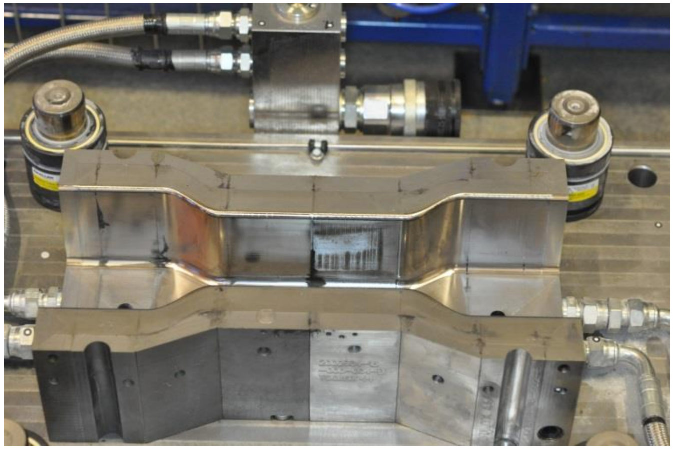
Simulation of Wear Processes in LS-DYNA

Thomas Borrvall, Anders Jernberg and Mikael Schill, DYNAmore Nordic AB

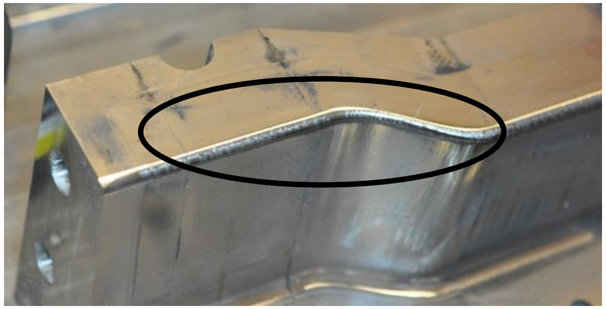
Liang Deng and Mats Oldenburg, Luleå Technical University

Motivation and Example

- Hot forming process reduces life length of tools
  - Hot blank is formed and subsequently cooled (quenching)
  - High contact pressures and cyclic temperatures
  - Scratches due to sliding wear along radii
- Significant cost incurred in replacing worn out tools
  - Important to understand the mechanisms behind wear
- A dog bone wear test illustrates



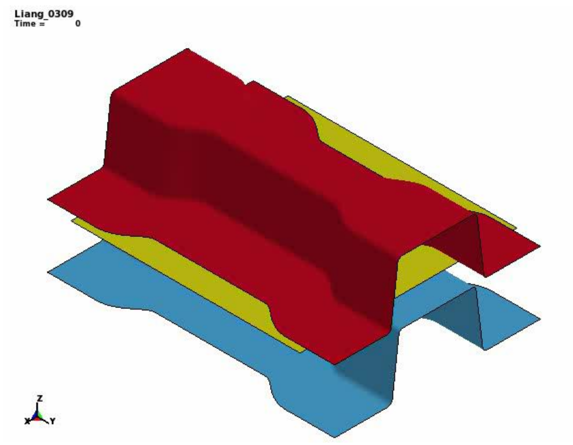
Upper tool



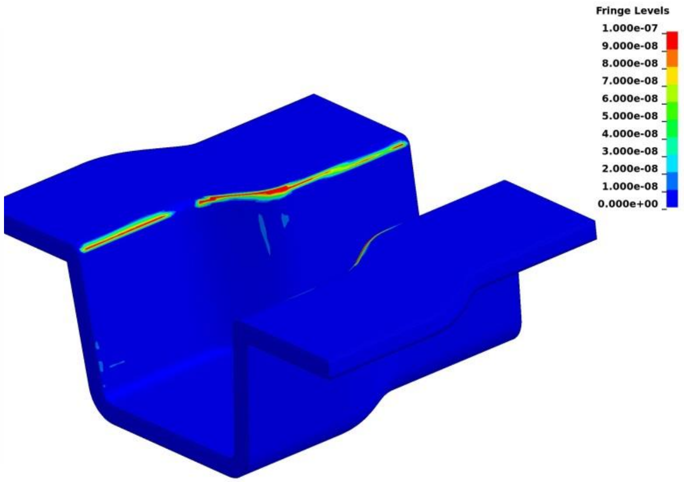
Wear after several strokes

Wear Simulation of Dog Bone

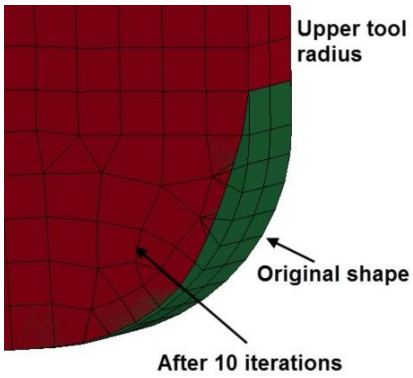
- Possible to post-process wear in LS-PrePost
  - Standard wear law or user defined
- Iteratively modify geometry to simulate long term use
  - Several LS-DYNA runs with intermediate processing in LS-PrePost



Simulation

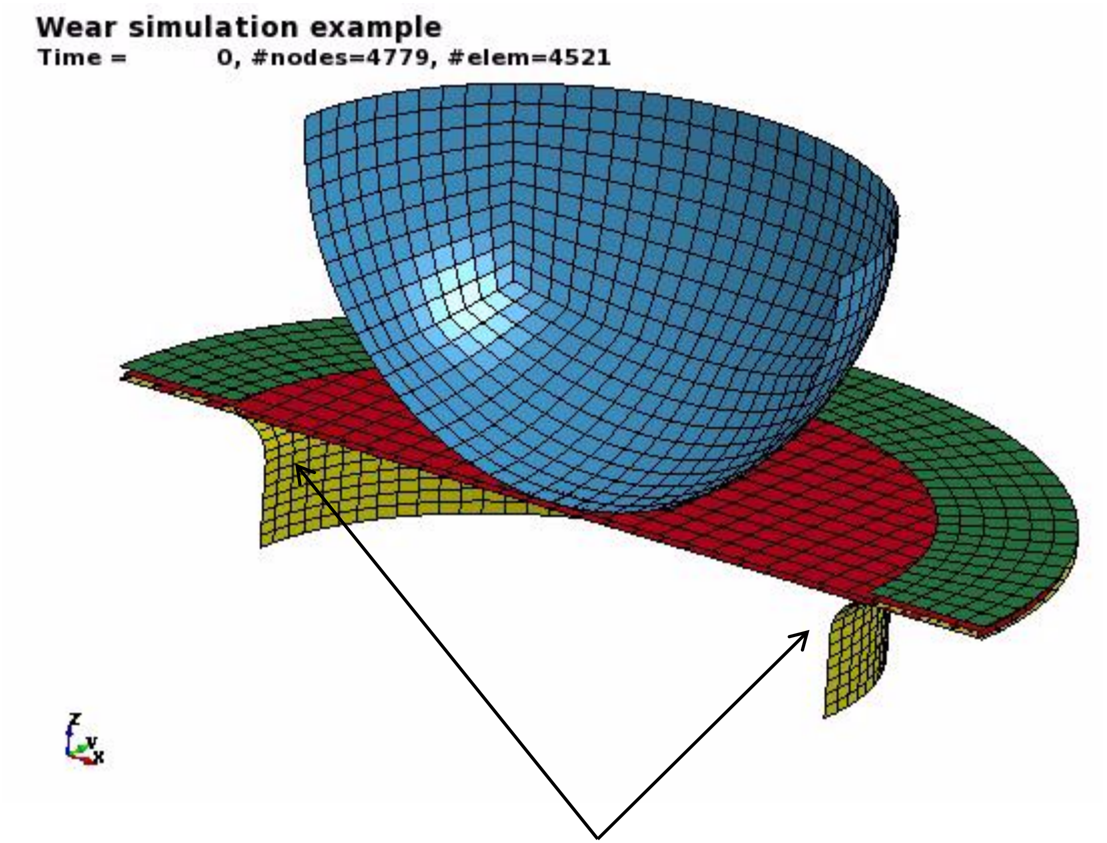


Wear from simulation



Modified tool geometry

Tutorial – wear in deep draw of spherical cup



In particular interested in the wear along the radii of the die

Archard's law

- LS-DYNA computes the wear as a nodal quantity which is expressed in rate form as

$$\dot{w} = K\frac{pv}{H}$$

- w is nodal wear depth in direction of the surface normal. Surface normal is computed from average of element normals. Shell element normals in a wear interface must to be consistently oriented.
- $K(p, v)$ is a dimensionless scale factor which can be given as a function of contact pressure p and sliding velocity v.
- $H(T)$ is the hardness for the contact side. The hardness can be given as a function of nodal temperature T.
- p is the nodal pressure.
- v is the sliding velocity.

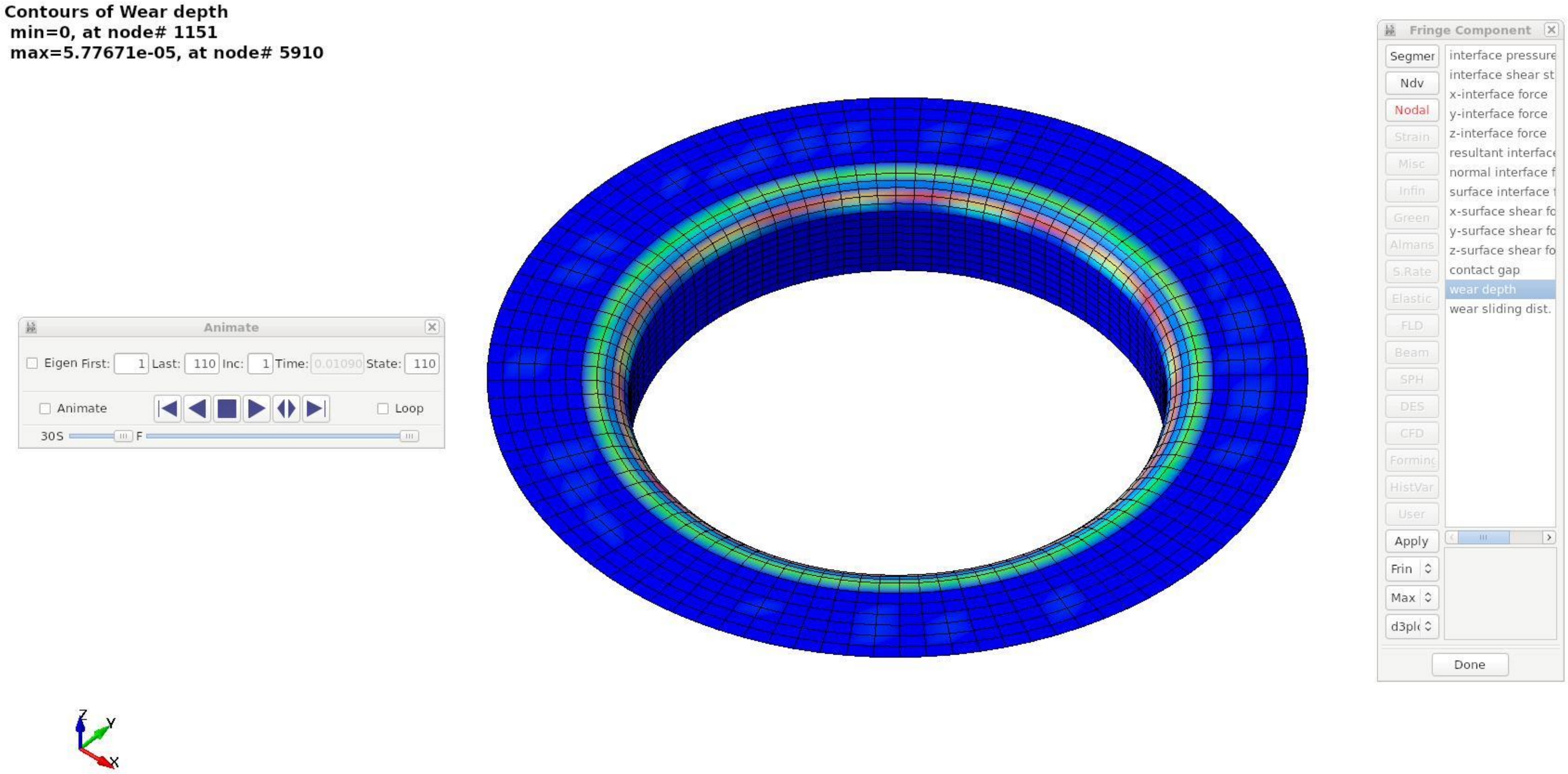
Add wear to an LS-DYNA input deck

***CONTACT_ADD_WEAR**

Card 1	1	2	3	4	5	6	7	8
Variable	CID	WTYPE	P1	P2	P3	P4	P5	P6
Type	I	I	F	F	F	F	F	F
Default	None	0	0.0	0.0	0.0	0.0	0.0	0.0

- Use any of these contacts
  - ***CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_{MORTAR}**
  - ***CONTACT_FORMING_SURFACE_TO_SURFACE_{MORTAR}**
  - ***CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR**
  - ***CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE**
- Set **SPR** and/or **MPR** to ”1” on the contact for interfaces of interest
- Define ***CONTACT_ADD_WEAR**
  - Point to contact **CID**
  - **WTYPE=0** for Archard’s law
  - **P1-P3** parameters (K, H_s, H_m)
- Set **NWEAR>0** on ***DATABASE_EXTENT_INTFOR**
  - **NWEAR=1** for wear depth
  - **NWEAR=2** for wear depth and sliding distance

Post process wear in LS-PrePost



```
Contours of Wear depth
 min=0, at node# 1151
 max=5.77671e-05, at node# 5910
```

Animate

☐ Eigen First: 1 Last: 110 Inc: 1 Time: 0.01090 State: 110

☐ Animate ☐ Loop

30S F

- Run the input in LS-DYNA with *s=intfor* on the command line
- Open the intfor file in LS-PrePost
- In the fringe menu, **wear depth** and **wear sliding dist.** are found

Process wear in LS-DYNA

1 LS-DYNA sim = 1 wear cycle
NCYC cycles = 1 wear stage

The intfor file is for assessing the influence of wear for a given tool geometry, but to understand how wear affects the process in the long haul (thousands of repetitions) we need to import the wear information back into LS-DYNA

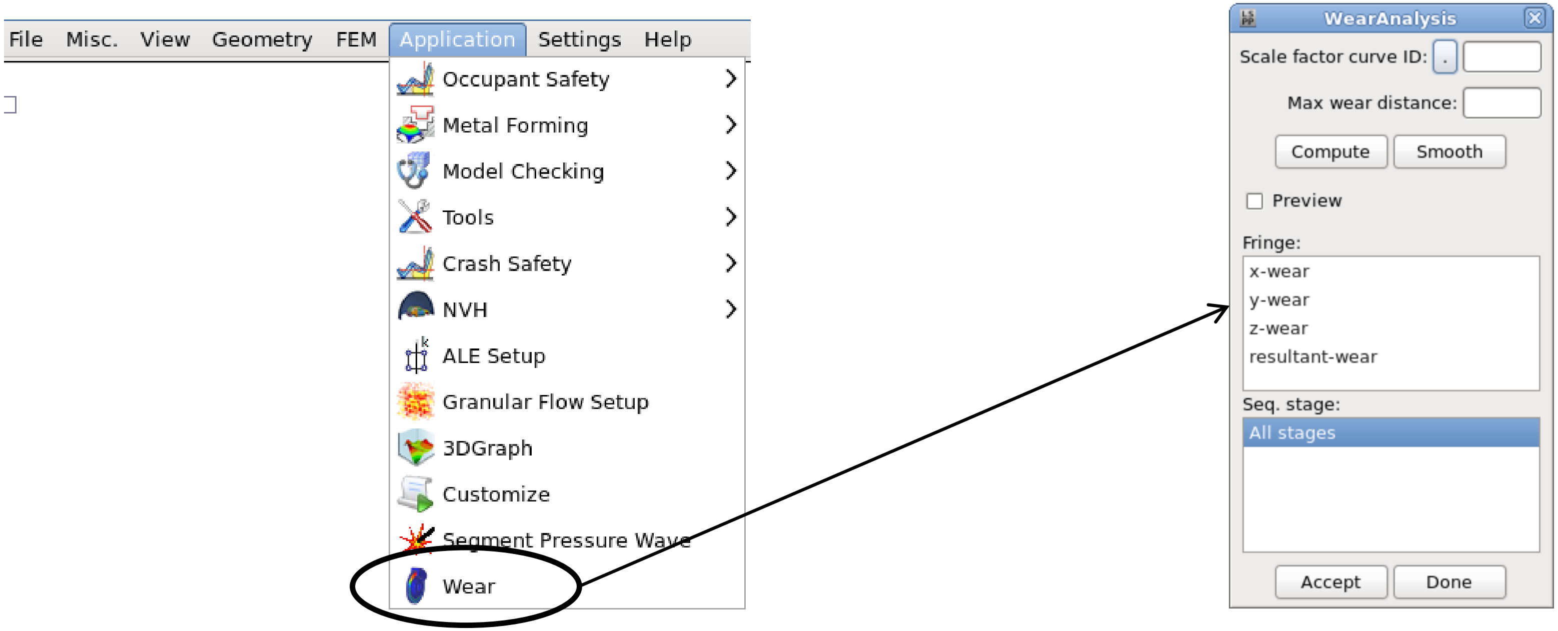
***INITIAL_CONTACT_WEAR**

Card 1	1	2	3	4	5	6	7	8
Variable	CID	NID	WDEPTH	NX	NY	NZ	ISEQ	NCYC
Type	I	I	F	F	F	F	I	I
Default	None	None	None	None	None	None	None	None

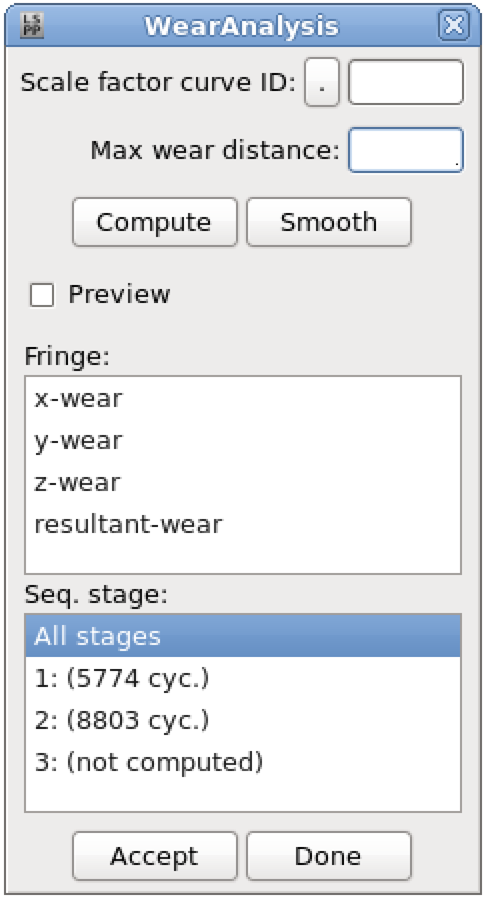
- Set **SPR** and/or **MPR** to ”2” on the contact, this will also make the side available to the intfor file
- Set **NCYC>0** on ***INTERFACE_SPRINGBACK_LSDYNA** to get ***INITIAL_CONTACT_WEAR** data written to *dynain*
- Each card corresponds to a wear increment for a given node for 1 cycle
- It is assumed that the results are valid for 1 stage, i.e., NCYC cycles

Rerunning with wear information

- Include the *dynain* file as is to the *original* input
  - The dynain should not contain anything but the ***INITIAL_CONTACT_WEAR** cards
- Option A – No intermediate processing, rerun the file in LS-DYNA
  - Each node subject to wear will be moved by the wear depth in the direction of wear, times **NCYC**, thus completing 1 stage
  - The user is ”blind”, difficult to obtain a reasonable geometry change
- Option B – Intermediate processing, open the file in LS-PrePost

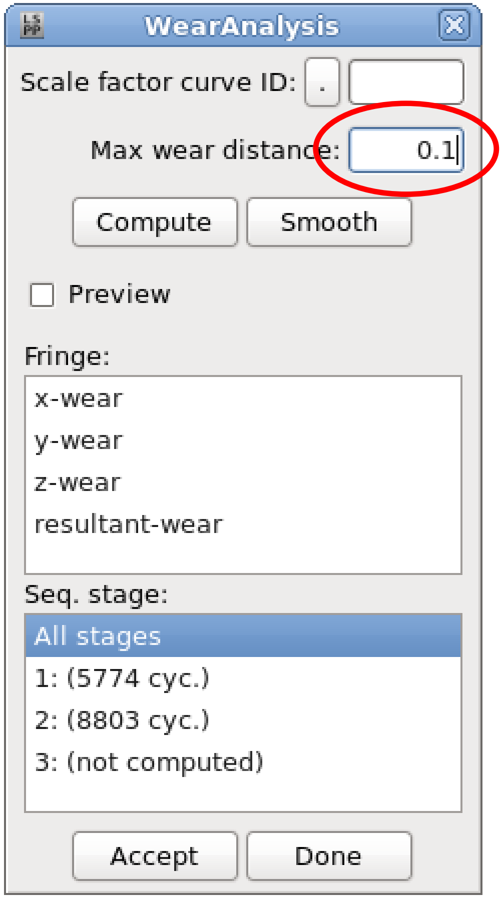


The LS-PrePost Wear Interface



- An ***INITIAL_CONTACT_WEAR** card gives information from *one* cycle, now the intermediate processing step amounts to determine the wear from one *stage*
- Option 1
  - Leave *Scale factor curve ID* blank

The LS-PrePost Wear Interface



Scale factor curve ID: .

Max wear distance: 0.1

□ Preview

Fringe:

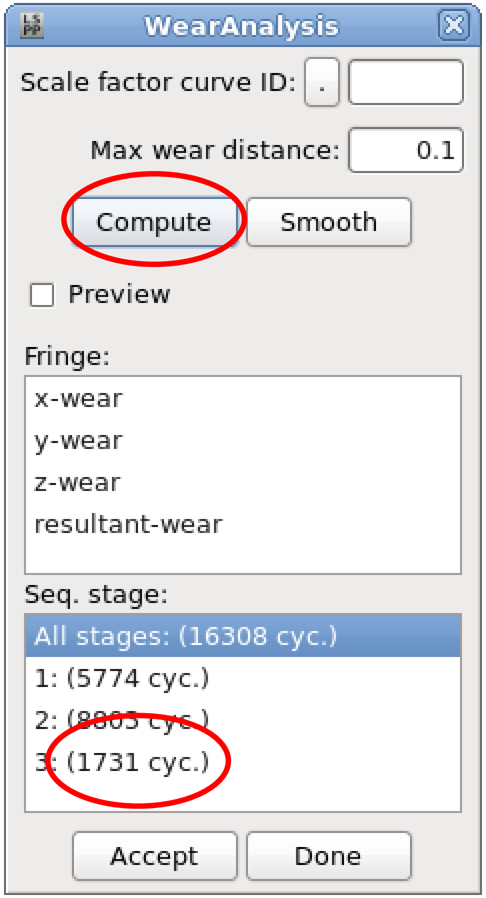
x-wear
y-wear
z-wear
resultant-wear

Seq. stage:

All stages
1: (5774 cyc.)
2: (8803 cyc.)
3: (not computed)

- An ***INITIAL_CONTACT_WEAR** card gives information from *one* cycle, now the intermediate processing step amounts to determine the wear from one *stage*
- Option 1
  - Leave *Scale factor curve ID* blank
  - Set a *Max wear distance*, corresponding to how much geometry change you allow based on the latest run

The LS-PrePost Wear Interface



Scale factor curve ID:

Max wear distance: 0.1

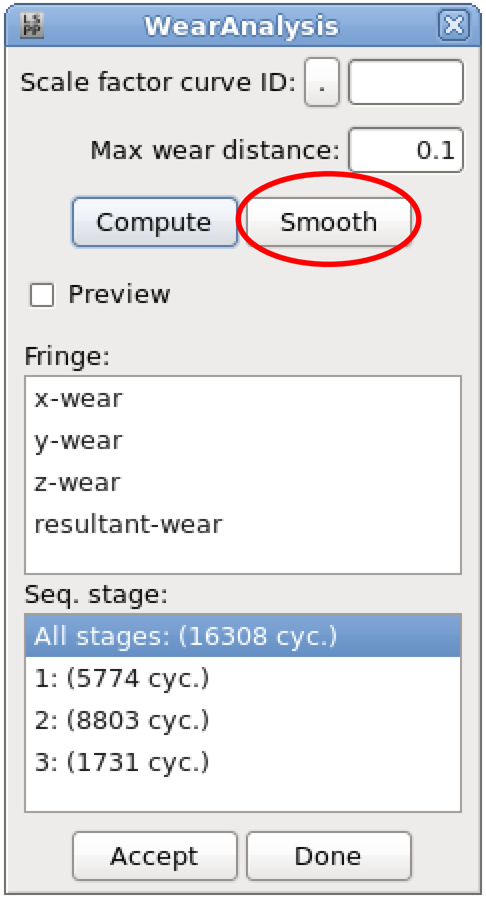
□ Preview

Fringe:
- x-wear
- y-wear
- z-wear
- resultant-wear

Seq. stage:
- All stages: (16308 cyc.)
- 1: (5774 cyc.)
- 2: (8803 cyc.)
- 3: (1731 cyc.)

- An ***INITIAL_CONTACT_WEAR** card gives information from *one* cycle, now the intermediate processing step amounts to determine the wear from one *stage*
- Option 1
  - Leave *Scale factor curve ID* blank
  - Set a *Max wear distance*, corresponding to how much geometry change you allow based on the latest run
  - Click *Compute*, LS-PrePost will determine the number of cycles required for the max wear at any node to reach *Max wear distance*, assuming the wear in each cycle is constant

The LS-PrePost Wear Interface



WearAnalysis

Scale factor curve ID:

Max wear distance: 0.1

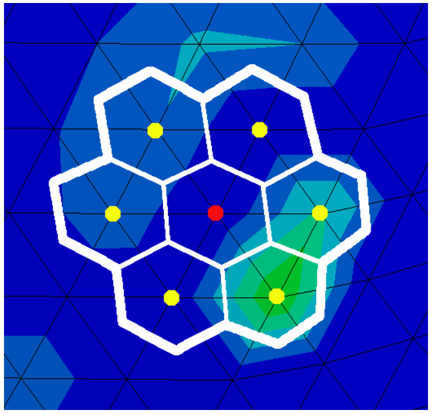
□ Preview

Fringe:

x-wear
y-wear
z-wear
resultant-wear

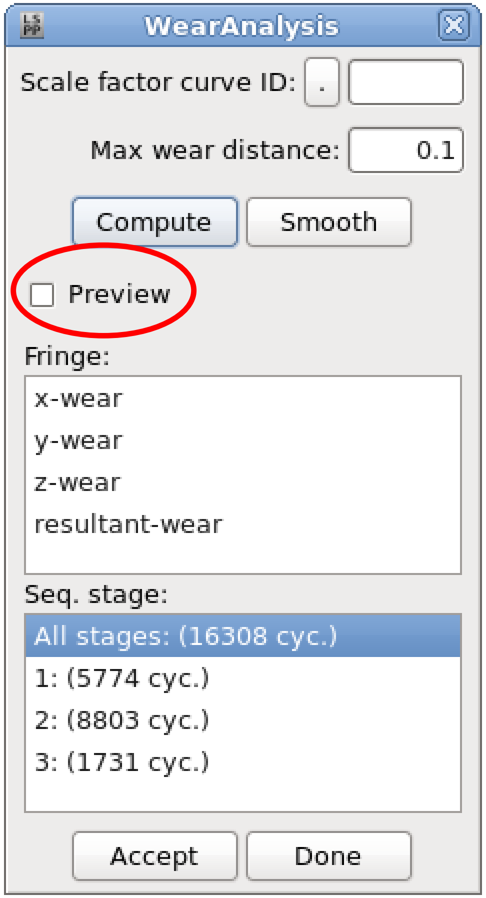
Seq. stage:

All stages: (16308 cyc.)
1: (5774 cyc.)
2: (8803 cyc.)
3: (1731 cyc.)



- Option 1
  - Leave *Scale factor curve ID* blank
  - Set a *Max wear distance*, corresponding to how much geometry change you allow based on the latest run
  - Click *Compute*, LS-PrePost will determine the number of cycles required for the max wear at any node to reach *Max wear distance*, assuming the wear in each cycle is constant
  - Repeatedly click *Smooth* to smooth the geometry change, to even out local ”spots” in the wear

The LS-PrePost Wear Interface



Scale factor curve ID: .

Max wear distance: 0.1

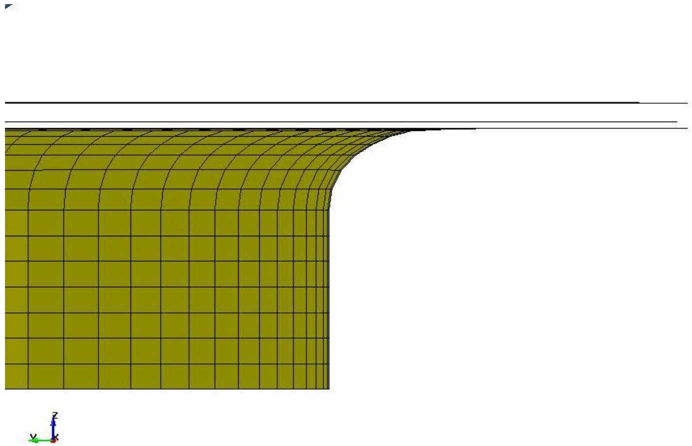
□ Preview

Fringe:

x-wear
y-wear
z-wear
resultant-wear

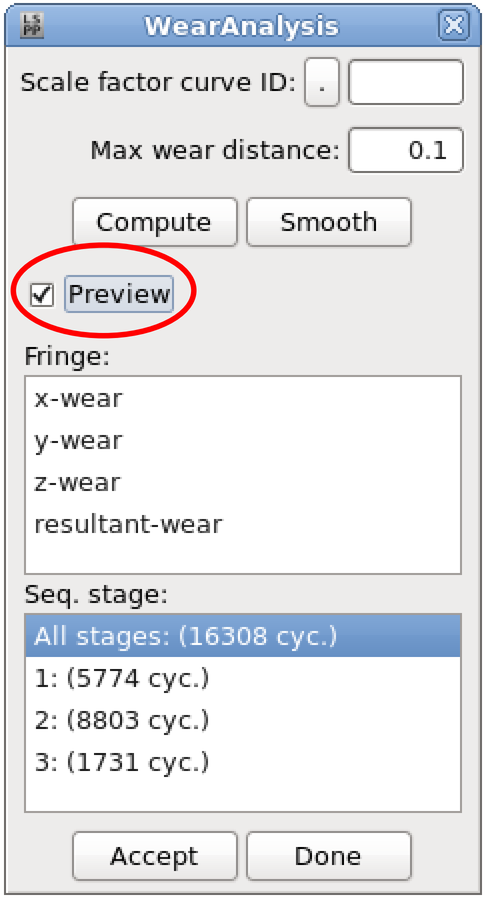
Seq. stage:

All stages: (16308 cyc.)
1: (5774 cyc.)
2: (8803 cyc.)
3: (1731 cyc.)



- Option 1
  - Leave *Scale factor curve ID* blank
  - Set a *Max wear distance*, corresponding to how much geometry change you allow based on the latest run
  - Click *Compute*, LS-PrePost will determine the number of cycles required for the max wear at any node to reach *Max wear distance*, assuming the wear in each cycle is constant
  - Repeatedly click *Smooth* to smooth the geometry change, to even out local ”spots” in the wear
  - The geometry change can and should be previewed by checking *Preview* throughout

The LS-PrePost Wear Interface



WearAnalysis

Scale factor curve ID: [.] [ ]

Max wear distance: 0.1

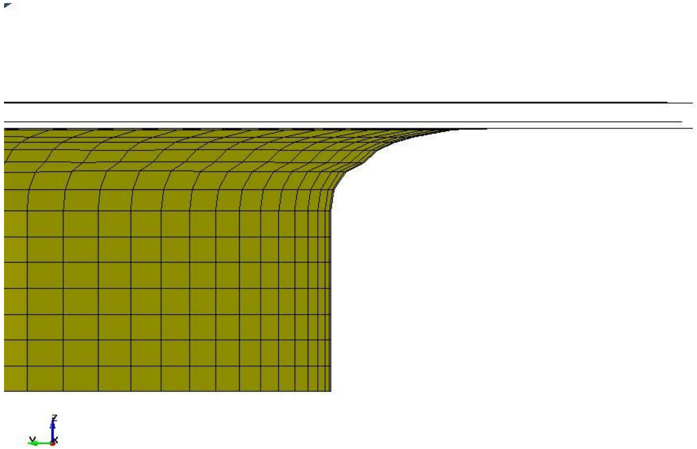
☑ Preview

Fringe:

- x-wear
- y-wear
- z-wear
- resultant-wear

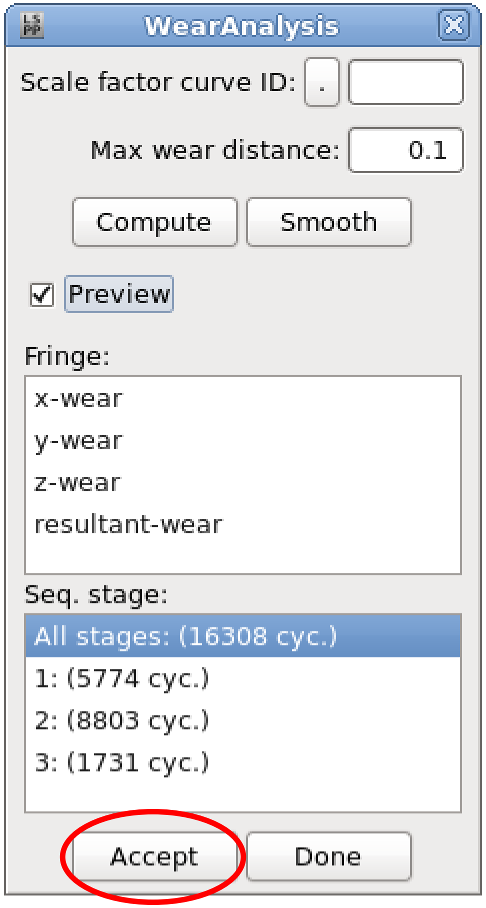
Seq. stage:

- All stages: (16308 cyc.)
- 1: (5774 cyc.)
- 2: (8803 cyc.)
- 3: (1731 cyc.)



- Option 1
  - Leave *Scale factor curve ID* blank
  - Set a *Max wear distance*, corresponding to how much geometry change you allow based on the latest run
  - Click *Compute*, LS-PrePost will determine the number of cycles required for the max wear at any node to reach *Max wear distance*, assuming the wear in each cycle is constant
  - Repeatedly click *Smooth* to smooth the geometry change, to even out local ”spots” in the wear
  - The geometry change can and should be previewed by checking *Preview* throughout

The LS-PrePost Wear Interface



WearAnalysis

Scale factor curve ID:

Max wear distance: 0.1

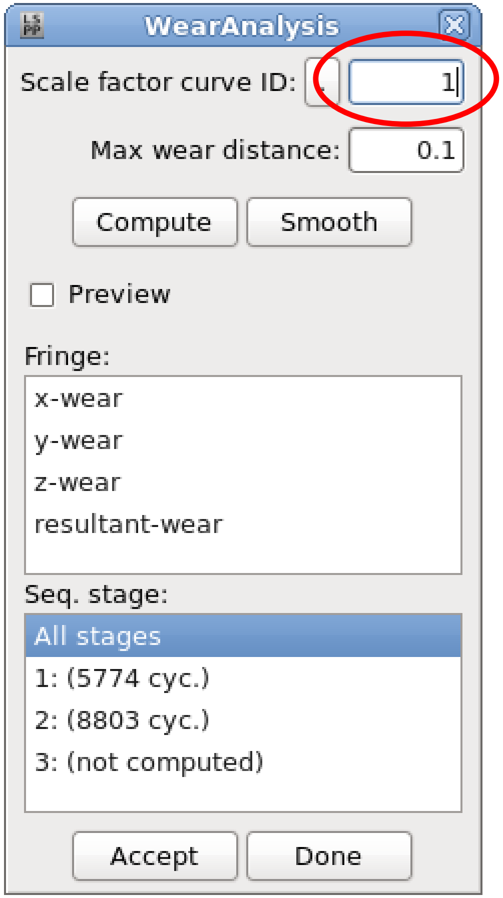
☑ Preview

Fringe:
- x-wear
- y-wear
- z-wear
- resultant-wear

Seq. stage:
- All stages: (16308 cyc.)
- 1: (5774 cyc.)
- 2: (8803 cyc.)
- 3: (1731 cyc.)

- An ***INITIAL_CONTACT_WEAR** card gives information from *one* cycle, now the intermediate processing step amounts to determine the wear from one *stage*
- Option 1
  - Leave *Scale factor curve ID* blank
  - Set a *Max wear distance*, corresponding to how much geometry change you allow based on the latest run
  - Click *Compute*, LS-PrePost will determine the number of cycles required for the max wear at any node to reach *Max wear distance*, assuming the wear in each cycle is constant
  - Repeatedly click *Smooth* to smooth the geometry change, to even out local ”spots” in the wear
  - The geometry change can and should be previewed by checking *Preview* throughout
  - If anything goes wrong, set a new *Max wear distance* and repeat the procedure, click *Accept* when satisfied

The LS-PrePost Wear Interface



Scale factor curve ID: 1

Max wear distance: 0.1

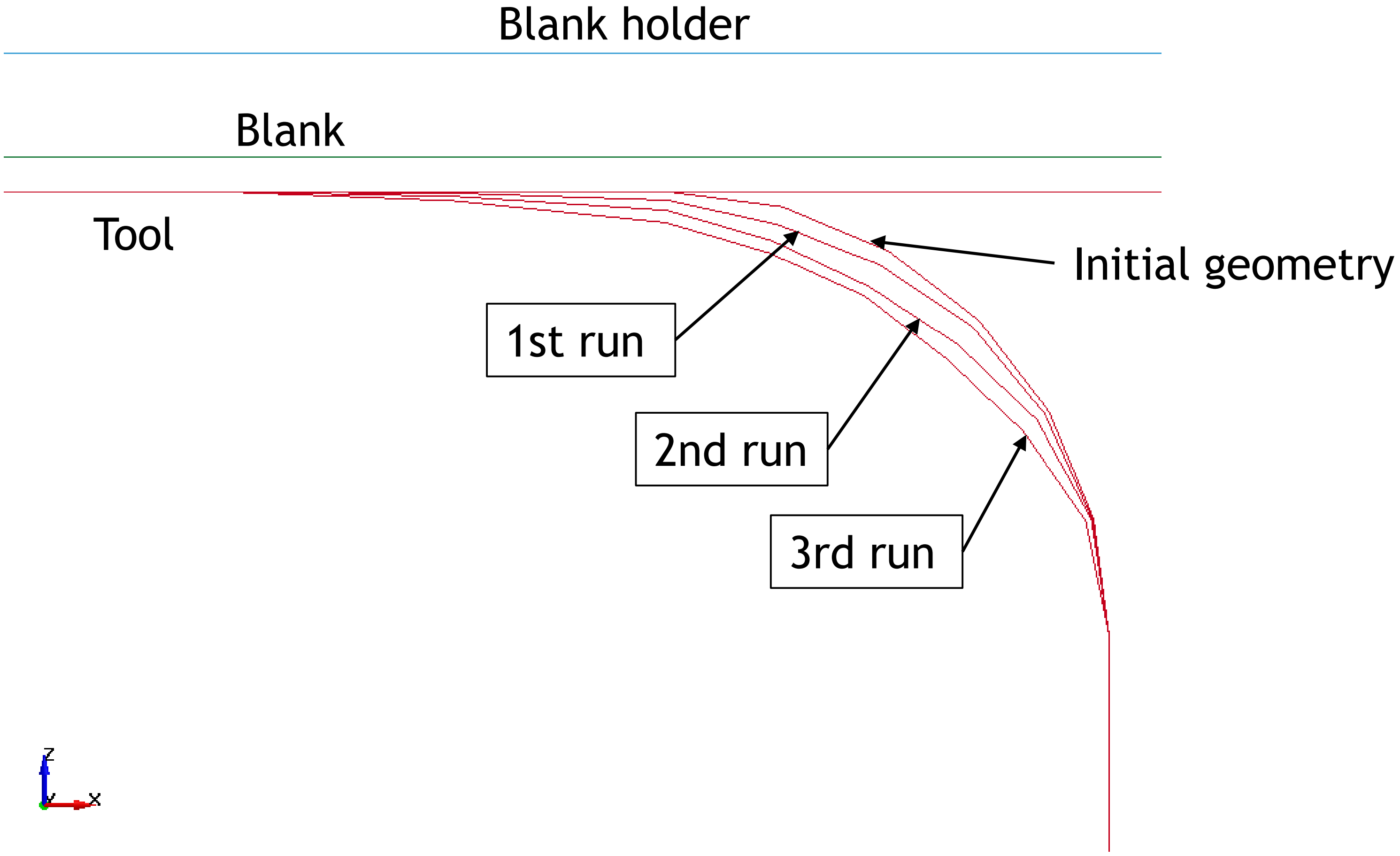
□ Preview

Fringe:
- x-wear
- y-wear
- z-wear
- resultant-wear

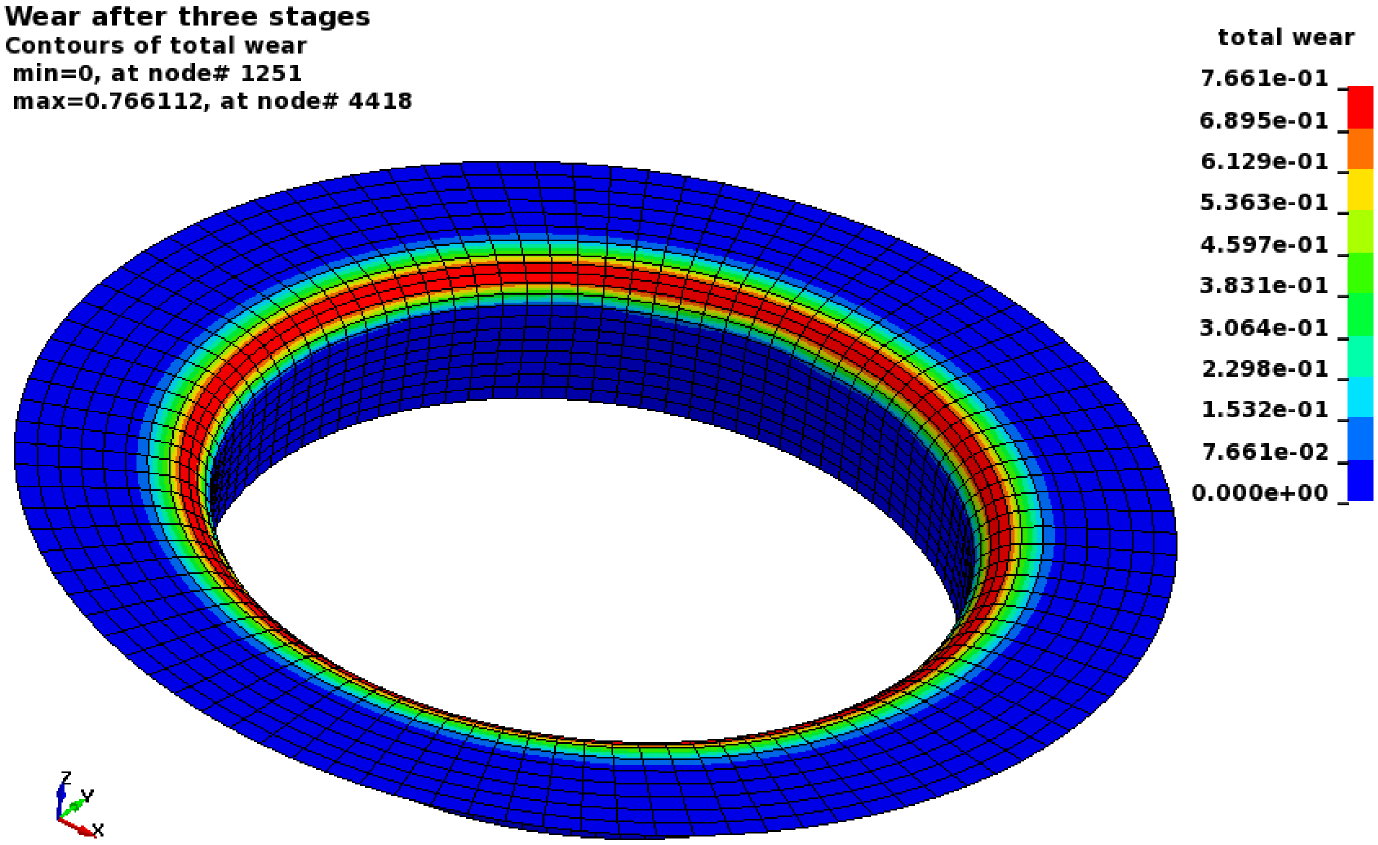
Seq. stage:
- All stages
- 1: (5774 cyc.)
- 2: (8803 cyc.)
- 3: (not computed)

- Sometimes the surface hardness changes with depth, for which the wear can *not* be assumed constant in each cycle
- Option 2
  - Set *Scale factor curve ID* to a curve containing a scale factor as function of total wear depth *d*
  - The procedure is then follows the one of Option 1
- After the geometry change is accepted, save the file to a new input for rerunning in LS-DYNA with the updated tool geometry
- This describes one stage in the entire wear process, to be repeated

Wear after three consecutive wear simulations



Fringe plot of total wear



Wear after three stages
Contours of total wear
min=0, at node# 1251
max=0.766112, at node# 4418

Automated multi-stage run script

- For cases when the intermediate processing step can be assumed identical everything can be done using a script

run.sh

```
#!/bin/bash
mppsub 32 mppdyna_d_dev_abcde_platformmpi -c -l -b i=run.k memory=400m
lspre43 –nographics c=wear.cfile
Clean

mppsub 32 mppdyna_d_dev_abcde_platformmpi -c -l -b i=run.k memory=400m
lspre43 –nographics c=wear.cfile
Clean

mppsub 32 mppdyna_d_dev_abcde_platformmpi -c -l -b i=run.k memory=400m
lspre43 –nographics c=wear.cfile
# Don't Clean. keeping the results from last run
# All previous wear results are now saved in "run.k”

#Note that the submit script, mppsub in this case, must not return the prompt until the
#simulation has finished completely.
#The ”original_input.k” is copied to a ”run.k” before this script is started.
```

LS-PrePost command file

wear.cfile

```
openc keyword "/disk/home/anders/wear/original_input.k”
import keyword nooffset
import keyword "/disk/home/anders/wear/dynain”
save keywordoutversion 7
wear maxdist 0.5
wear compute
wear smooth
wear accept
save keyword "/disk/home/anders/wear/run.k”
exit
```

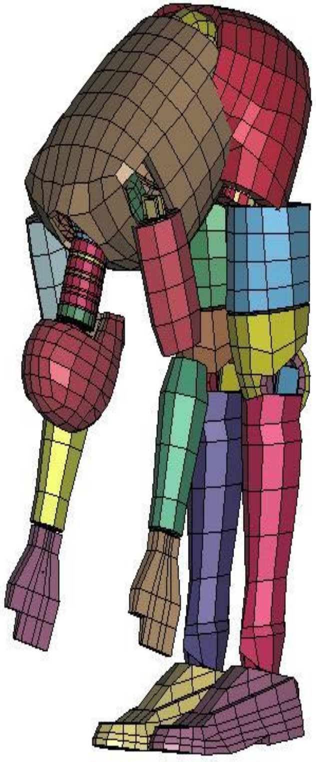
- Read in the original keyword file
- Import the dynain file
- Set the max wear distance to 0.5
- Compute the wear
- Smooth the wear once
- Update the node coordinates
- Write a new input file, run.k

Concluding remarks and general recommendations

- LS-DYNA and LS-PrePost can be used in parallel to simulate wear processes
  - ”In the beginning” but conceptually works well
  - Manual or automated processing
- For the best results
  - The contact pressure and friction should be smooth
  - Mesh density should be relatively fine for parts where wear is important
  - Stiff contact will localize wear for faceted geometries, soften to distribute
  - One wear stage must not change the geometry too much as this will result in unrealistic wear in subsequent stages
- Part is available in R9.0 and all is available in R10
  - R9.0 – post processing only
  - R10 – entire process simulation

Future work – mainly driven by user feedback

- LS-DYNA
  - More contact types(?)
  - More advanced wear laws(?)
- LS-PrePost
  - Local smoothing(?)
  - Manual interaction(?)



Thank you!

DYNA MORE

Your LS-DYNA distributor and more