MEGA User's Manual

Michael Hillman[*] Kuan-Chung Lin[†] Saili Yang[‡] Guohua Zhou[§]

November 4, 2019

This user manual provides basic guidance for users on how to use the Meshfree Explicit Galerkin Analysis (MEGA) program.

1 Introduction

Meshfree methods are a numerical methodology which rely on only a set of nodes and surfaces to solve the problem, and in contrast to finite element methods, no element topology is required whatsoever.

The input of the program is divided into two aspects: (1) *the model*: geometry of the problem, including nodal discretization, definition of sets of nodes, and surfaces, and (2) *the control file*: specification of boundary conditions, initial conditions, material properties, and other simulation parameters.

Correspondingly, two files are required for running MEGA. The first file provides the geometric definition of the problem along with identification of sets of nodes, and has an LS-DYNA input file format and is stored as '**.k*' (* here indicates a wildcard, i.e., whatever you want the file to be called). Information on the exact format of this file is readily available online, but can be generated with a variety of preprocessors.

The second file is a text file (herein called the 'control file') named '*Control.dat*', which gives all the rest of the information needed in order to run a numerical model, such as specification of boundary conditions, material properties, total simulation time, and so on. These parameters are specified in terms of node sets, unless they are global parameters such as output frequency and simulation time.

2 Basics

We have the following basic definitions based on the decomposition of the problem into two aspects:

The model The problem geometry and node sets, contained in the '**.k*' file.

Node sets Part of the model. Sets of nodes specified in the preprocessor, contained in the '**.k*' file.

[*] Kimball Assistant Professor, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802, mhillman@psu.edu

[†] Graduate Student Researcher, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802

[‡] Graduate Student Researcher, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802

[§] Postdoctoral Scholar, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802

The control file The control file contains all other information needed for the simulation which is not specified in the model: (1) assigned values of material parameters, boundary conditions, etc., to node sets, and (2) global parameters such as simulation time, output frequency, output variables, and so on.

3 The model

The geometry of the problem is described by a LS-DYNA **.k* file. In this manual it is assumed that it is generated by the software *CUBIT* or *TRELIS* unless otherwise specified. The specific format of the '**.k*' file is given in the manuals of LS-DYNA. There is no restriction on naming the '**.k*' file, but the name needs to be specified in the MEGA control file, '*Control.dat*' to inform the MEGA code to read the geometric information from the given '**.k*' file.

The node sets are assigned when building the model. See the Numerical Examples section for examples on how to do so.

4 Control file

The control file gives the problem statement, aside from geometry. A *card* system is utilized to specify initial conditions, boundary conditions, material properties, and so on.

Each *card* begins with an asterisk '*' and is followed directly by the name of the card. Only if the input card names matches exactly with the ones defined in MEGA code including the spaces between the words, the card will be read effectively to MEGA, otherwise the card is not read. The input is FORTRAN free format, however the actual FORTRAN types are given below. Warnings are given in the case that the information after the card cannot be read or is not properly specified.

The cards are described in the following subsections. Each card starts with the definition of the card, followed by its explanation, separated by a short straight line '————'. The 'Comment line' in the card can be any characters within one line to describe or name some data since it is not read by MEGA.

4.1 Set names

This section describes a special kind of card where you can give a names to the node sets specified in *CUBIT* or *TRELIS*. This is for convenience of the user and readability of the input file. Other cards which assign nodal information can either use the set number (specified in the preprocessor), or set name.

- ∗SET NAME

  Comment line

  SET_NO

  Comment line

  SET_NAME

  Comment line

  ————

  SET_NO: Integer
  value = Nodeset number specified in *CUBIT* or *TRELIS*.

  SET_NAME : Character
  value = the name to give this node set

Example To give node set 2 the name 'SOIL', the entire card would read as:
*SET NAME
#SET NO (in CUBIT or TRELIS)
2
#NAME OF THIS SET
SOIL

4.2 Mandatory Cards for Global Parameters

This section describes mandatory cards for global parameters. If these cards are not specified, then the program will exit. The log file will describe why the program exited (which card was not given or read incorrectly).

- ∗MODEL

  Comment line

  ls-dyna file

  ————

  ls-dyna file: Character

  value = name of the geometry LS-DYNA *.k file to be read in by MEGA.

  Example To tell MEGA to use the file 'SoilModel.k' for the model definition, the entire card would read as follows:
  *MODEL
  # Name
  SoilModel.k

- TIME PARAMETERS

  Comment line

  TIME_END

  Comment line

  IAUTO_TS

  Comment line

  TIME STEP

  ————

  TIME_END: Double precision
  value = Simulation end time.

  IAUTO_TS : Integer
  0 = user-specified fixed time step
  1 = automatically selected time step size, estimated in MEGA code

  TIME STEP: Double precision
  if IAUTO_TS = 0, value = time step
  if IAUTO_TS = 1, value = safety scaling factor for automatic time step, i.e., what fraction of the estimated critical time step to use for the simulation.

Example To have a total simulation time of 30 seconds, with a user-defined time step of 2.0E-6 the entire card would read:

```
*TIME PARAMETERS
#TIME_END
30.0
#IAUTO_TS
0
#TIME STEP IF IAUTO_TS = 0 / FACTOR FOR TIME STEP IF IAUTO_TS = 1
2.0E-6
```

Example To have a total simulation time of 10 seconds, with an automatically selected time step with a safety factor of 0.9 the entire card would read:

```
*TIME PARAMETERS
#TIME_END
10.0
#IAUTO_TS
1
#TIME STEP IF IAUTO_TS = 0 / FACTOR FOR TIME STEP IF IAUTO_TS = 1
0.9
```

- *OUTPUT PARAMETERS

  TIME_OUTPUT

  Comment line

  OUTPUT VARIABLES

  TIME_OUTPUT: Double precision
  value = output time frequency

  OUTPUT VARIABLES: A column of character
  values = the names of variables to output

  The *OUTPUT VARIABLES* is a list containing of any of the following:

  node_id: output id number of each node.
  nodal_volume: output volume value of each node.
  nodal_mass: output mass of each node.
  bodyid: output body id number of each node.
  material_type: output material type of each node.
  Poissons_ratio: output Poisson's ration of each node.
  Youngs_modulus: output Young's modulus of each node.
  density: output density of each node.
  displacement: output displacements of each node.
  pre_disp_force: output reaction forces of each node.
  stress: output stresses of each node.
  velocity: output velocities of each node.
  acceleration: output accelerations of each node.
  fint: output internal force of each node.
  fixity: output the boundary condition (fixity) of each node.

initial_coordinate: output initial coordinates of each node.
init_velocity: output initial velocities of each node.
physical_window: output support size of each node.
normalized_window: output normalized support size of each node.
characteristic_distance: output the local characteristic nodal spacing for critical time step estimation.
max_wave_vel: output the local max wave velocity for critical time step estimation.
IJKspace:output the IJK space of the node (in the bin search algorithm).

The following are also defined for specifying general output parameters:

basic: output node_id, displacement, velocity and bodyid for each node.
all: output all the variables that are available to output.
unformatted: unformatted FORTRAN output (slightly faster output, but less readable by human)

Example Output at the frequency of 0.033 seconds, just the displacement and plastic strain:

```
*OUTPUT PARAMETERS
# TIME_OUTPUT
0.033
# OUTPUT VARIABLES
displacement
eps
```

Example Output at the frequency of 2.0E-4 seconds, the basic variables as well as eps, in unformatted output:

```
*OUTPUT PARAMETERS
# TIME_OUTPUT
2.0E-4
# OUTPUT VARIABLES
basic
eps
unformatted
```

4.3 Material parameters

This section describes the card for assigning material parameters for node sets. If these cards are not specified, or if the specification is inconsistent (e.g., assigning two different materials to the same node, or only partial specification of material properties) then the program will exit. The log file will describe why the program exited.

- ∗MATERIALS

  Comment line

  SET NO or *NODE SET NAME*

  Comment line

  TYPE

Comment line

PROPS

SET NO or *NODE SET NAME*: Integer or character

value = the node set number specified in the *.k file by *CUBIT* or *TRELIS*, or name of node set specified in the SET NAME card.

TYPE: Integer
1 = Elasticity
2 = J2 Plasticity
3 = Drucker Prager

PROPS: Double precision (25)
value = depends on TYPE

TYPE 1 : Elasticity
$PROPS(1)$ = Poisson ratio, υ
$PROPS(2)$ = Young's Modulus, E
$PROPS(3)$ = Density, ρ
$PROPS(21)$ = Mass proportional damping coefficient, α

TYPE 2 : J2 Plasticity
$PROPS(1)$ = Poisson ratio, υ
$PROPS(2)$ = Young's Modulus, E
$PROPS(3)$ = Density, ρ
$PROPS(4)$ = Initial yield stress
$PROPS(5)$ = B (parameter in hardening rule)
$PROPS(6)$ = CE (parameter in hardening rule)
$PROPS(21)$ = Mass proportional damping coefficient, α

B and CE are the parameters for the exponential hardening rule with yield function: $f(\sigma, \overline{e}^p) = \overline{\eta} - \sqrt{2/3} \times Y \times (1 + B\overline{e}^p)^{CE}$ where $\overline{\eta}$ is the tensor norm of the deviatoric portion of the Cauchy stress, and $\overline{e}^p$ is the effective plastic strain. For example, B = 0 and CE = 0 give perfect plasticity.

TYPE 3 : Drucker-Prager with tension cut-off and isotropic scalar damage
$PROPS(1)$ = Poisson ratio, υ
$PROPS(2)$ = Young's Modulus, E
$PROPS(3)$ = Density, ρ
$PROPS(4)$ = Type of Drucker Prager yield surface (DP-TYPE) in relation to soil parameters.
$PROPS(7)$ = The angle of dilatancy, ψ (not currently used)
$PROPS(8)$ = Tension cut-off value
The tension cut-off provides another yields surface, which limits the amount of tension the material can sustain.

$PROPS(9)$ = Damage parameter 1 (damage initial threshold value) k_i
$PROPS(10)$ = Damage parameter 2 (damage accumulating rate) k_c
$PROPS(11)$ = Maximum permissible damage d_f
$PROPS(21)$ = Mass proportional damping coefficient, α

The Cauchy stress is obtained via the scalar damage law $\sigma = (1 - d) \times \overline{\sigma}$, where σ is the Cauchy stress, and $\overline{\sigma}$ is the stress calculated from plasticity. The damage evolution function is $d = k_c(\eta - k_i)/(\eta(k_c - k_i))$, where η is the strain tensor norm and $k_c > \eta > k_i$. The value of d is limited to the maximum permissible damage d_f. Typically $d_f = 0.95$, and is set to a finite value for numerical stability reasons.

The shear yield function for Drucker-Prager is defined as $f(\sigma, \overline{e}^p) = \overline{\eta} - p \times q_\phi - K_\phi$ where $\overline{\eta}$ is the tensor norm of the deviatoric portion of the Cauchy stress, p is the pressure, and the parameters K_ϕ and q_ϕ are calculated based on the angle of friction ϕ and cohesion c as follows, or are directly input:

If = DP-TYPE 1, the yield surface completely circumscribes the Mohr-Coulomb yield surface

$$K_\phi(c, \phi) = \frac{6c\cos(\phi)}{\sqrt{3/2}(3-\sin(\phi))},\ q_\phi(\phi) = \frac{6\sin(\phi)}{\sqrt{3/2}(3-\sin(\phi))}$$

$PROPS(5)$ = The angle of initial friction, ϕ
$PROPS(6)$ = Cohesion, c

If = DP-TYPE 2, the yield surface circumscribes the middle of the Mohr-Coulomb yield surface

$$K_\phi(c, \phi) = \frac{6c\cos(\phi)}{\sqrt{3/2}(3+\sin(\phi))},\ q_\phi(\phi) = \frac{6\sin(\phi)}{\sqrt{3/2}(3+\sin(\phi))}$$

$PROPS(5)$ = The angle of initial friction, ϕ
$PROPS(6)$ = Cohesion, c

If = DP-TYPE 3, the yield surface inscribes the Mohr-Coulomb yield surface

$$K_\phi(c, \phi) = \frac{3c\cos(\phi)}{\sqrt{(9+\sin^2(\phi))}},\ q_\phi(\phi) = \frac{\sin(\phi)}{\sqrt{(9+\sin^2(\phi))}}$$

$PROPS(5)$ = The angle of initial friction, ϕ
$PROPS(6)$ = Cohesion, c

If = DP-TYPE 4, directly input values the values of q_ϕ in $PROPS(5)$ and K_ϕ in $PROPS(6)$

$PROPS(5) = q_\phi$
$PROPS(6) = K_\phi$

TYPE 6 : J2 Plasticity with isotropic damage
$PROPS(1)$ = Poisson ratio, υ
$PROPS(2)$ = Young's Modulus, E
$PROPS(3)$ = Density, ρ
$PROPS(4)$ = Initial yield stress

$PROPS(5)$ = B (parameter in hardening rule)
$PROPS(6)$ = CE (parameter in hardening rule)
$PROPS(7)$ = not used
$PROPS(8)$ = not used
$PROPS(9)$ = Damage parameter 1 (damage initial threshold value) k_i
$PROPS(10)$ = Damage parameter 2 (damage accumulating rate) k_c
$PROPS(11)$ = Maximum permissible damage d_f
$PROPS(21)$ = Mass proportional damping coefficient, α

B and CE are the parameters for the exponential hardening rule with yield function: $f(\sigma, \overline{e}^p) = \overline{\eta} - \sqrt{2/3} \times Y \times (1 + B\overline{e}^p)^{CE}$ where $\overline{\eta}$ is the tensor norm of the deviatoric portion of the Cauchy stress, and $\overline{e}^p$ is the effective plastic strain. For example, B = 0 and CE = 0 give perfect plasticity

Example Assign the node set named 'BAR' as a plastic material (type 2), with Poisson's ratio 0.3, Young's modulus 78.2E9, density 2700, initial yield radius 0.29E9, hardening parameter B=125, and hardening parameter CE = 0.1

```
*MATERIALS
# SET NO. OR NODE SET NAME
BAR
# TYPE
2
# props
0.3 78.2E9 2700 0.29E9 125
0.1 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

4.4 Optional Cards

This section describes optional cards. Generally, these will be specified, although if they are omitted, the default values are assigned which can be convenient for certain problem parameters. Default values are given in this section.

- **BODY DEFINITION*

  Comment line

  SET NO or *NODE SET NAME*

  Comment line

  BODY ID

  Comment line

  BODYFORCE

  SET NO or *NODE SET NAME*: Integer or character

  value = the node set number specified in the *.k file by *CUBIT* or *TRELIS*, or name of node set specified in the SET NAME card.

BODY ID: Integer

value = value of body ID for MEGA to assign to this node set number. Bodies with different IDs will interact using contact algorithms. Otherwise, they will be 'glued' together.

default value = 1 for all bodies

BODYFORCE: Double precision(3)

value = value of body force in x, y, z direction

default value =no body force for any body

Example To give the node set named 'BAR' a body ID of 1, and give it a gravity force in the y direction of -9.81, the following card would be used:

```
*BODY DEFINITION
#SET NO. OR NODE SET NAME
BAR
#BODY ID
1
#BODYFORCE
0 -9.81 0
```

- ∗FEM OUTPUT

  Comment line

  IFEM_OUTPUT

  IFEM_OUTPUT: integer (0 or 1)
  value = 0 for not using mesh output;
  value = 1 for using mesh output;
  default value = no mesh output.

- ∗BOUNDARY CONDITION (*repeatable card*)

  Comment line

  SET NO or *NODE SET NAME*

  Comment line

  FIXITY

  Comment line

  STEPS

  Comment line

  DISPLACEMENTS

  Comment line

  DURATIONS

  SET NO or *NODE SET NAME*: Integer or character

  value = the node set number specified in the *.k file by *CUBIT* or *TRELIS*, or name of node set specified in the SET NAME card.

FIXITY: Integer(3)

values = i, j, k for boundary condition in x, y, z direction at the node set by *SET NO* or node set by *NODE SET NAME*:
0 = free condition.
1 = fixed with zero displacement.
2 = non-zero displacement. The displacement will be prescribed linearly in time according to the loading steps.
default = free condition on all nodes.
STEPS: Integer

value = number of loading steps

Note that this value does not need to be specified unless non-zero displacements are indicated in the variable *FIXITY*.

DISPLACEMENTS: Double precision(3)

values = non-zero displacements. Columns (three): displacement in x, y, z. Rows (number of loading steps): values to apply sequentially.

Note that this value does not need to be specified unless non-zero displacements are indicated in the variable *FIXITY*.

DURATIONS: Double precision(3)

values = time durations to apply non-zero displacements. Columns (two): start and end time for steps. Rows (number of loading steps): time durations to apply sequentially.

Note that this value does not need to be specified unless non-zero displacements are indicated in the variable *FIXITY*.

Example Fix the node set named 'WALL' in the z direction only, with x and y free

```
*BOUNDARY CONDITION
# SET NO. OR NODE SET NAME
WALL
#FIXITY
0 0 1
```

Example Prescribe to node set named 'BLADE', two steps of non-zero values 0.6 0.0 -0.6 over time $t = 0s$ to $t = 0.5s$, then 3.0 0.0 0.0 over time $t = 0.5s$ to $t = 3.5s$:

```
*BOUNDARY CONDITION
# SET NO. OR NODE SET NAME
BLADE
#FIXITY
2 1 2
#STEPS
2
#DISPLACEMENT (X Y Z)
0.6 0.0 -0.6
3.0 0.0 0.0
#TIME DURATION (T1 T2)
0.0 0.5
0.5 3.5
```

Note that *multiple definitions of boundary conditions for a single node will result in undefined behavior!* This motivates the introduction of the next card.

- ∗ORDERED BOUNDARY CONDITION (*repeatable card*)

  Comment line

  SET NO or *NODE SET NAME*

  Comment line

  FIXITY

  Comment line

  STEPS

  Comment line

  DISPLACEMENTS

  Comment line

  DURATIONS

  This card is the same as '*BOUNDARY CONDITION', but '*ORDERED BOUNDARY CONDITION' means that the boundary condition defined in this card overwrites the values of any of the common nodes in the previously defined ones in the control file. Thus the *order* specified top to bottom of boundary conditions cards in the control file matters if this card is used.

  Example Let's suppose we have defined two node sets. One contains the entire domain "ALL_NODES", another contains some nodes we want completely fixed "FIXED_NODES". In addition to the fixed nodes, lets say all other nodes should have rollers in the Y and Z directions. In order to avoid defining boundary conditions twice for a node which would result in undefined behavior, one can apply them sequentially using this card. In this example, the following cards would be used to first apply the roller condition to all nodes, then overwrite that condition for the nodes we want fixed:
  *ORDERED BOUNDARY CONDITION
  # SET NO. OR NODE SET NAME
  ALL_NODES
  #FIXITY
  0 1 1

  *ORDERED BOUNDARY CONDITION
  # SET NO. OR NODE SET NAME
  FIXED_NODES
  #FIXITY
  1 1 1

- ∗INITIAL CONDITION

  Comment line

  SET NO or *NODE SET NAME*

  Comment line

  VEL

  SET NO or *NODE SET NAME*: Integer or character

value = the node set number specified in the *.k file by *CUBIT* or *TRELIS*, or name of node set specified in the SET NAME card.

VEL: Double precision(3)
value = the initial velocity vector in x, y, z direction (default = 0 for all directions)

Example Give the node set 'WHOLE_BAR' an initial condition of -0.1 in the x direction only:
*INITIAL CONDITION
#SET NO. OR NODE SET NAME
WHOLE_BAR
#VEL
-0.1 0 0

4.5 Advanced Cards

The use of these cards is for expert users or to override default values for specific reasons. It is not recommended to use these cards unless you know how they will effect the simulation. These parameters have defaults for ease of use of the code and for obtaining the best simulation results.

- *WINDOW DEFINITION

  Comment line

  SET NO or *NODE SET NAME*

  Comment line

  WINDOW

  SET NO or *NODE SET NAME*: Integer or character

  value = the node set number specified in the *.k file by *CUBIT* or *TRELIS*, or name of node set specified in the SET NAME card.

  WINDOW: Double precision
  value = the normalized support size for the RK approximation
  default = 1.7

  Example Give the node sets named 'WHOLE_BAR' a normalized window of 1.3:
  *WINDOW DEFINITION
  #SET NO. OR NODE SET NAME
  WHOLE_BAR
  # WINDOW
  1.3

- *SUPPORT

  Comment line

  SPHERICAL

Example value = 1 use spherical support

value = else, use box support

- ∗RK PARAMETERS

  Comment line

  RK_DEGREE, *RK_CONT*, *RK_IMPL*

  ————

  RK_DEGREE: Integer
  value = degree of complete monomials that the RK approximation can reproduce
  default = 1

  RK_CONT: Integer
  value = k, where the the continuity of the RK approximation is selected as C^k
  default = 3

  RK_IMPL: Integer
  1 = Implicit gradients in RK approximation
  0 = Direct gradients in RK approximation
  default = 0

  Example Use linear basis, C^2 continuity, and implicit gradients
  *RK PARAMETERS
  # *RK_DEGREE*, *RK_CONT*, *RK_IMPL*
  1 2 1

- ∗QL PARAMETERS

  Comment line

  QL_USE, *QL_COEFFICIENT*

  ————

  QL_USE: Integer
  1= Quasi-linear RK approximation is used
  0= Quasi-linear RK approximation is not used
  default = 1

  QL_COEFFICIENT: Double precision value = Quasi-linear coefficient
  default = 0.001

  Example Do not use quasi-linear approximation
  *QL PARAMETERS
  # USE, COEFFICIENT
  0 0.001

- ∗HPC PARAMETERS

  Comment line

  HPC_SCHEME

———

HPC_SCHEME: Integer
1=parallel computing is used
0=parallel computing is not used
default = 1

Example Use parallel computing
*HPC PARAMETERS
# *HPC_SCHEME*
1

- *GHOSTING PARAMETERS

  Comment line

  GHOST_BUFFER

  ———

  GHOST_BUFFER: Integer
  value = number of layers of ghost nodes to use for domain decomposition
  default = 1

  Example Use one layer of ghost nodes:
  *GHOSTING PARAMETERS
  # *GHOST_BUFFER*
  1

- *INTEGRATION PARAMETERS

  Comment line

  TYPE

  ———

  TYPE: Integer
  0 = Direct nodal integration (DNI)
  1 = Stabilized non-conforming nodal integration (SNNI)
  2 = Naturally stabilized nodal integration (NSNI)
  default = 2

  Example Use SNNI integration in the simulation
  *INTEGRATION PARAMETERS
  # *TYPE*
  1

- *FORMULATION PARAMETERS

  Comment line

  IFINITE_STRAIN, *ILAGRANGIAN*

IFINITE_STRAIN: Integer
0 = small deformation
1 = finite deformation
default = 1
ILAGRANGIAN: Integer
0 = semi-Lagrangian RKPM method
1 = Lagrangian RKPM method
default = 0

Example Use finite strain formulation with the Lagrangian method
*FORMULATION PARAMETERS
# *IFINITE_STRAIN*, *ILAGRANGIAN*
1 1

- ∗STABILIZATION CONTROL

  Comment line

  IUSE_STABILIZATION_CONTROL

  Comment line

  STABILIZATION_CONTROL_COEFFICIENT

  IUSE_STABILIZATION_CONTROL: Integer
  0 = do not modify stabilization
  1 = control the contribution of stabilization; will limit the contribution to internal force
  default = 0
  STABILIZATION_CONTROL_COEFFICIENT: Double precision
  value = factor in front of re-normalization of stabilization contribution (1.0=same magnitude)
  default = n/a

  Example use stabilization control, do not let stabilization have a bigger contribution than 0.1 times the magnitude of the internal force:
  *STABILIZATION CONTROL
  # *IUSE_STABILIZATION_CONTROL*
  1
  # *STABILIZATION_CONTROL_COEFFICIENT*
  0.1

- ∗*PERIODIC SEARCH
  Comment line

  TIME

  TIME: Double precision

Example Use periodic search in simulation, search nodes' neighbors after every 0.01s

```
*PERIODIC SEARCH
#SEARCH_PERIOD
0.01
```

5 Numerical examples

5.1 One-dimensional elastic wave propagation

5.1.1 Problem description

A one-dimensional (1D), elastic bar impacting a rigid wall is modeled. The length is $L = 5m$. The material is linear elastic with the given properties of density $\rho = 100kg/m^3$, Young's modulus $E = 1$ MPa, and Poission's ratio $v = 0.0$. The initial velocity for the whole bar is 0.1 m/sec. The simulated displacement history at the free end point by RKPM in MEGA code with a Lagrangian option is given in Figure 12 (blue circle) and compared with the exact solution (black solid line). As can be seen, the numerical result matches well with exact solution.

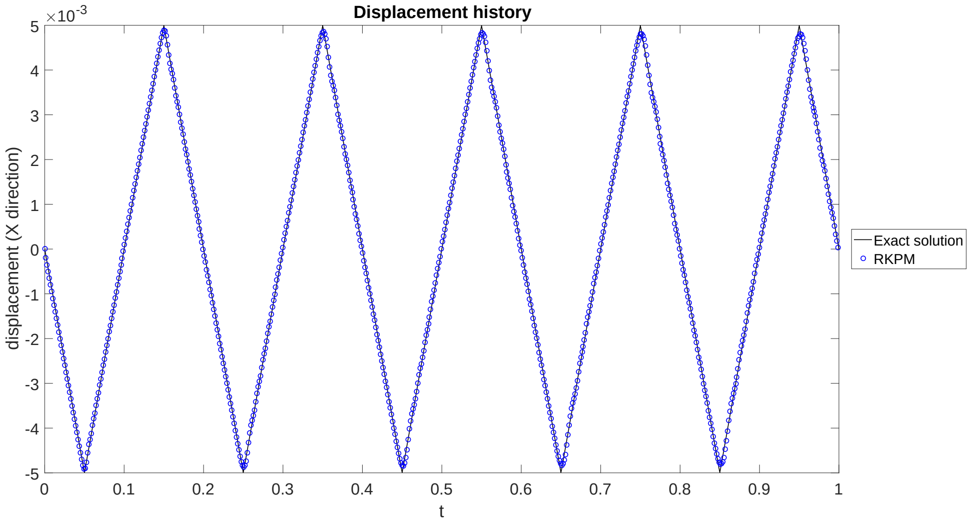
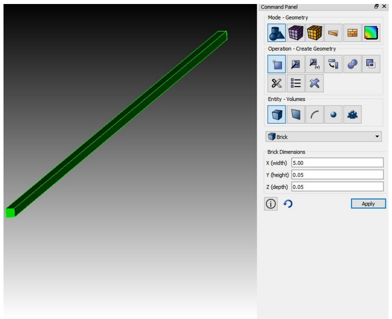


Figure 1: The displacement history at the free end point of 1D wave propagation problem

5.1.2 Input files for one-dimensional wave propagation

I Process of creating the model in *CUBIT* or *TRELIS*

Step 1 Creating the Brick



Brick

Brick Dimensions
X (width) 5.00
Y (height) 0.05
Z (depth) 0.05

(1) Click **Volumes**

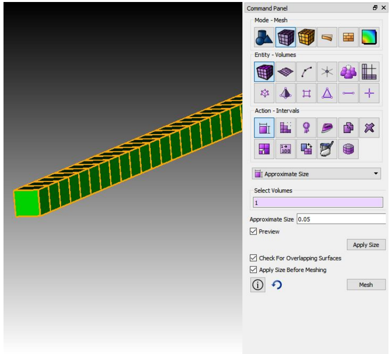
(2) Click **Create**

(3) Select the drop-down menu as **Brick**

(4) Enter values for x 5.00 y 0.05 z 0.05

(5) Click **Apply** to create the bar

Step 2.1 Volume Meshing Using Hexahedron



Approximate Size

Select Volumes
1

Approximate Size 0.05

☑ Preview

☑ Check For Overlapping Surfaces

☑ Apply Size Before Meshing

(1) In meshing mode, click **Volumes** entity

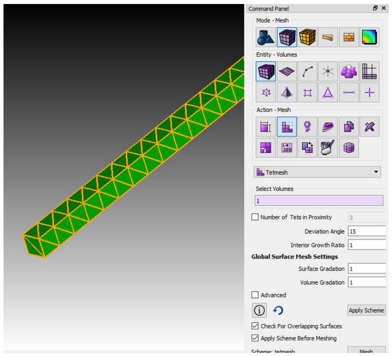
(2) Click **Intervals** action

(3) Select **Volume** field

(4) Set the **Approximate Size** to 0.05

(5) Click **Apply** and **Mesh**

Step 2.2 Volume Meshing Using Tetrahedron (an alternative meshing scheme)



Tetmesh

Select Volumes
1

□ Number of Tets in Proximity 3

Deviation Angle 15

Interior Growth Ratio 1

Global Surface Mesh Settings

Surface Gradation 1

Volume Gradation 1

□ Advanced

☑ Check For Overlapping Surfaces

☑ Apply Scheme Before Meshing

Scheme: tetmesh

(1) In meshing mode, click **Volumes** entity

(2) Click **Intervals** action

(3) Select **Volume** field, Set the **Approximate Size** to 0.05, Click **Apply**

(4) Click **Mesh** action

(5) Select **Tetmesh**, Select **Volume** field, set **Surface Graduation** and **Volume Graduation** as 1

(6) Click **Apply Scheme** and **Mesh**

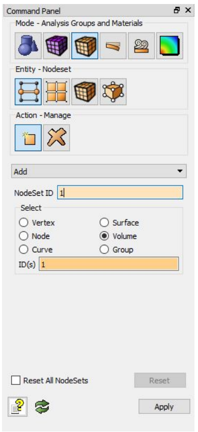
Step 3 Create node sets

After identifying the desired surfaces and volumes for creating nodesets in the explorer on the left of the screen, in the command prompt (at the bottom of the screen) type:

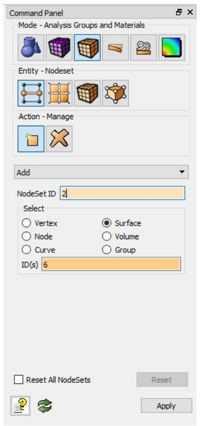
∗ nodeset 1 volume all

∗ nodeset 2 surface 6

Alternatively, the GUI can be used as follows:



NodeSet ID 1
Select
○ Vertex ○ Surface
○ Node ● Volume
○ Curve ○ Group
ID(s) 1
□ Reset All NodeSets



NodeSet ID 2
Select
○ Vertex ● Surface
○ Node ○ Volume
○ Curve ○ Group
ID(s) 6
□ Reset All NodeSets

(1) In Analysis Groups and Materials mode, click **Nodeset**

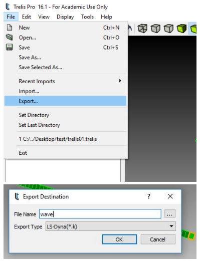
(2) Click **Nodeset** action

(3) Select **Volume** or **Surface** field

(4) Set ID to 1 for volume, 2 for surface

(5) Click **Apply**

Step 4 Exporting the Model



File Name wave
Export Type LS-Dyna(*.k)

(1) Click **File**

(2) Click **Export**

(3) Select LS_Dyna($*.k$)

Journal file: All these process are recorded in *.gou file. A user can use it directly in command line. It is a good idea to store these commands in a file for your own reference with comments, which can be copy-pasted into the command line:

```
# GEOMETERY #
create brick x 5.00 y 0.05 z 0.05
volume 1 size 0.05
mesh volume all
# BOUNDARY CONDITIONS #
#material properties, apply velocity
nodeset 1 volume 1
#to restrain in x on end of bar
nodeset 2 surface 6
```

II Control file

Based on the previous discussions and the problem definition, the following control file is used. Typically, to start a new problem, use an old control file and modify it to ensure correct syntax of the cards is used. In this example, each of the types of different options in this file are separated by comment lines which have no specific format, and only need to be different than any card names.

```
# # # MODEL # # #

*MODEL
# Name
bar.k
```

```
# # # PARAMETERS # # #

*TIME PARAMETERS
# TIME_ END
1
#IAUTO_TS (AUTO SELECT TIME STEP = 1)
0
# TIME STEP IF IAUTO_TS = 0 / FACTOR FOR TIME STEP IF IAUTO_TS = 1
.45E-3

*FORMULATION PARAMETERS
# IFINITE _ STRAIN, ILAGRANGIAN
0 1

# # # SET NAMES # # #

*SET NAME
#SET NO (CUBIT or TRELIS)
1
#NAME OF THIS SET
WHOLE_BAR

*SET NAME
#SET NO (CUBIT or TRELIS)
2
#NAME OF THIS SET
WALL_NODES

# # # ASSIGN PARAMETERS TO NODE SETS # # #

*ORDERED BOUNDARY CONDITION
#SET NO. OR NODE SET NAME
WHOLE_BAR
#FIXITY
0 1 1

*ORDERED BOUNDARY CONDITION
#SET NO. OR NODE SET NAME
WALL_NODES
#FIXITY
1 1 1

*INITIAL CONDITION
#SET NO. OR NODE SET NAME
WHOLE_BAR
#VEL
-0.1 0 0
```

```
# MATERIAL PROPERTIES #
#
#E = 29.0e6 Pa
#Poisson's ratio = 0.0
#density = 0.73395 x 1E-3 kg per meter cubed
#

*MATERIALS
#SET NO. OR NODE SET NAME
WHOLE_BAR
#TYPE
1
#props
0 1e6 100 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

# # # OUTPUT PARAMETERS # # #

*OUTPUT PARAMETERS
#TIME_OUTPUT
.001

*OUTPUT VARIABLES
all
```

5.1.3 ParaView for postprocessing

Output files are in VTK format are generated for each output step. For example, `MEGAOUT_0054.vtk` is the 54th output file. ParaView (https://www.paraview.org/) is a free software for postprocessing these files. The following gives the specific steps on how to use ParaView to visualize the numerical results.

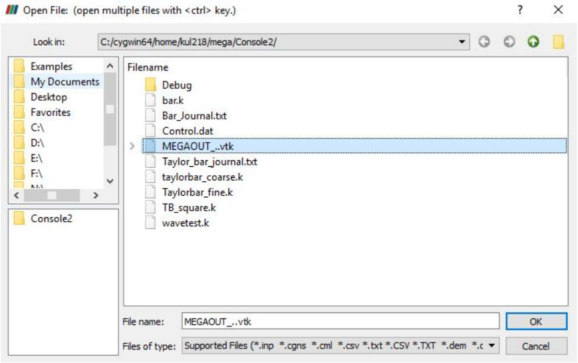
Step 1 Open .vtk file.(Figure.2)

Step 2 Apply the model.(Figure.3)

Step 3 Select representation type (Points and Point size).(Figure.4)

Step 4 Find Data.(Figure.5)

Step 5 Select points.(Figure.6)



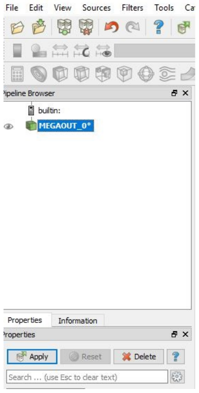
Look in: C:/cygwin64/home/kul218/mega/Console2/

Filename
Debug
bar.k
Bar_Journal.txt
Control.dat
MEGAOUT_..vtk
Taylor_bar_journal.txt
taylorbar_coarse.k
Taylorbar_fine.k
TB_square.k
wavetest.k

File name: MEGAOUT_..vtk

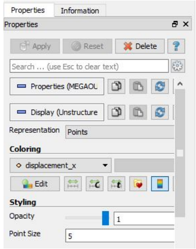
Files of type: Supported Files (*.inp *.cgns *.cml *.csv *.txt *.CSV *.TXT *.dem *.c

Figure 2: Open vtk file



builtin:
MEGAOUT_0*

Figure 3: Apply the model



Representation Points

displacement_x

Opacity 1

Point Size 5

Figure 4: Select representation type

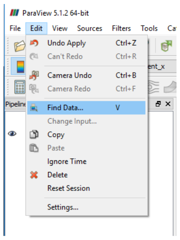


Figure 5: Find Data

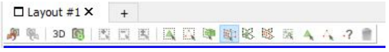
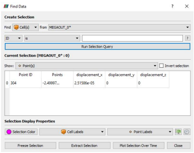


Figure 6: Select points

Step 6 Plot selection over time.(Figure.7)



Create Selection

Find: Cell(s) from MEGAOUT_0*

ID is

Current Selection (MEGAOUT_0* : 0)

Show: Point(s) □ Invert selection

	Point ID	Points	displacement_x	displacement_y	displacement_z
0	304	-2.49997...	2.51586e-05	0	0

Selection Display Properties

Figure 7: Plot selection over time

Step 7 Apply the Plot.(Figure.8)

Step 8 Select values.(Figure.9)

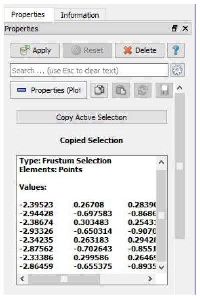
Step 9 Output figures.(Figure.10)

Step 10 Output values in Excel.(Figure.11)

(1) Create a new Layout (Top)

(2) Click SperadSheet View (Bottom)

(3) Output data



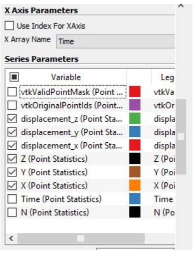
Copied Selection

```
Type: Frustum Selection
Elements: Points

Values:

-2.39523    0.26708     0.2839(
-2.94428    -0.697583   -0.868(
-2.38674    0.303483    0.2543:
-2.93326    -0.650314   -0.907(
-2.34235    0.263183    0.2942(
-2.87562    -0.702643   -0.8551
-2.33386    0.299586    0.2646!
-2.86459    -0.655375   -0.8935
```

Figure 8: Apply the Plot



X Axis Parameters

☐ Use Index For XAxis

X Array Name: Time

Series Parameters

▣	Variable		Leg
☐	vtkValidPointMask (Point ...	🟥	vtkVa
☐	vtkOriginalPointIds (Point...	🟪	vtkOr
☑	displacement_z (Point Sta...	🟩	displa
☑	displacement_y (Point Sta...	🟦	displa
☑	displacement_x (Point Sta...	🟥	displa
☑	Z (Point Statistics)	⬛	Z (Poi
☑	Y (Point Statistics)	🟫	Y (Poi
☑	X (Point Statistics)	🟧	X (Poi
☐	Time (Point Statistics)	🟦	Time
☐	N (Point Statistics)	⬛	N (Po

Figure 9: Select values

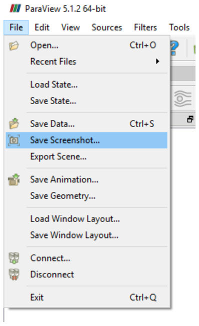


Figure 10: Output figurese

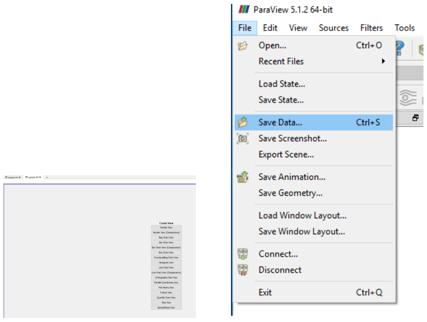


Figure 11: Output values in Excel

5.2 Taylor bar impact problem

5.2.1 Problem description

This problem models that an aluminum cylindrical bar is impacting a rigid wall which is assumed to be frictionless. This benchmark problem is used to examine if a code is able to compute a nonlinear large deformation problem accurately. The radius and height of the bar are 0.391 cm and 2.346 cm, respectively, and the material properties are: density $\rho = 2700$ kg/m^3, Young's modulus $E = 78.2$ GPa, Poisson's ratio $\nu = 0.3$, and J2 plasticity is used with an initial yield strength $Y = 0.29$ GPa, with an exponential hardening rule $Y(1 + 125\bar{e}^p)^{0.1}$ where $\bar{e}^p$ is the effective plastic strain. The initial impact velocity is 373 m/sec. The final height of the bar after deformation was experimentally measured as 1.651 cm (Taylor 1948, Wilkins and Guian 1973), and numerical results and experiemntal data are compared in Table 1. As we can see, the Lagrangian RKPM result by using MEGA matches well with the experimental data with a very small difference 0.24%, and the semi-Lagrangian RKPM solution is also very accurate, which indicates the success of the RKPM implementation into MEGA code for nonlinear large deformation problems.

The construction of the model is made slightly more complicated due to the intent to avoid a non-uniform paved mesh.

Table 1: Comparison of deformed height and radius for Talyor bar impact problem

	Lagrangian RKPM	Semi-Lagrangian RKPM	Experimental data
Height (cm)	1.647	1.691	1.651
Radius (cm)	0.792	0.799	NA

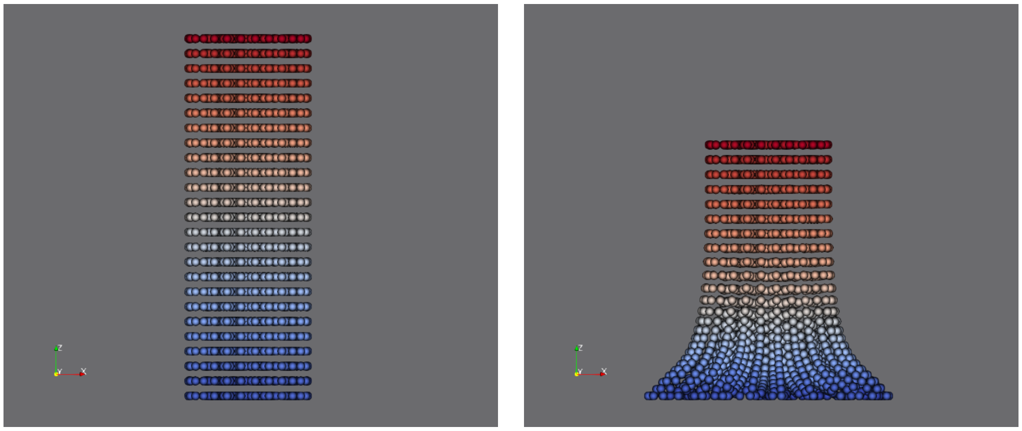


Figure 12: Numerical solution: left, undeformed profile; right, final deformed profile

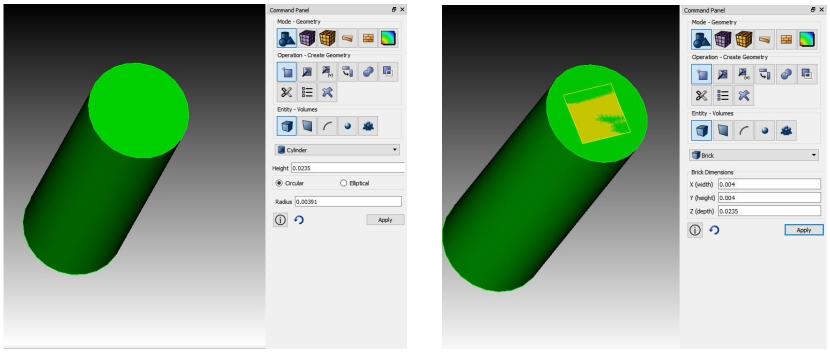
5.2.2 Input files preparation for Taylor bar impact problem

I Process of creating the model in *CUBIT* or *TRELIS*

Step 1 Creating the Model

(1) Create cylinder radius 0.00391 height 0.0235 (Figure.13)

(2) Create brick 0.004, 0.004, 0.0235 (Figure.13)

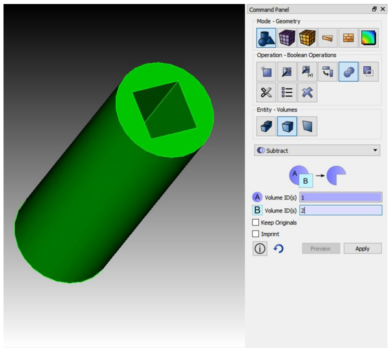


Cylinder
Height 0.0235
● Circular ○ Elliptical
Radius 0.00391

Brick
Brick Dimensions
X (width) 0.004
Y (height) 0.004
Z (depth) 0.0235

Figure 13: Creating the Brick

(3) subtract body 2 from 1 (Figure.14)



Subtract
A Volume ID(s) 1
B Volume ID(s) 2
□ Keep Originals
□ Imprint

Figure 14: Subtract

(4) Create brick 0.004, 0.004, 0.0235 (Figure.15)

(5) Webcut.(Figure.15)

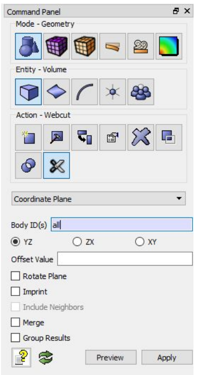
[5.1]Webcut body all with plane xplane

[5.2]Webcut body all with plane yplane

[5.3]Webcut body 8 6 4 1 with plane xplane rotate 45 about z

[5.4]Webcut body 8 6 4 1 with plane xplane rotate 135 about z

Step 2 Merge.(Figure.16)


Command Panel

Mode - Geometry

Entity - Volume

Action - Webcut

Coordinate Plane

Body ID(s) all

● YZ ○ ZX ○ XY

Offset Value

□ Rotate Plane

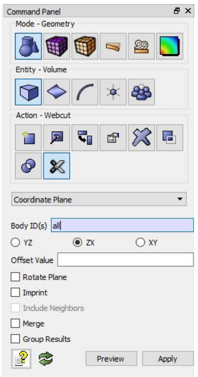
□ Imprint

□ Include Neighbors

□ Merge

□ Group Results

Preview Apply


Command Panel

Mode - Geometry

Entity - Volume

Action - Webcut

Coordinate Plane

Body ID(s) all

○ YZ ● ZX ○ XY

Offset Value

□ Rotate Plane

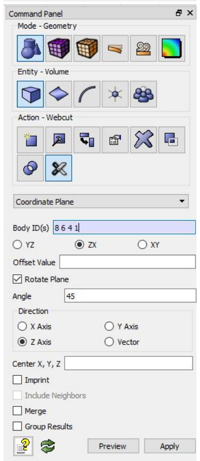
□ Imprint

□ Include Neighbors

□ Merge

□ Group Results

Preview Apply


Command Panel

Mode - Geometry

Entity - Volume

Action - Webcut

Coordinate Plane

Body ID(s) 8 6 4 1

○ YZ ● ZX ○ XY

Offset Value

☑ Rotate Plane

Angle 45

Direction

○ X Axis ○ Y Axis

● Z Axis ○ Vector

Center X, Y, Z

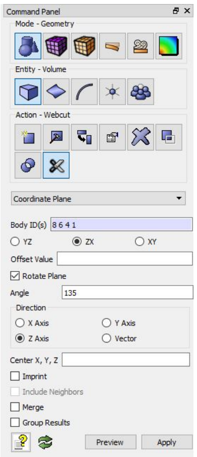
□ Imprint

□ Include Neighbors

□ Merge

□ Group Results

Preview Apply


Command Panel

Mode - Geometry

Entity - Volume

Action - Webcut

Coordinate Plane

Body ID(s) 8 6 4 1

○ YZ ● ZX ○ XY

Offset Value

☑ Rotate Plane

Angle 135

Direction

○ X Axis ○ Y Axis

● Z Axis ○ Vector

Center X, Y, Z

□ Imprint

□ Include Neighbors

□ Merge

□ Group Results

Preview Apply

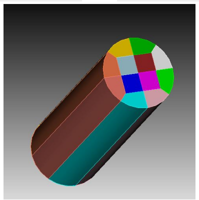
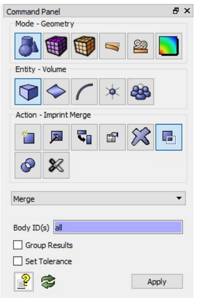


Figure 15: Webcut

(1) Merge all volume: This mechanism attempts to take two overlapping surfaces and merge them into one surface shared by two volumes.



Command Panel

Mode - Geometry

Entity - Volume

Action - Imprint Merge

Merge

Body ID(s) all

□ Group Results

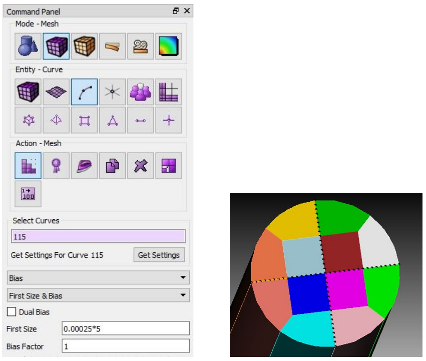
□ Set Tolerance

Apply

Figure 16: Merge

Step 3 Mesh

(1) Curve scheme equal size 0.001. (Figure.17)



Command Panel

Mode - Mesh

Entity - Curve

Action - Mesh

Select Curves

115

Get Settings For Curve 115

Get Settings

Bias

First Size & Bias

□ Dual Bias

First Size 0.00025*5

Bias Factor 1

Figure 17: Scheme Curve

(2) Curve sides interval 2. (Figure.18)

(3) Surface scheme map : mesh surface. (Figure.19)

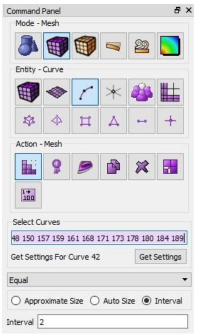
(4) Curve scheme equal size 0.001 (Figure.20)

(5) volume all scheme map : mesh volume all (Figure.20)

Step 4 Nodeset.(Figure.21)

(1) nodeset 1 add volume all

(2) nodeset 2 add surface



Select Curves: 48 150 157 159 161 168 171 173 178 180 184 189

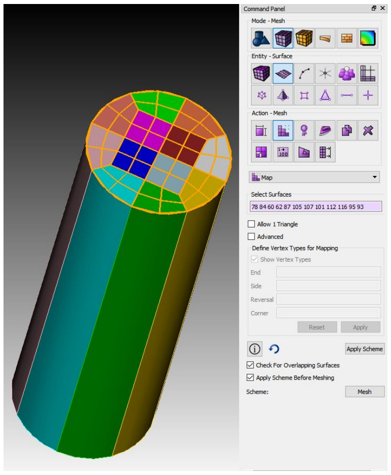
Get Settings For Curve 42 — Get Settings

Equal

○ Approximate Size ○ Auto Size ● Interval

Interval: 2

Figure 18: Interval Curve



Map

Select Surfaces: 78 84 60 62 87 105 107 101 112 116 95 93

□ Allow 1 Triangle

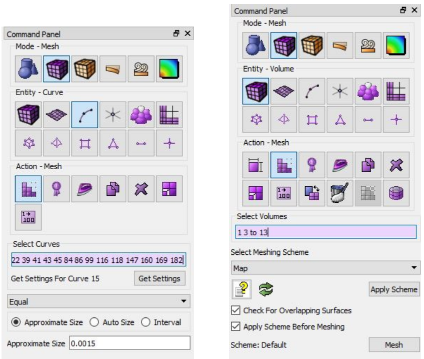
□ Advanced

☑ Check For Overlapping Surfaces

☑ Apply Scheme Before Meshing

Scheme:

Figure 19: Surface scheme map



Select Curves: 22 39 41 43 45 84 86 99 116 118 147 160 169 182

Get Settings For Curve 15

Equal

● Approximate Size ○ Auto Size ○ Interval

Approximate Size: 0.0015

Select Volumes: 1 3 to 13

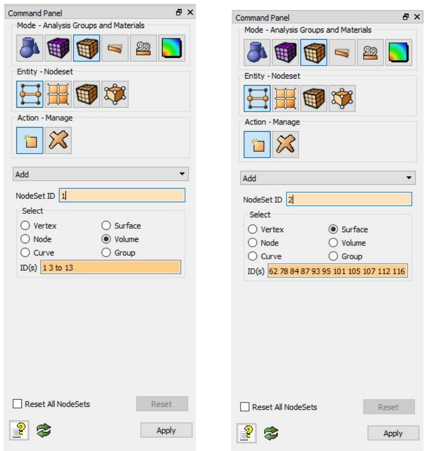
Select Meshing Scheme: Map

☑ Check For Overlapping Surfaces

☑ Apply Scheme Before Meshing

Scheme: Default

Figure 20: volume map



Add
NodeSet ID 1
Select
○ Vertex ○ Surface
○ Node ● Volume
○ Curve ○ Group
ID(s) 1 3 to 13
□ Reset All NodeSets

Add
NodeSet ID 2
Select
○ Vertex ● Surface
○ Node ○ Volume
○ Curve ○ Group
ID(s) 62 78 84 87 93 95 101 105 107 112 116
□ Reset All NodeSets

Figure 21: Nodeset

II Control file

```
# # # MODEL # # #
*MODEL
# Name
taylorbar_coarse.k

# # # PARAMETERS # # #
*TIME PARAMETERS
# TIME_END
1.0E-04
#IAUTO_TS (AUTO SELECT TIME STEP = 1)
0
# TIME STEP IF IAUTO_TS = 0 / FACTOR FOR TIME STEP IF IAUTO_TS = 1
5.0E-09

*INTEGRATION PARAMETERS
# # TYPE (0 = DNI, 1 = SNNI, 2 = NSNI)
0

*FORMULATION PARAMETERS
# IFINITE _ STRAIN, ILAGRANGIAN
1 1

# # # SET NAMES # # #
*SET NAME
# SET NO (in CUBIT or TRELIS)
1
#NAME OF THIS SET
BAR
```

```
*SET NAME
#SET NO (in CUBIT or TRELIS)
2
#NAME OF THIS SET
WALL

# # # ASSIGN PARAMETERS TO NODE SETS # # #
*BOUNDARY CONDITION
#SET NO. OR NODE SET NAME
WALL
#FIXITY
0 0 1

*WINDOW DEFINITION
#SET NO. OR NODE SET NAME
BAR
#WINDOW
1.5

*INITIAL CONDITION
#SET NO. OR NODE SET NAME
BAR
#VEL
0 0 -373

*MATERIALS
#SET NO. OR NODE SET NAME
BAR
#TYPE
2
#props
0.3 78.2e9 2700 0.29e9 125
0.1 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

# # # OUTPUT PARAMETERS # # #
*OUTPUT PARAMETERS
#TIME_OUTPUT
5.0E-07

*OUTPUT VARIABLES
all
```