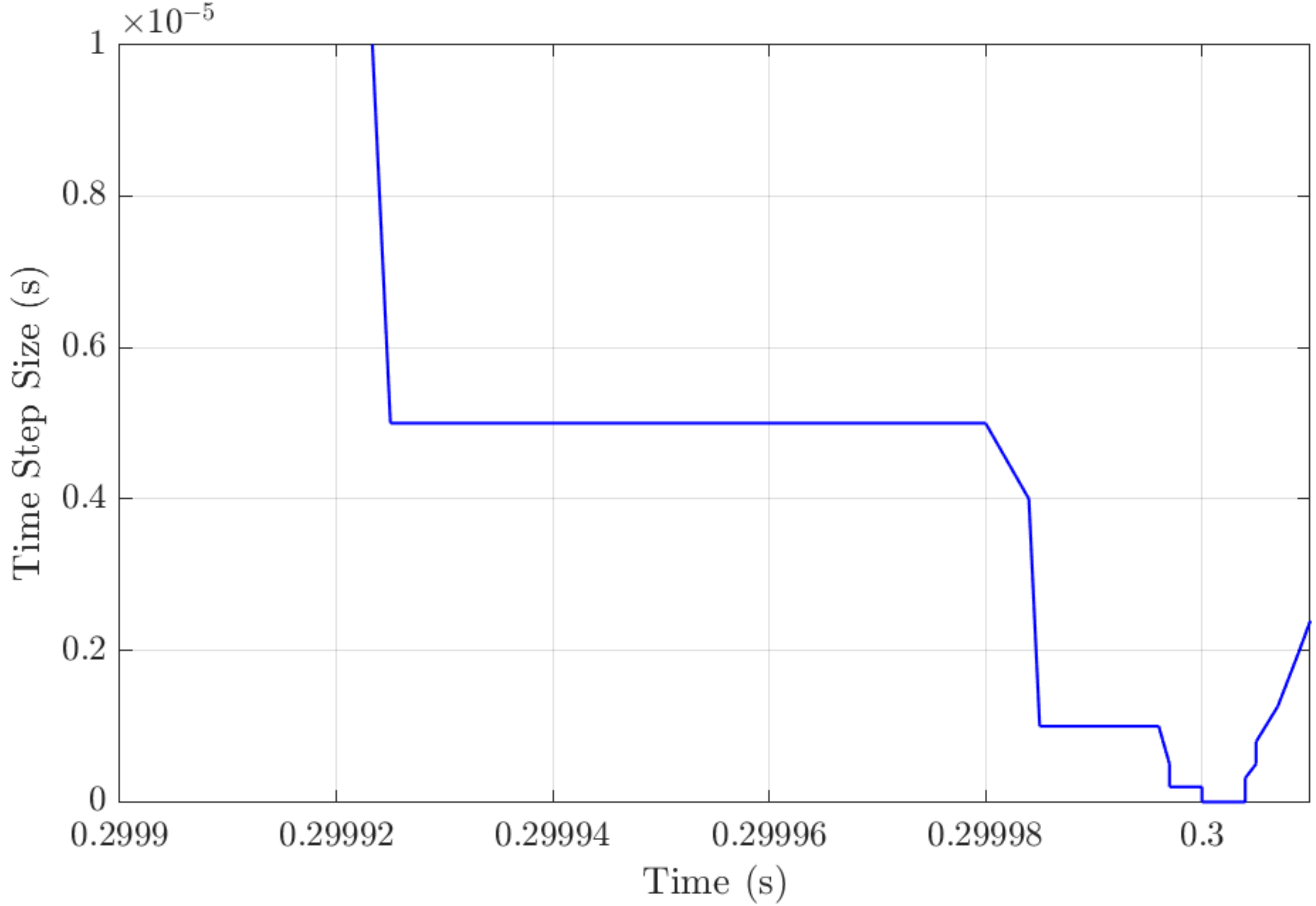
Fermilab

Methodology

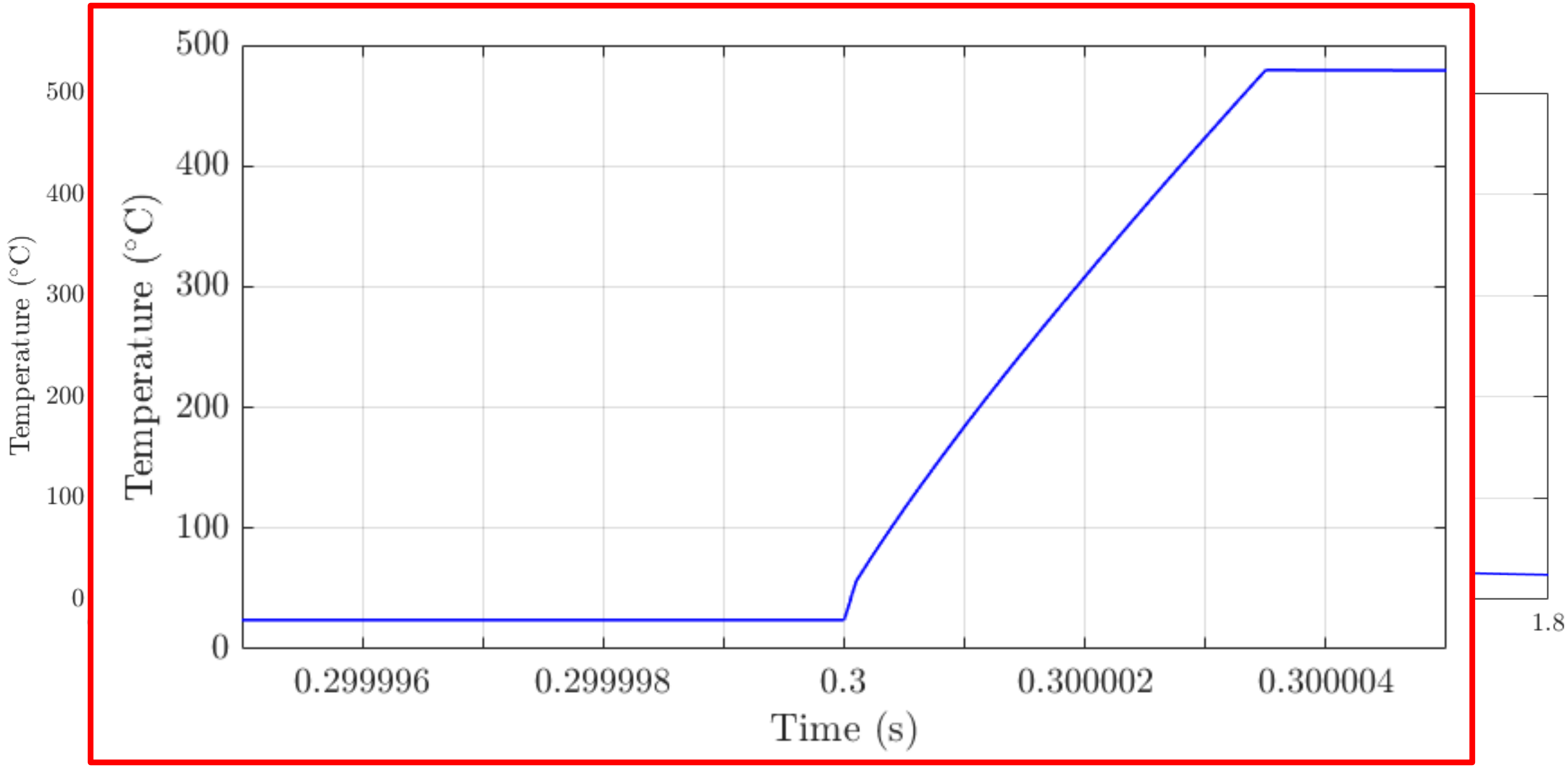
- Decrease the maximum time step size in implicit analysis just before the explicit analysis.



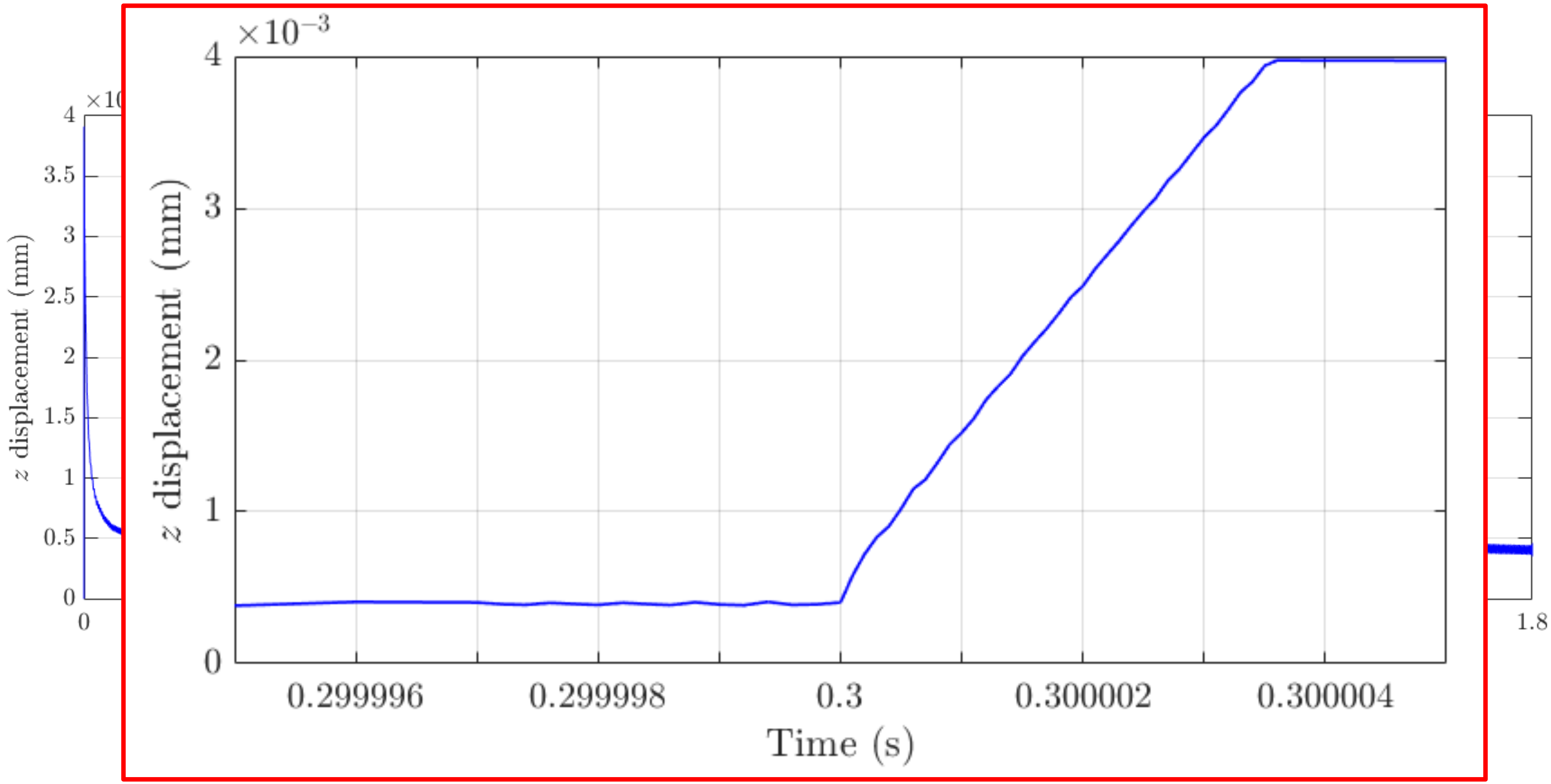
Results – Time Step Size



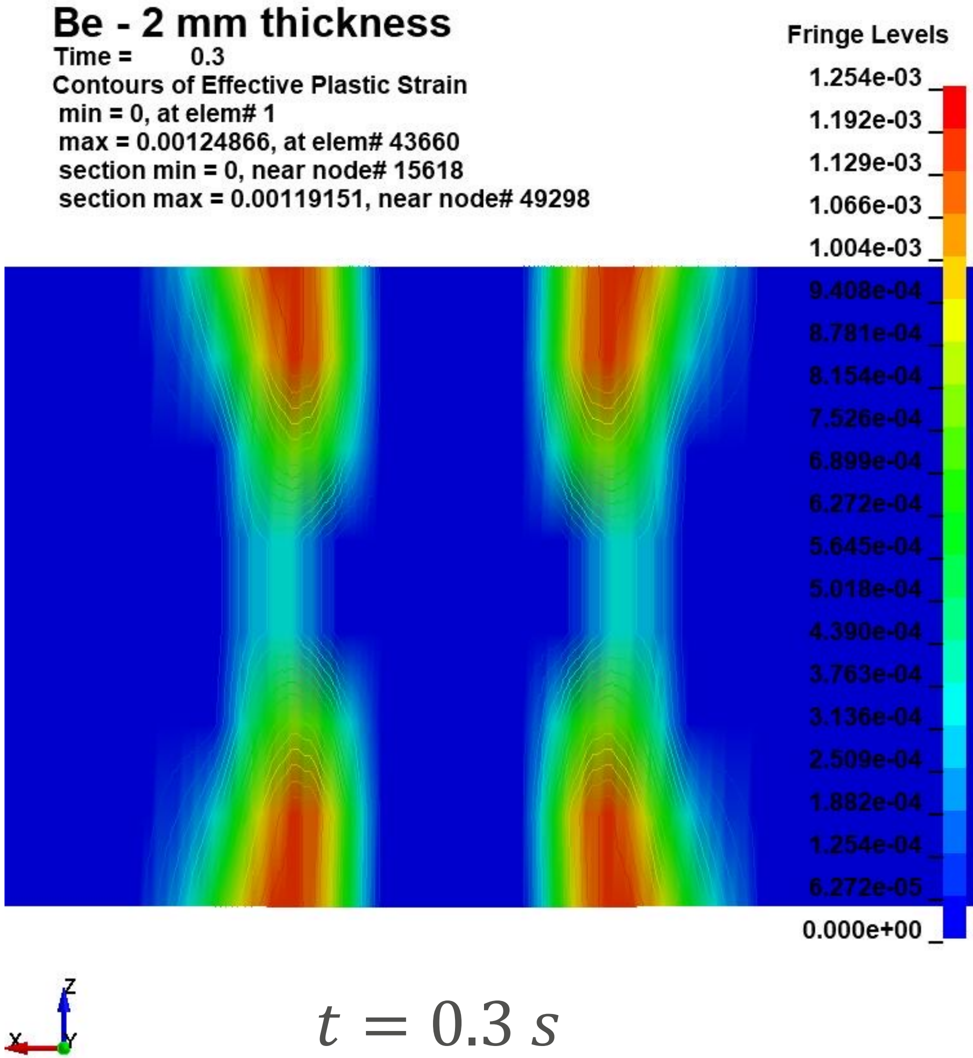
Results – Temperature load



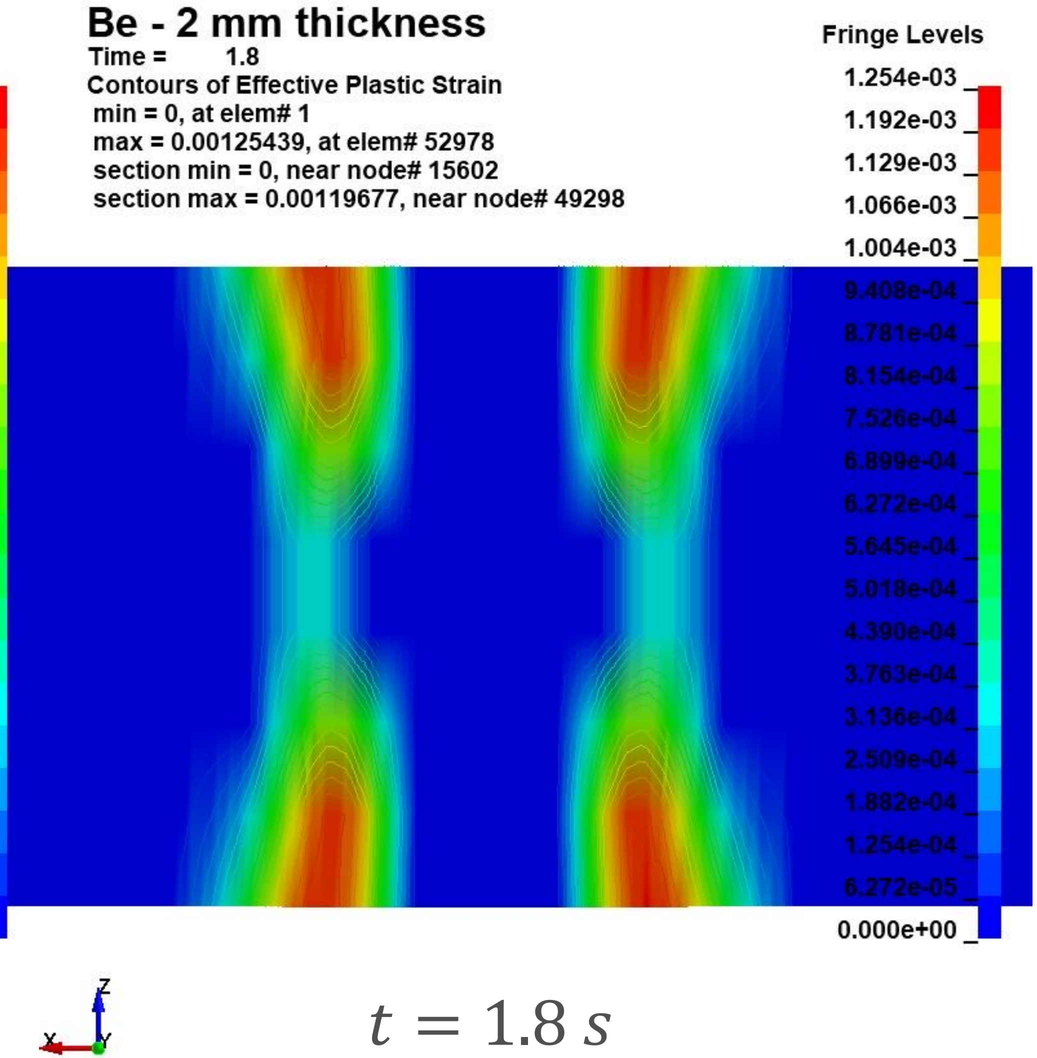
Results – Displacements



Results – Plastic Strain

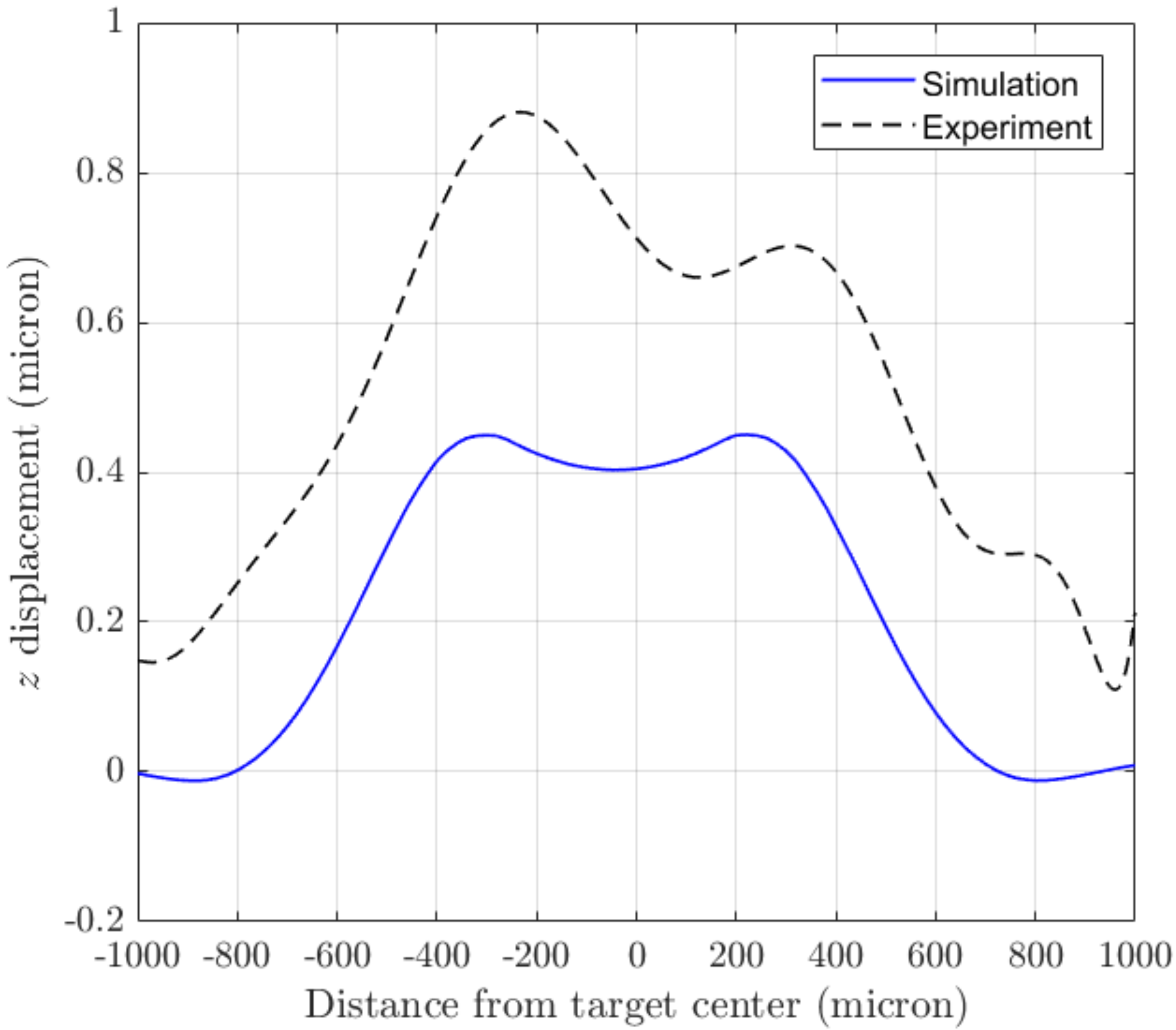


$t = 0.3\ s$

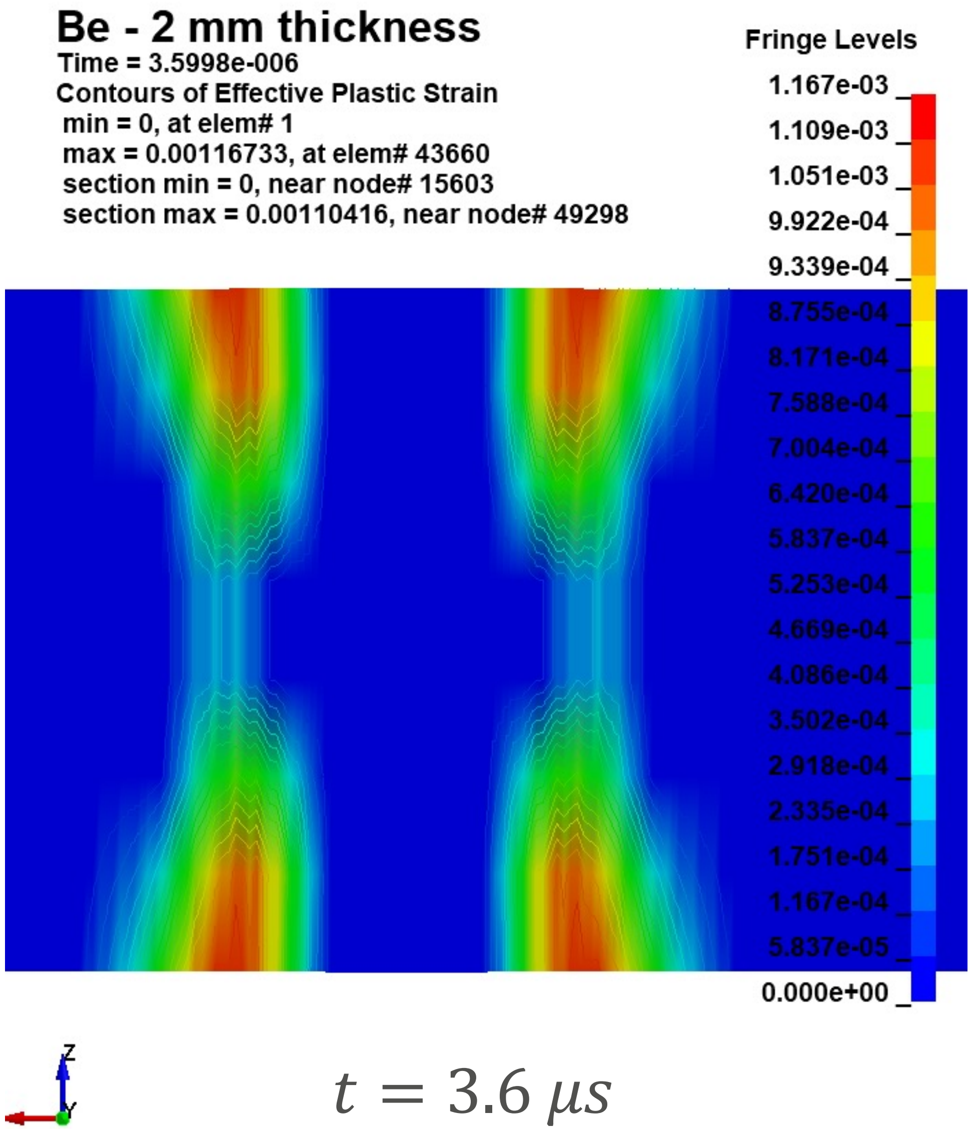


$t = 1.8\ s$

Comparison FEM vs. Experimental Results

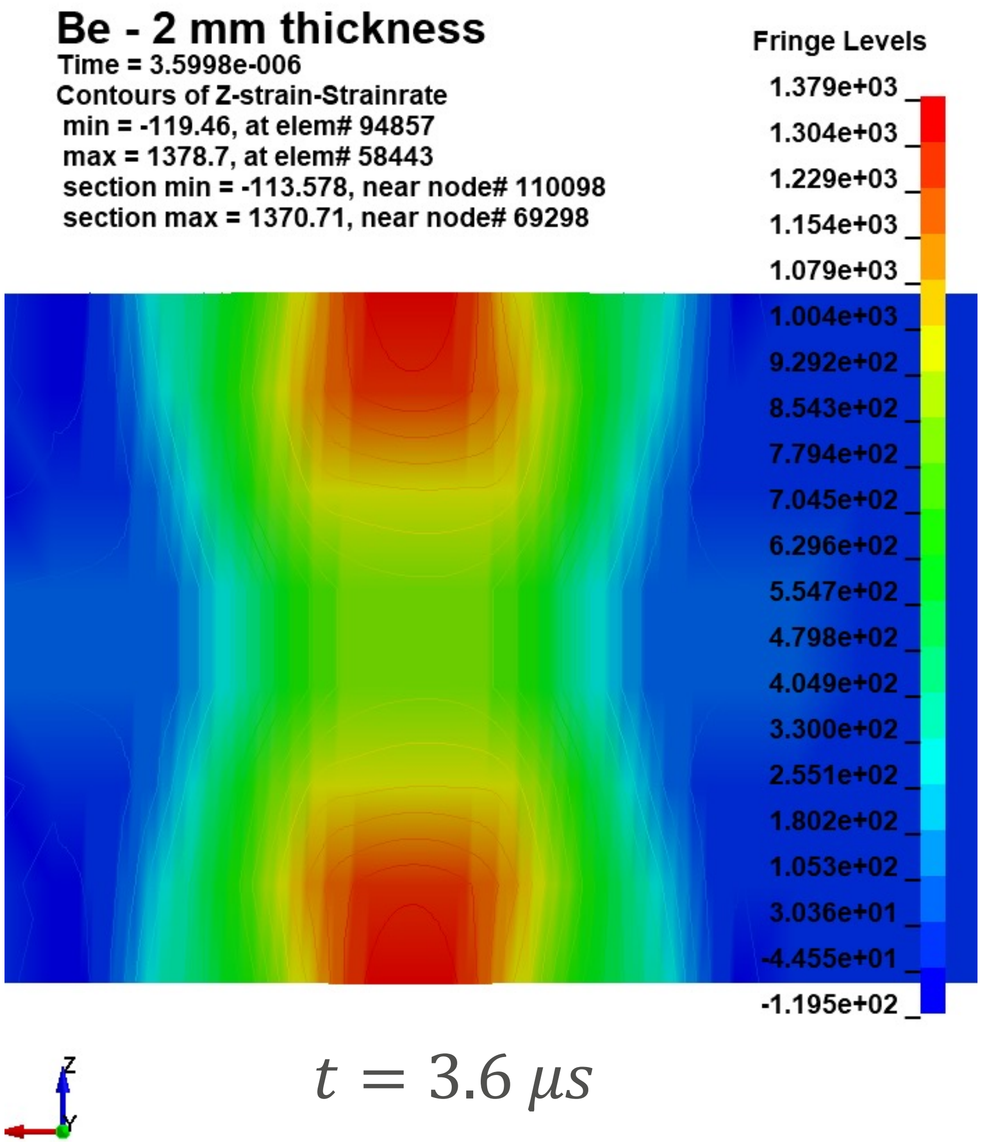


Results – Plastic Strain



$t = 3.6\ \mu s$

Results – Strain Rate



$t = 3.6\ \mu s$

Equation of State (EOS) - Introduction

Functional relationship among the thermodynamic variables for a system:

$$f(P, V, T) = 0$$

P = pressure

V = volume (or $\rho = {}^{1}/_{V}$, density)

T = temperature (or E, internal energy)

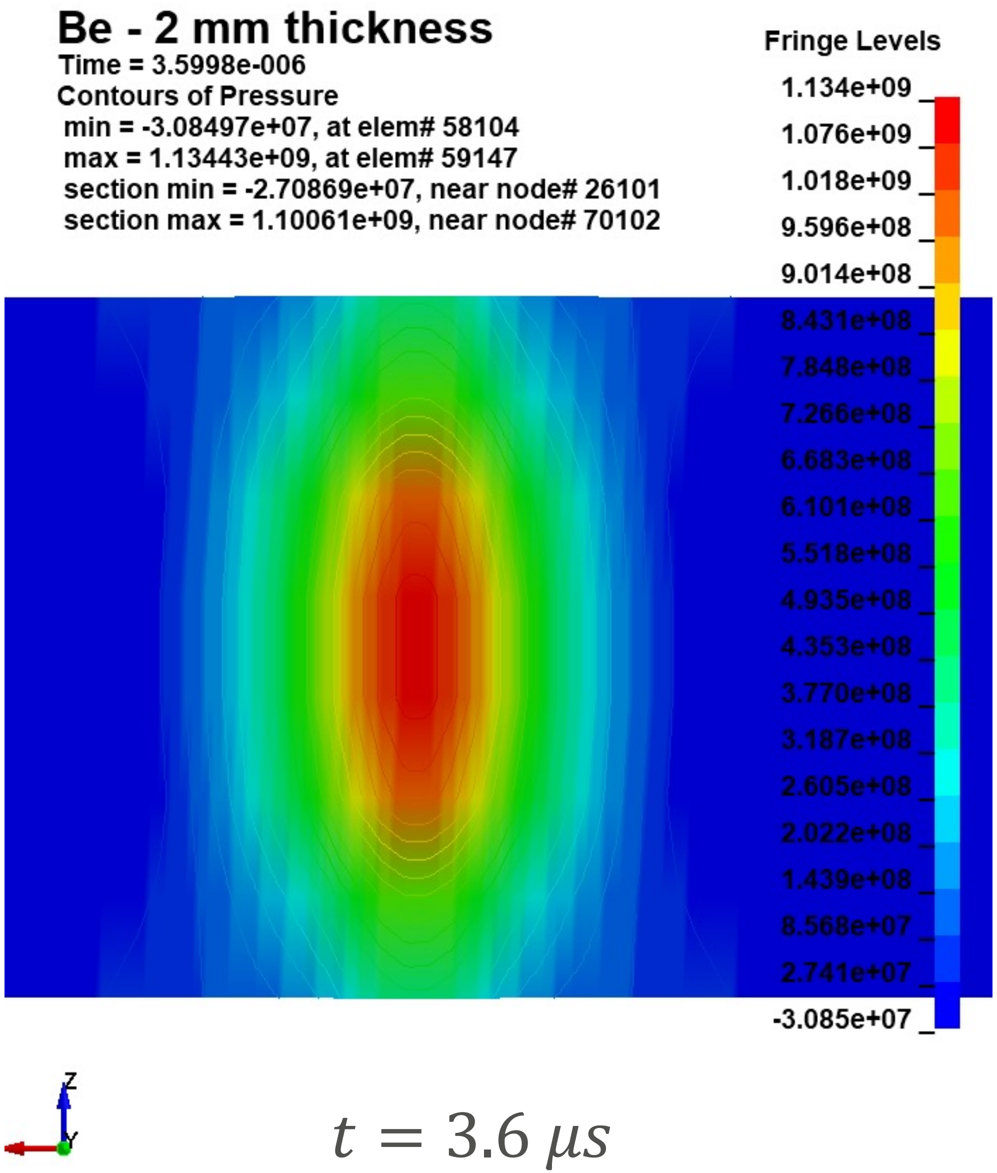
Johnson-Cook model has fixed density.

Large pressure due to beam heating.

Density changes with pressure and temperature.

User defined EOS will include density change

Equation of State (EOS) - Pressure



$t = 3.6\ \mu s$

Equation of State (EOS) - Model

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

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

$\eta = {}^{\rho}/_{\rho_0} = {}^{V_0}/_{V}$

ρ_0 = reference density

K_{T0} = initial bulk modulus

$K_T' = \frac{dK_T}{d\rho}$

*Gerald I. Kerley - **SANDIA REPORT - Equations of State for Be, Ni, W, and AU**

Equation of State (EOS) – User Subroutine

1. Calculate the **bulk modulus**

$$v_R = {}^{1}/_{\eta} = {}^{V}/_{V_0} = {}^{\rho_0}/_{\rho}$$

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

2. Update the **pressure**

$$P(v_R) = \frac{3}{2K_{T0}}\left(v_R^{-7/3} - v_R^{-5/3}\right)\left[1 + \frac{3}{4}(K_T' - 4)\left(v_R^{-2/3} - 1\right)\right]$$

and **internal energy** per unit reference volume:

$$E = e_M\,\rho_0 = \frac{9}{8K_{T0}}\left(v_R^{-2/3} - 1\right)^2\left[1 + \frac{1}{2}(K_T' - 4)\left(v_R^{-2/3} - 1\right)\right]$$

Equation of State (EOS) – User 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     example scalar user implementation of the gruneisen 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 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
```

Fermilab

Thanks for your attention