# LS-DYNA User-Defined Internal Ballistic Modeling

**Sirri Oğuz**

*Pyrotechnics System Manager – Commercial Crew Program*

NASA/JSC - Propulsion and Power Division, Houston, TX

*2024 International LS-DYNA Conference, Metro Detroit, Michigan, October 22-23, 2024*

# Executive Summary

- LS-DYNA Explicit Finite Element Analysis software and CADPROG Internal Ballistic Analysis code are integrated as a single modeling tool to analyze pyro-mechanical devices.

- Coupling these two codes allows the ballistic calculations to be performed without modeling the propellant in an FEA mesh. At each LS-DYNA time step, CADPROG is run analytically using the kinematic data (stroke, velocity, chamber volume) fed back from the LS-DYNA simulation. The calculated pressure is then applied to a piston or any applicable surface mesh in LS-DYNA model interactively.

- All external loads such as friction, damping loads, assisting/resisting loads, locking/unlocking loads are modeled in LS-DYNA and coupled with pressure loads from CADPROG to create a fully defined equation of motion

- Both codes are written in Fortran 77. CADPROG is integrated in the LS-DYNA source code as a user-defined subroutine and recompiled to generate a custom executable.

# LS-DYNA Overview

- LS-DYNA is a physics-based explicit FEA code that uses time integration to solve short duration transient and highly nonlinear dynamic problems
  - Extensive material model and Equation-of-State library, highly accurate contact algorithm and ability for Fluid-Structure Interaction (FSI) make LS-DYNA an excellent tool for evaluating complex mechanical and structural systems.

- Widely used for detonation problems using built-in JWL equation of state.

- While modeling the deflagration of a solid propellant with LS-DYNA is possible, it is very limited and mostly applicable to airbag systems that use gaseous nitrogen generated by burning of sodium azide.

- Allows users to define their own custom material models, equations of state and loading models through user-defined subroutines.

# CADPROG Overview

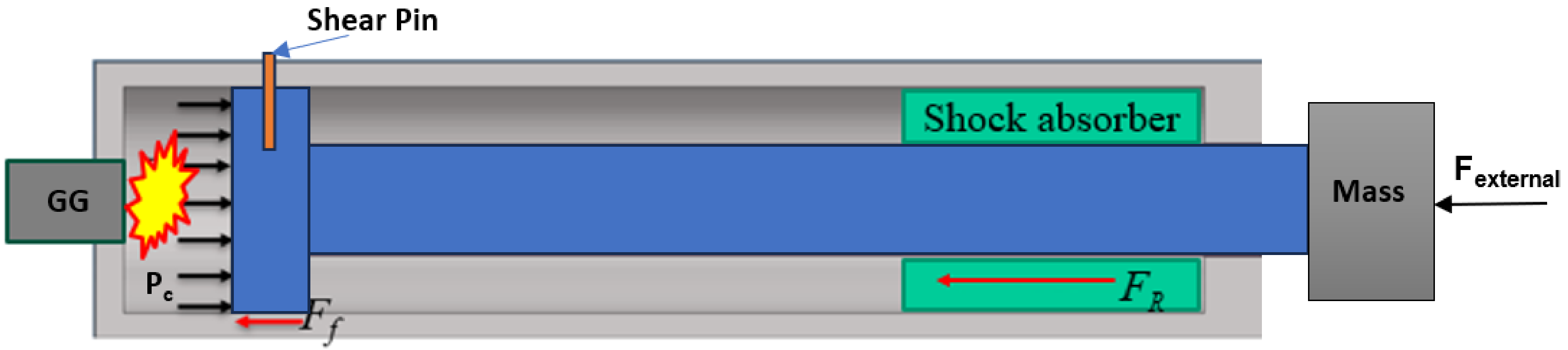
- CADPROG is a one-dimensional internal ballistic code developed by NSWC Indian Head Division and is widely used to analyze CAD/PAD devices and ejection seat catapults
  - Differential Equations of State and Energy are solved using 4$^{th}$ Order Runge-Kutta Method

- Capable of modeling solid propellants with many geometries, such as multi-perforated cylindrical grains, spherical ball powders, flakes, slabs.

- Single and Hi-Lo dual chamber CAD/PAD devices can be modeled with a mixture of up to three propellant types.

- NASA/JSC made various updates to the source code over the years to improve its capabilities:
  - *Improved heat-loss module for during and post-burnout heat loss*
  - *Added stroke-dependent external load capability*

# Internal Ballistics - Governing Equations




- *Conservation of Energy*

$$c_v \frac{d(m_p T)}{dt} = c_v T_0 \frac{dm_p}{dt} - A_P P_C \frac{dx}{dt} - \frac{d(E_{HL})}{dt}$$

$$\frac{dm_p}{dt} = \rho.S.r$$

- *Noble-Abel Equation of State*

$$r = BP_c^n$$

$$P_c[V_c + A_P x - \frac{C - m_p}{\rho} - \eta m_p] = m_p F \frac{T}{T_0}$$

- *Equation of Motion*

$$m\frac{d^2x}{dt^2} = P_c A_P - F_f - F_R - F_{ext}$$

With LS-DYNA/CADPROG feedback method, Acceleration is calculated by LS-DYNA using the CADPROG pressure and LS-DYNA Modeled Loads (friction, crush resistance, external aero, etc.)

# Creating LS-DYNA User-Defined Executable

- Fortran source code that contains a collection of subroutines is required to create a custom LS-DYNA executable.
  - The Fortran file dyn21.f can be obtained from ANSYS.
  - dyn21.f is modified by the user to integrate the user-defined loading, in this case the entire CADPROG code.
  - dyn21.f must be compiled by using the latest version of Intel® Visual Fortran Compiler.
    - Running the *makefile* supplied by LSTC compiles the file and links with the object files that are also provided.
    - The new LSDYNA.EXE file should be selected as the solver when running LS-DYNA.
  - Subroutine '*loadsetud*' receives TIME, TIMESTEP, elemental X and V input and initial conditions from the LS-DYNA run  and returns the calculated pressure curve, udL(i), interactively for each element "i" on selected pressure surface.

# User-Defined Loading Keyword Commands

- User-defined loading is invoked in LS-DYNA keyword file (input deck) with following keyword commands:

***USER_LOADING_SET**

| Card 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Variable | SID | LTYPE | LCID | CID | SF1 | SF2 | SF3 | IDULS |

ELEMENT SET id    Load Type "PRESSS"

***USER_LOADING**

| Card 1… | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Variable | PARM1 | PARM2 | PARM3 | PARM4 | PARM5 | PARM6 | PARM7 | PARM8 |

PARMi : CADPROG input parameters

# User-Defined Mods to dyna21.f

```
      subroutine loadsetud(time,lft,llt,crv,iduls,parm,nod,nnml)
# 803

c
c*******************************************************************
c|  Livermore Software Technology Corporation  (LSTC)            |
c|  ------------------------------------------------------------ |
c|  Copyright 1987-2008 Livermore Software Tech. Corp            |
c|  All rights reserved                                          |
c*******************************************************************
c
c     Input (not modifiable)
c       time : analysis time
c       x    : coordinate of node or element center
c       d    : displacement of node or lement center
c       v    : velocity of node or lement center
c       temp : temperature of node or element center
c       crv  : value of LCURV at time=time
c       iduls: id of user_loading_set
c       parm : parameters from user_loading
c       nod  : internal node numbers
c       nnml : offset for node block
c     Output (defined by user)
c       udl : user-defined load curve value
      include 'nlqparm'
      common/aux8loc/
     & x1(nlq),x2(nlq),x3(nlq),v1(nlq),
     & v2(nlq),v3(nlq),d1(nlq),d2(nlq),
     & d3(nlq),temp(nlq),udl(nlq),
     & xx11(nlq),xx21(nlq),xx31(nlq),
     & xx12(nlq),xx22(nlq),xx32(nlq),
     & xx13(nlq),xx23(nlq),xx33(nlq),
     & xx14(nlq),xx24(nlq),xx34(nlq),
     & xctr(nlq),yctr(nlq),zctr(nlq),
     & f(nlq), tr1(nlq), tr2(nlq), tr3(nlq)
c
c
C
C     S.OGUZ 5/17/2023
C
      dimension parm(*),nod(*)
      common/bk02/iburn,dt1,dt2,isdo
      COMMON /RUNDAT/ XX(8),OLDX(8),FF(8),X,P,T,C,V,ACCEL,
     +               PL,PLA,TH,TL,CH,CL,FC
       open(16, file='UD_load.out')
```

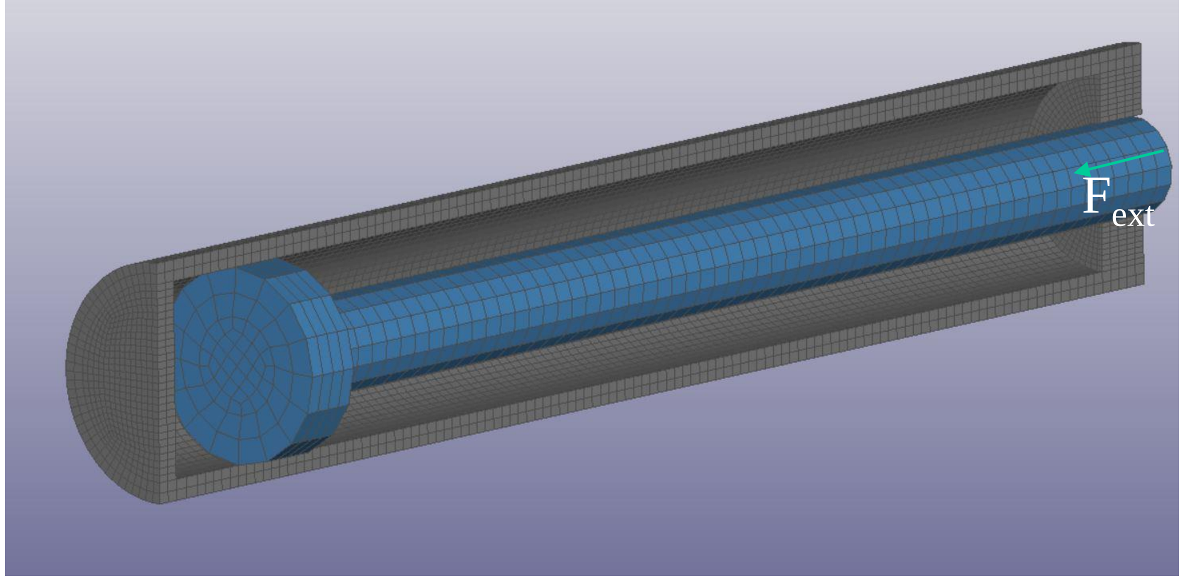
```
C
C        input to CADPROG:  DT1,TIME, D1(I),V1(I)
c        return from CADPROG:  PRESSURE
C%%%%%%%%%%%%    CADPROG PARAMETERS    %%%%%%%%%%%%%%%%
C  Fixed input parameters are read from CAGDROG subroutine
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      call CADPROG(parm)
      x=d1(1)
      V=v1(1)
C
C        CALL SUBROUTINE RUN  TO to start Runge-Kutta runs
c
       call RUN(DT1,TIME, ddd, vvv)
C
C   Assign calculated pressure to each element in the pressure segment set
C   Segment set is defined in *USER_LOADING_SET in keyword file
C
      do i=lft,llt
      udL(i)= P     ! P is the calculated pressure in CADPROG
      enddo
C
C     write a tabulated list of CADPROG results
C
      write(16,106) TIME,P,X,V,FC
  106 format(1X,F10.7,2x,F8.1,2x,F8.3,2X,F8.2, 2X,F9.3)
      continue
      return
      end
C
      subroutine cadprog(parm)
      .
      .....  read CADPROG parameters
      .
      .
      end
      return

      subroutine RUN(DT,TIME, ddd, vvv)
      .
      ....enter Runge-Kutta and all other CADPROG subroutines here
      .
      .
      end
      return
```

Pressure on each element

Output for plotting

CADPROG Subroutines

# Simplified Pyro Thruster Example




- Flexible Housing and Rigid Piston
- Piston pushes a 190 lbs mass
- FEA Model contains 26,256 solid elements
- Housing constrained at one end
- 10 inches of stroke

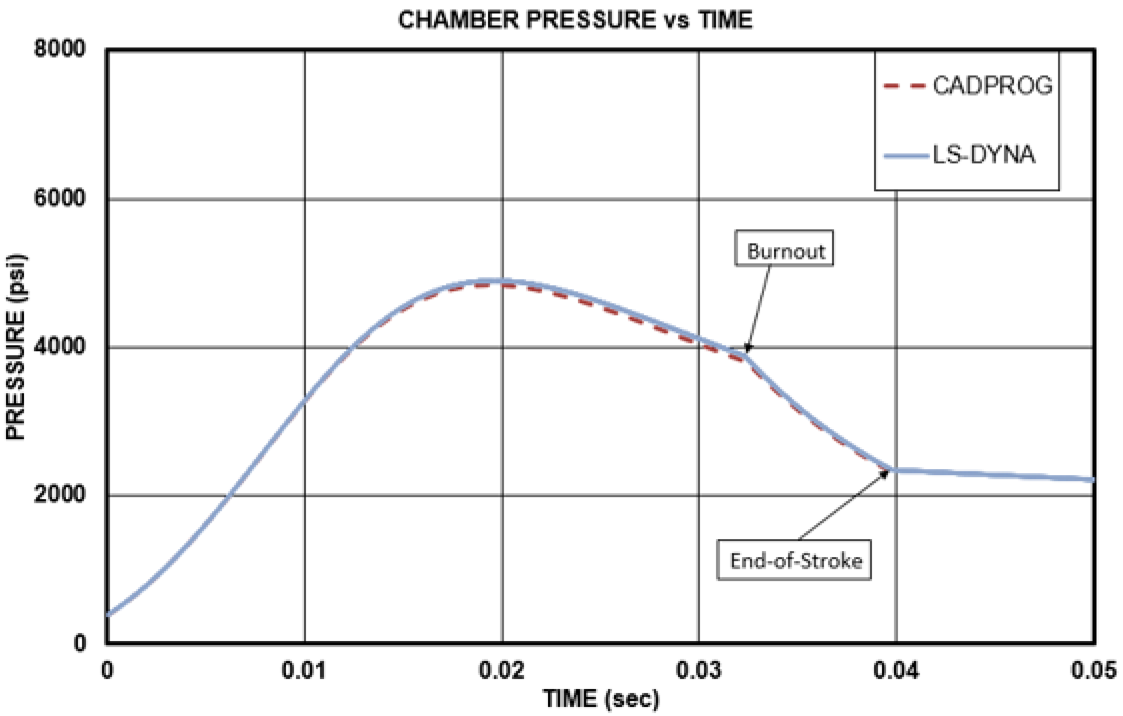
- *Propellant:* 7-perforated HES5808 grains
- *Igniter:* $BKNO_3$

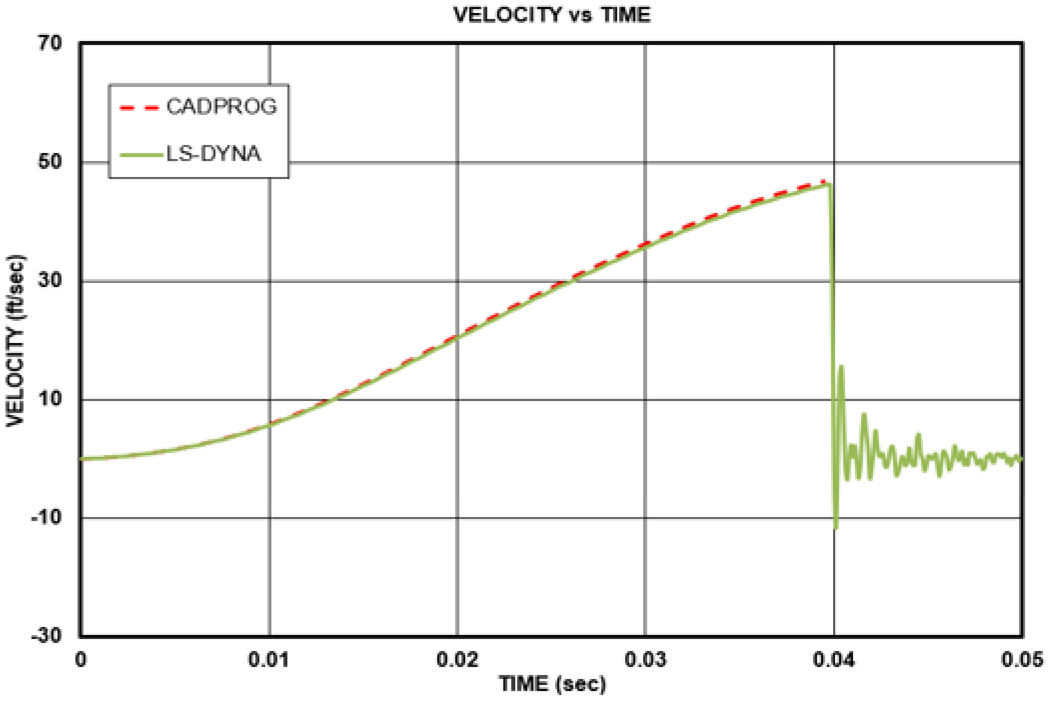
- Contact with static and dynamic friction coeff.
  *CONTACT_SURFACE_TO_SURFACE

- Various external assisting and resisting loads applied to piston end as nodal force.


| Grain Dimensions (in) | | | Density (gr/in$^3$) | Burn Rate (in/sec) | | Impetus (in-lb/gr) | $C_v$ (ft-lbf/lbm °K) |
|---|---|---|---|---|---|---|---|
| L | D | W | | B | N | | |
| 0.79 | 0.31 | 0.02 | 29.7 | 0.0032 | 0.76 | 8,730 | 574 |

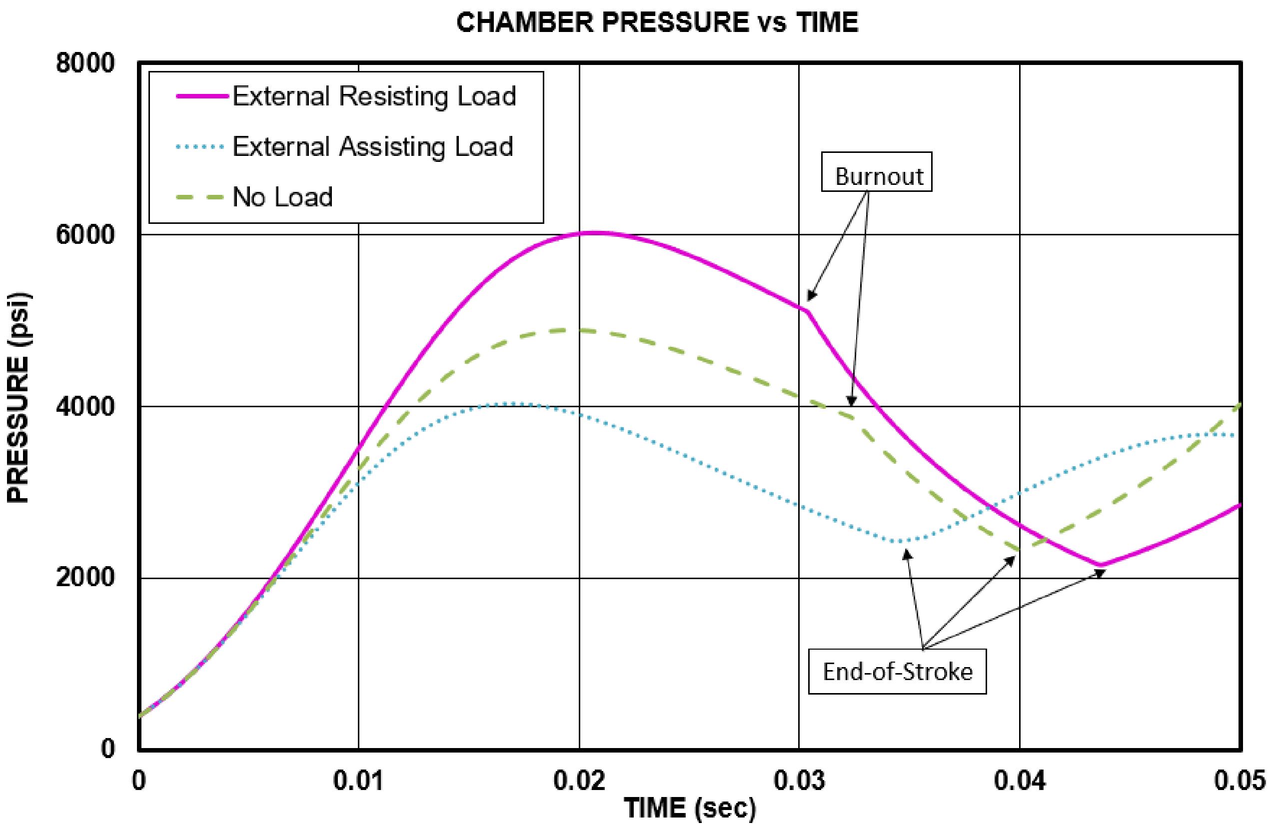
# LS-DYNA/CADPROG Baseline Run

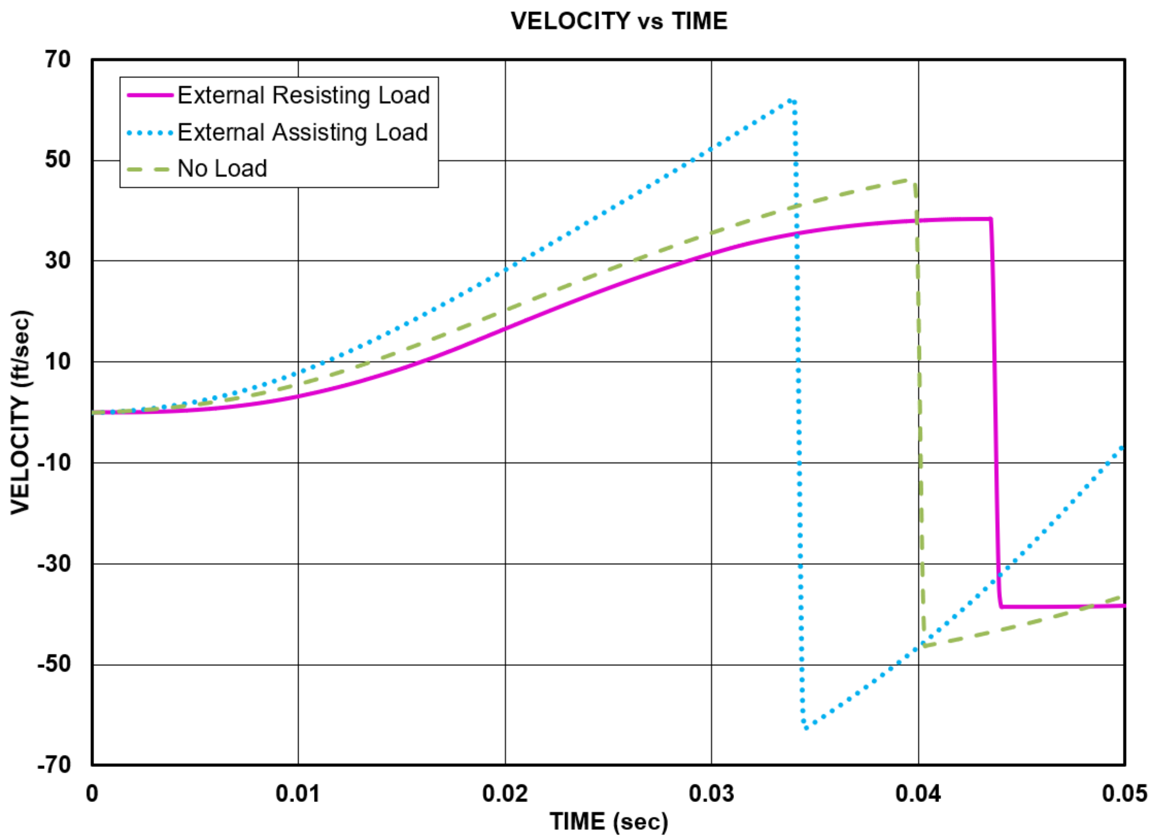



- As a baseline, a load case with zero external load was run to verify that the user-defined propellant burn subroutine is correctly implemented and yields the same pressure output as the original CADPROG code

# Pressure and Velocity Results for various load conditions

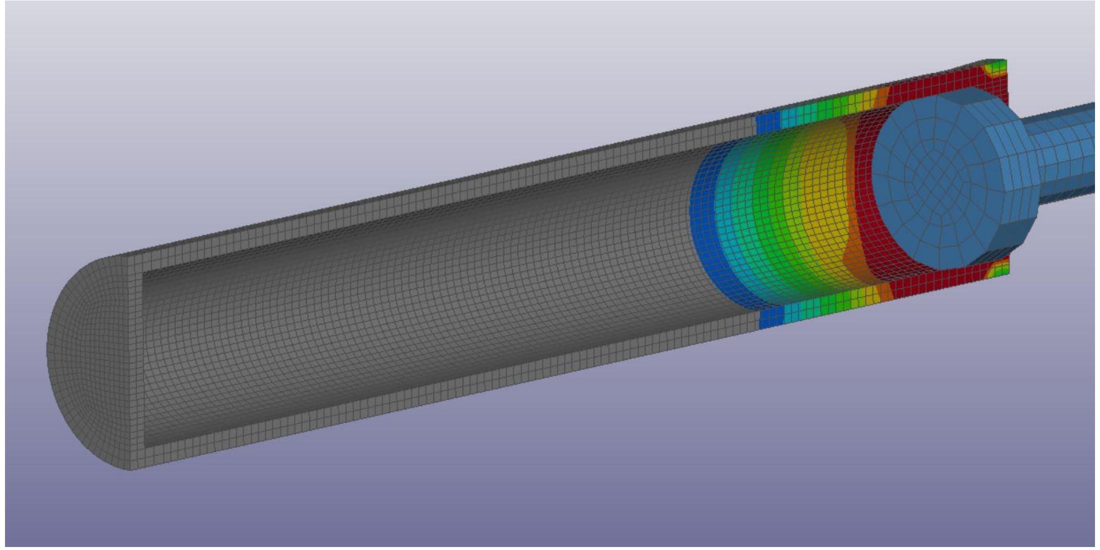


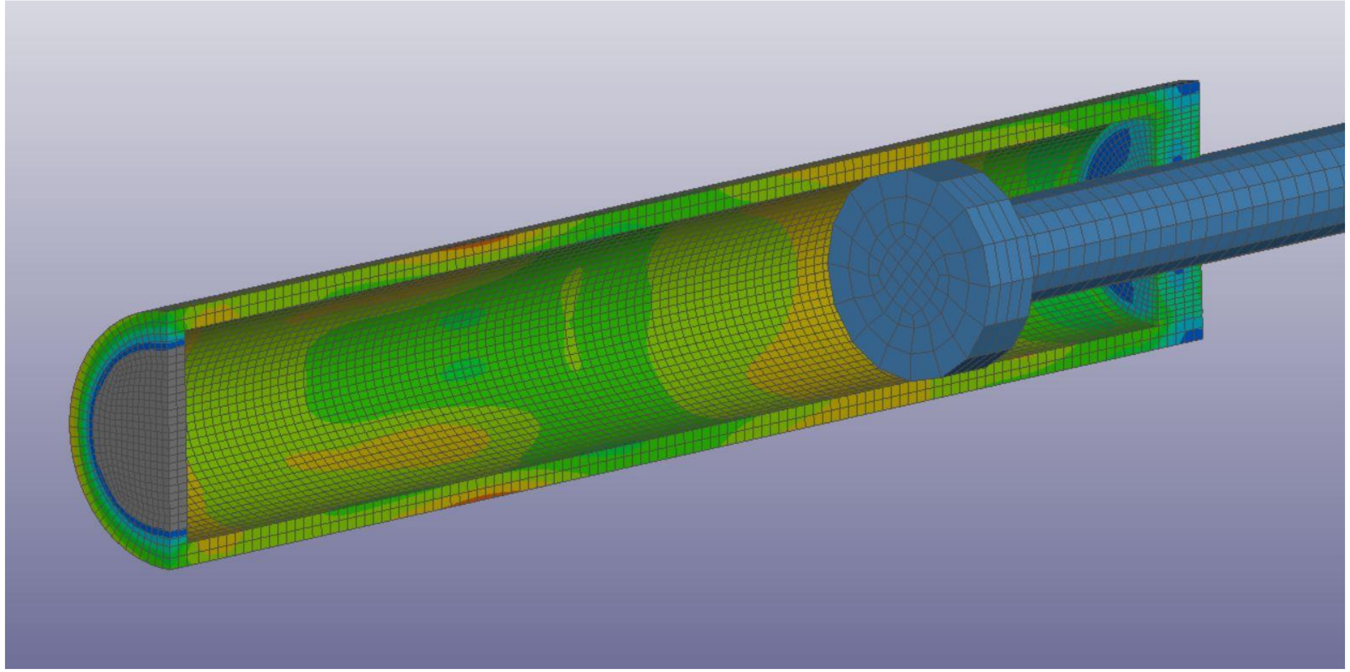

- **External resisting forces:** These forces decelerate the piston, causing the propellant to burn in a slower-expanding chamber, which subsequently increases in peak pressure.

- **External assisting forces:** These forces accelerates the piston, allowing the propellant to burn in a faster-expanding chamber, significantly reducing peak pressure.

# Post-Process Contours



End-of-Stroke Impact



Post impact bounce off

# Future Work

- In the current model, pressure is applied only to pre-selected meshed surfaces such as the piston area and initial chamber volume walls. A potential improvement would be to dynamically update the pressure application mesh surface (e.g., inner walls of the gas chamber) as the piston moves and the volume expands. This enhancement would allow for real-time full structural analysis during simulations.

- Integrating CADPROG as a user-defined pressure generator with LS-DYNA Conversion Element/Solution Element (CESE) solver could be a promising area for further research.
  - A user-defined equation of state with CADPROG within the CESE framework may address limitations of current EOS options, such as CESE_EOS_IDEAL_GAS or CESE_EOS_INFLATOR

- By using a custom EOS where CADPROG-generated pressures are mapped to the Euler-meshed chamber as initial conditions, a dual (high-low) chamber ballistic capability could be added. This would facilitate the modeling of gas flow from a high-pressure chamber to a low-pressure chamber using the CESE solver.

Questions?

# References

[1] Oguz, S and Salazar, F, "*Orion Parachute Riser Cutter Development*", 47th AIAA Joint Propulsion Conference, San Diego, CA, August 1-3, 2011.

[2] LS-DYNA Keyword User's Manual, Version R13, Livermore Software Technology Corporation.

[3] Holter, W.H., *Simplified Solutions to the Interior Ballistic Problems of Cartridge Actuated Devices*, U.S. Naval Weapons Laboratory, Report No. 1752, May 1962.

[4] Adoum, M., "*Examples Manual for *USER_LOADING Option*", 4th European LS-DYNA Users Conference

[5] Paul, B, *CAD Ballistics Seminar Notes*, OEA Aerospace Inc., 1999

# Nomenclature

$A_p$=piston area
$B$= burn rate coefficient
$C$=propellant mass
$c_v$=specific heat at constant volume
$E_{HL}$=heat loss
$F$=impetus
$F_{EXT}$=external force
$F_f$ =friction force
$m$=imparted mass
$m_p$=mass of propellant gas produced
$n$=burn rate exponent
$P_C$=chamber pressure
$r$=propellant burn rate
$S$=propellant surface area
$T$=gas temperature
$T_o$=isochoric flame temperature
$V_C$=initial chamber volume
$x$=stroke
$V$=velocity
$\rho$=propellant mass density
$\eta$=co-volume

CAD/PAD = Cartridge/Propellant Actuated Device

JWL = Johns-Wilkins-Lee Equation of State