Workshop: Dummy positioning

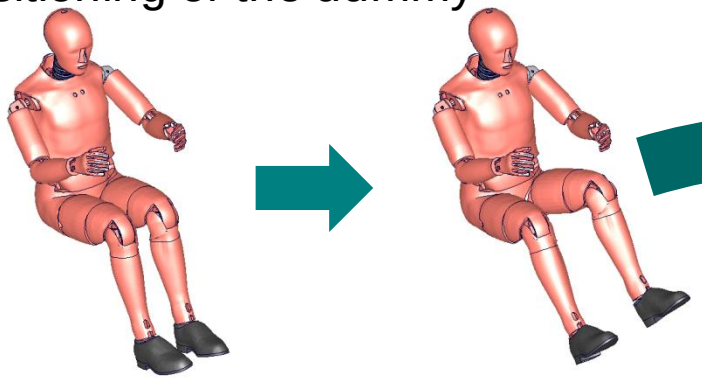
Alexander Gromer

DYNAmore Users Forum 2014

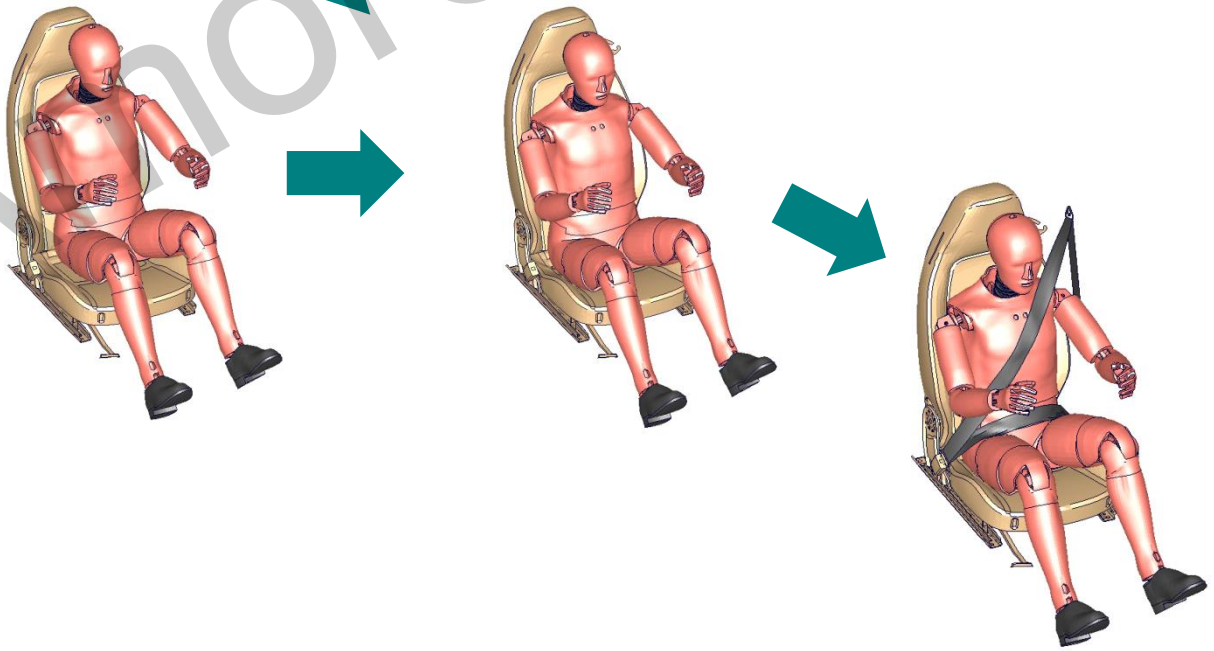
Bamberg, October 7th – 8th 2014

Incorporation of a dummy model into a car model

- Positioning of the dummy



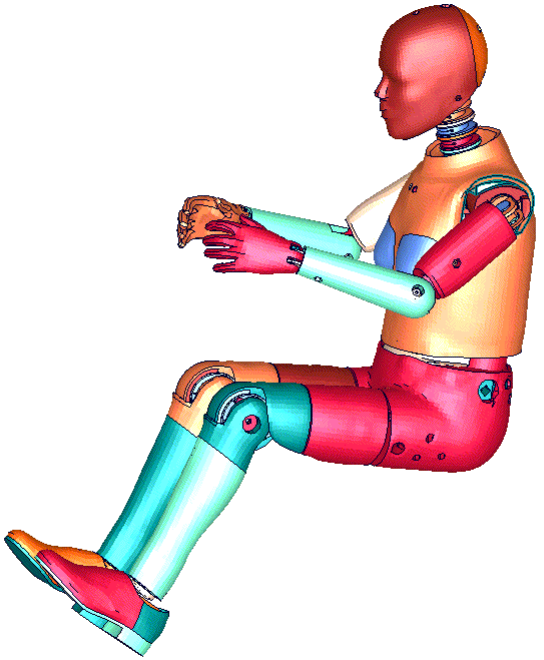
- Seat deformation



- Belt routing

DYNAmore/Humanetics: pre simulation template

- We provide a pre-simulation input where angles have to be adjusted
- Position your dummy in the preprocessor. Keep modified angles in mind.
- Set up a pre-simulation by inserting the angles into the delivered template
- Run the pre-simulation
- Extract nodes of the deformed dummy model into the original dummy input



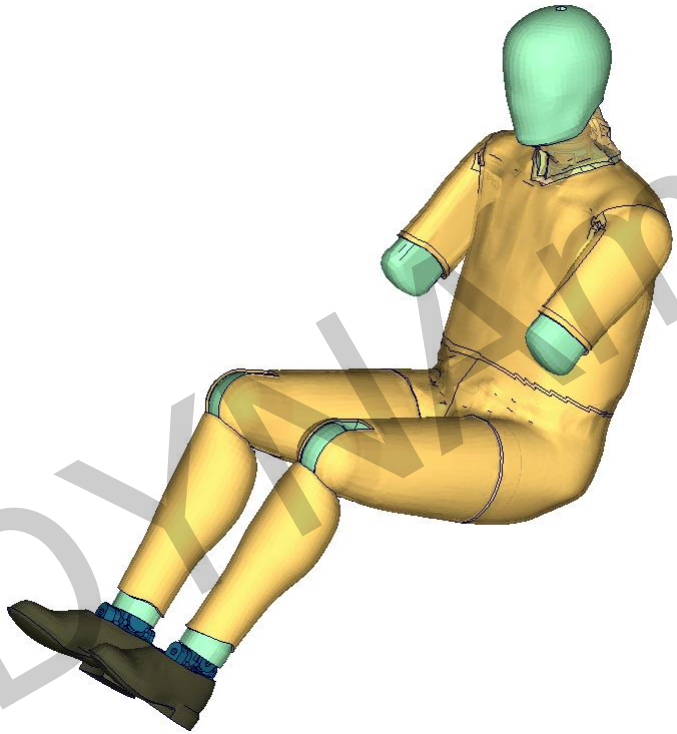
- Note: There are artificial joints for thorax bending

DYNAmore: New positioning procedure

- Main goal: simplify the positioning of the WorldSID 50th

1. Generate your target procedure in your preprocessor. Do not worry about penetrations, …. Save it as a normal key file.

origin



target

New positioning procedure

2. Run the script which is enclosed in your delivery package.

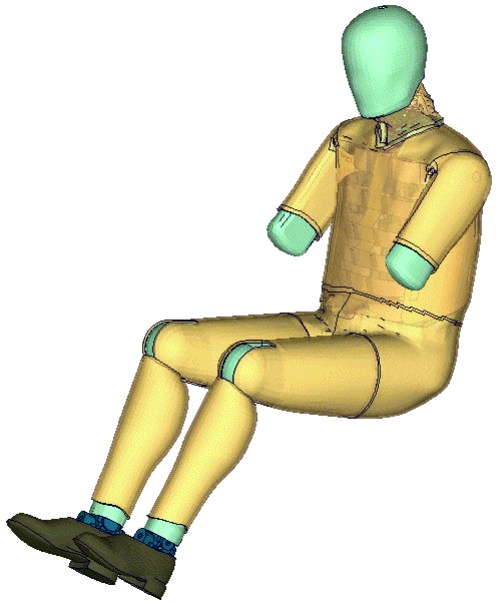
```
psg_wsid50_v3.0 -d dummyinput_origin.key -t target_pos_dummy.key
```

The script generate an positioning simulation file `dummyinput_origin_positioning.key`

3. Run the input in `dummyinput_origin_positioning.key` LS-DYNA (recommended version R6.1.2)

4. Setup the positioned dummy model with the nodal coordinates of the last state of the positioning run.

final



Very helpful: DYNA tool plot2coor

- Generate *NODE cards from d3plots
- Choosing a certain state
- Replace *NODE cards from existing keyword inputs
- `plot2coor -s last -in -k dummy.key d3plot` -> dummy.key.plot2coor
- `plot2coor -s last -in -k seat.key d3plot` -> seat.key.plot2coor

Tutorial: Dummy positioning in Primer

- PRIMER provides a function for dummy model positioning
- PRIMER generates cables which pull the dummy into the desired position

Tutorial: Dummy positioning in Primer

1. Open the dummy positioning tutorial model
2. Position the H350 to:

   H-Point (-1975.0 350.0 540.0)

   Pelvis & Thorax Angle to -23°
3. Save the origin dummy position as “origin”
4. Move legs and arms in right position
5. Save the positioned model as “target”
6. Follow the “dyna” positioning guide

7. Run the simulation

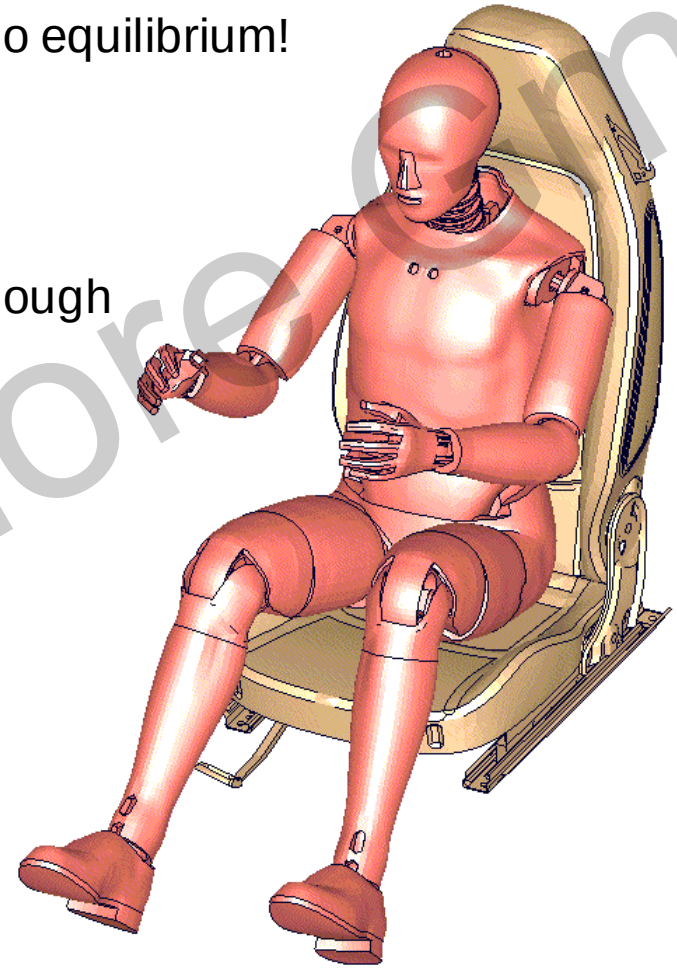
8. After the run: include the resultant nodal coordinates to your dummy model

Humanetics: new recommended procedure

- Due to material encryption PRIMER is not able to work
- Humanetics provides modified Tree files for each dummy model

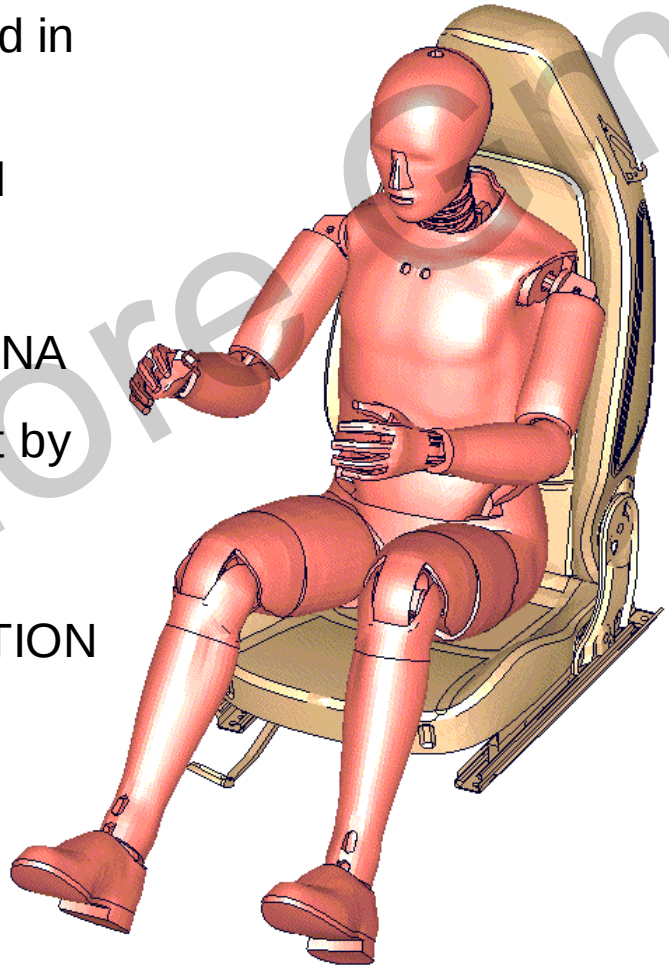
Getting the dummy into the seat by prescribed motion

- Only short simulation runs are needed (about 0.1s)
- At the end of the pre-simulation: no equilibrium!
- End-position is prescribed
- Apply slight damping
- For whiplash often not accurate enough



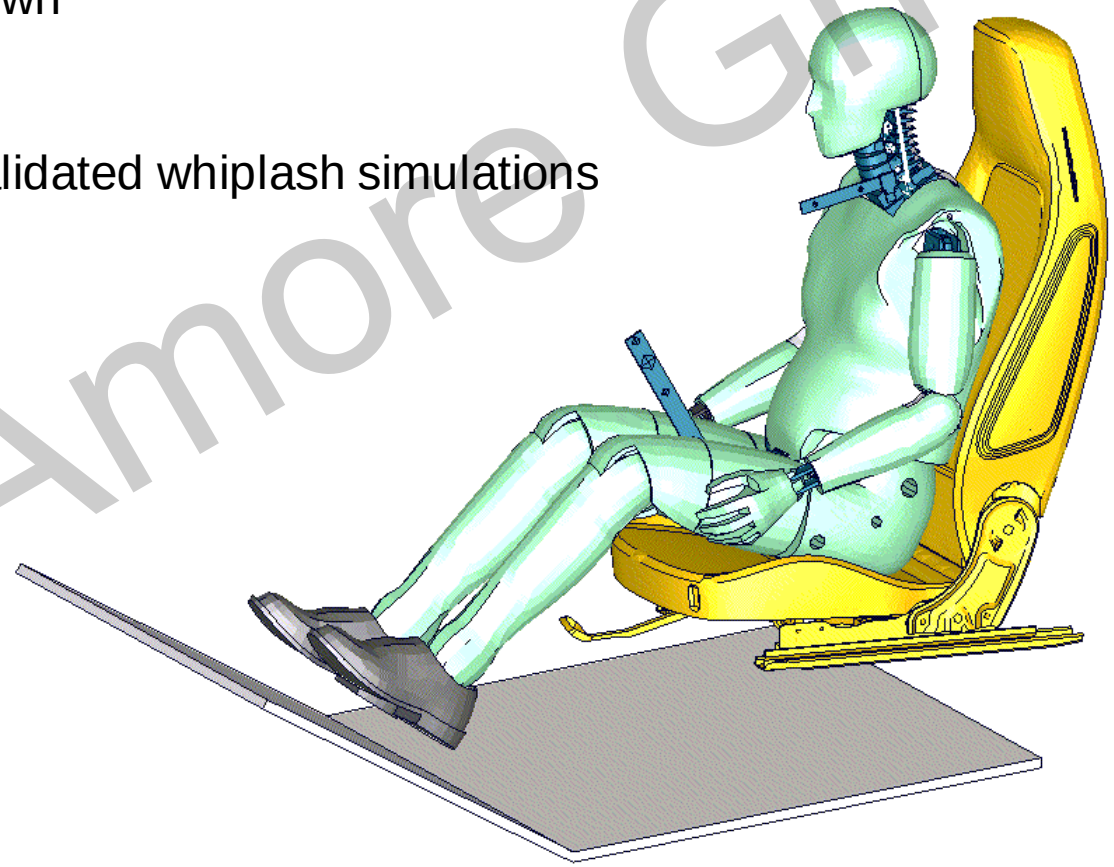
Getting the dummy into the seat by prescribed motion

- Therefore some different methods are established.
- In principal the dummy is set to rigid in any kind of way:
  - Cut out outer skin and set rigid
  - Use rigidify option of PRIMER
  - Use defomable2rigid of LS-DYNA
- Move the rigid dummy into the seat by using a LS-DYNA simulation.
- Most popular keyword is: *BOUNDARY_PRESCRIBED_MOTION
- Write out the nodal coordinates of last state and put into original seat model.



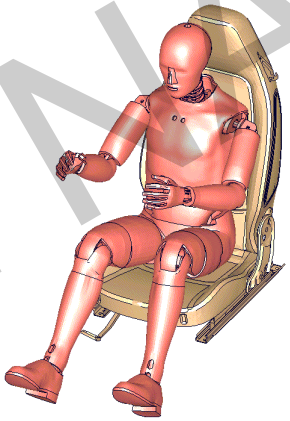
Getting the dummy into the seat under gravity load

- Long simulation runs are needed (about 1s)
- At the end of the pre-simulation: equilibrium!
- End-position is unknown
- Apply slight damping
- Is required for high validated whiplash simulations



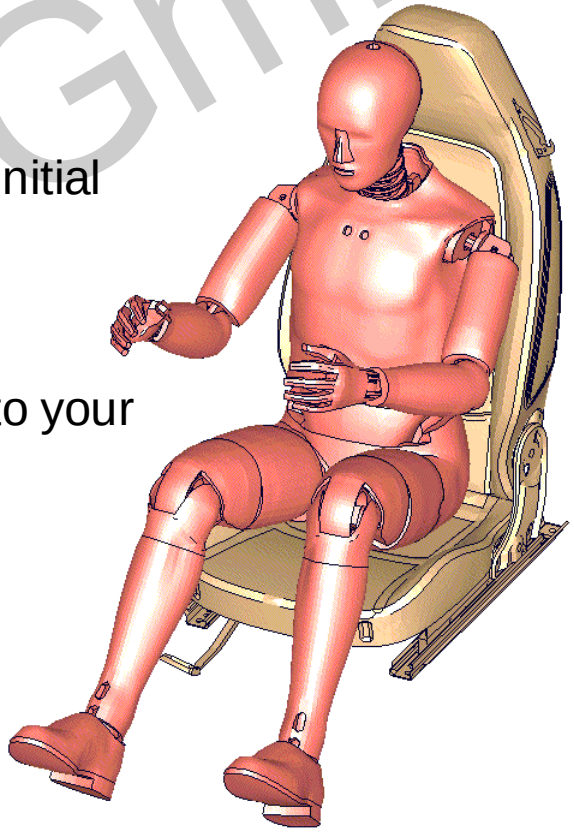
Including of pre-stressed vehicle seats

- *INITIAL_FOAM_REFERENCE_GEOMETRY. Only possible with MAT 57, 83, 77, 181, 183. Invokes initialization of deformation gradient of the element.
- This is also used for the deformable dummy parts like pelvis flesh.
- Easy to use.
- Parts can be arbitrary moved without loosing the correct initial stress
- Only available for solid elements
- For shells and beams use *INTERFACE_SPRINGBACK_LSDYNA. The stresses of the last state will be written into the dynain file in keyword format

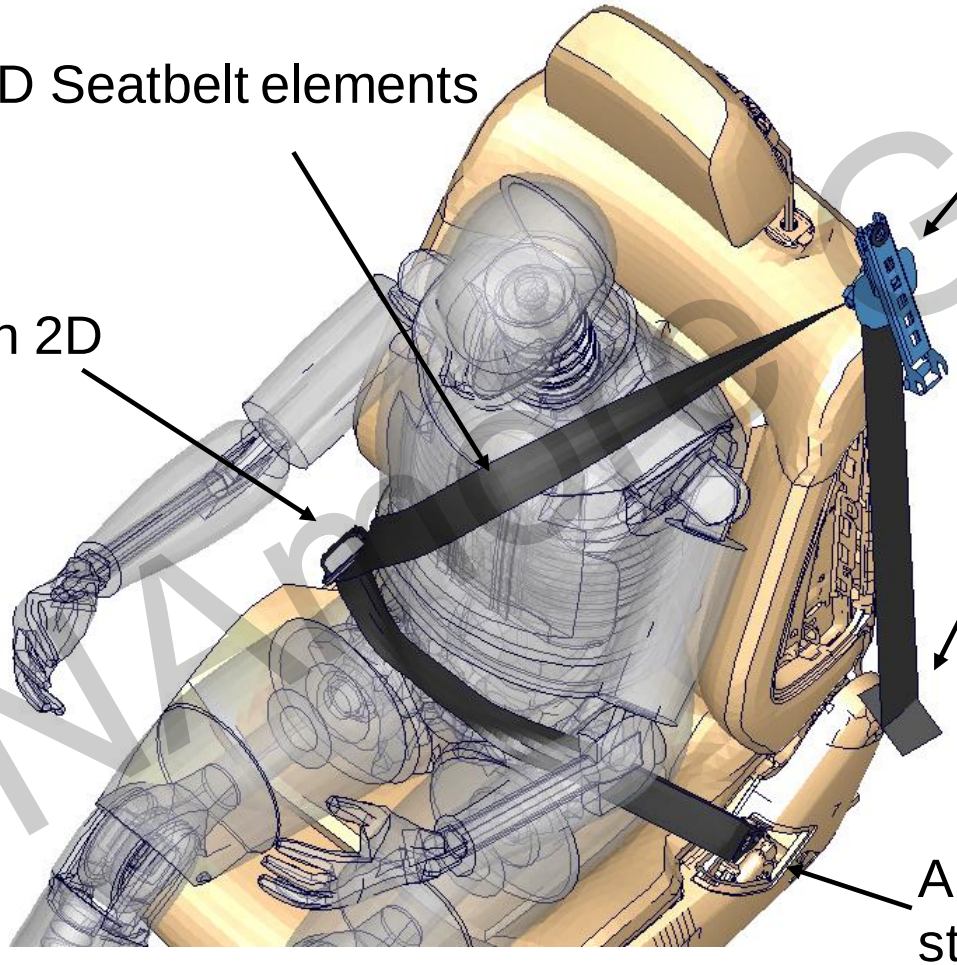


Tutorial: set up a prescribed seat deformation simulation

1. Read in your positioned dummy model in PRIMER
2. Use the rigidify feature to fix the dummy skeleton to one big rigid part
3. Write out the modified Dummy model
4. Adjust the *DEFINE_TRANSFORMATION to avoid initial penetrations
5. Run the model
6. Insert the deformed geometry and initial stresses into your origin models



Best practice example

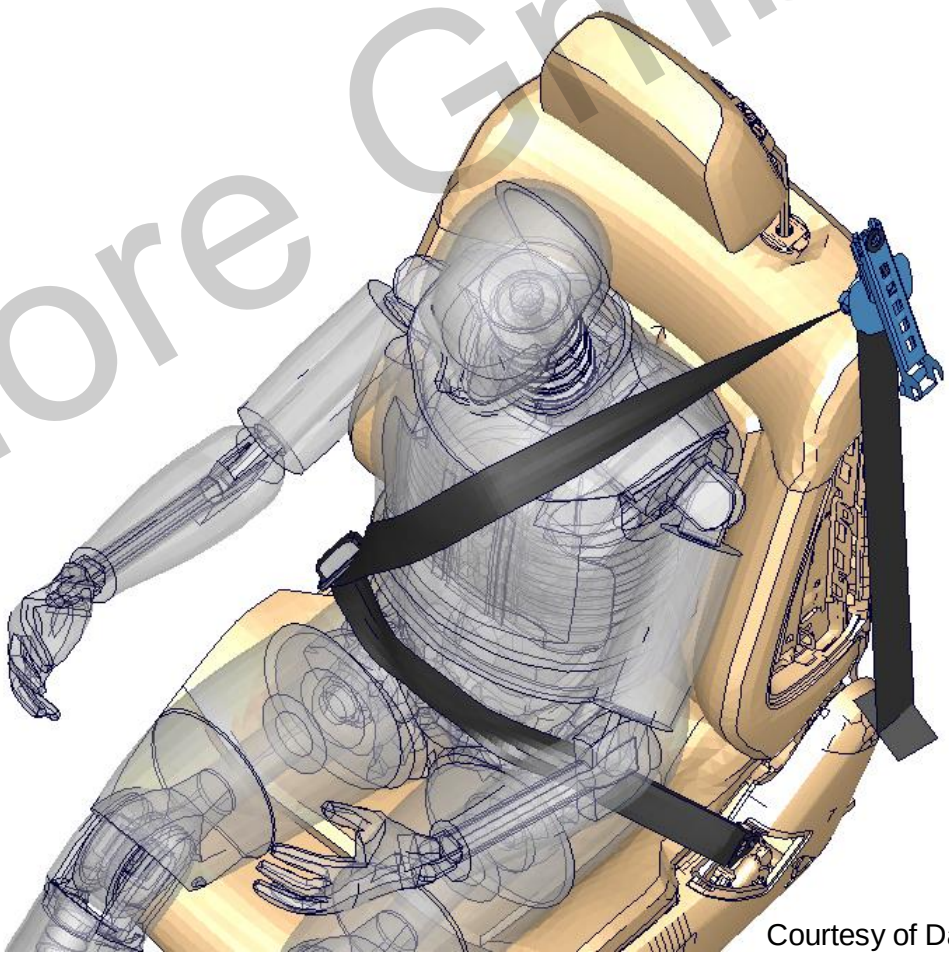


Belt should have contact to seat

Courtesy of Daimler

Tutorial: get your positioned dummy belted

1. Read in your positioned dummy and seat model in PRIMER
2. Go to the Occupants – Seatbelt routine



Courtesy of Daimler

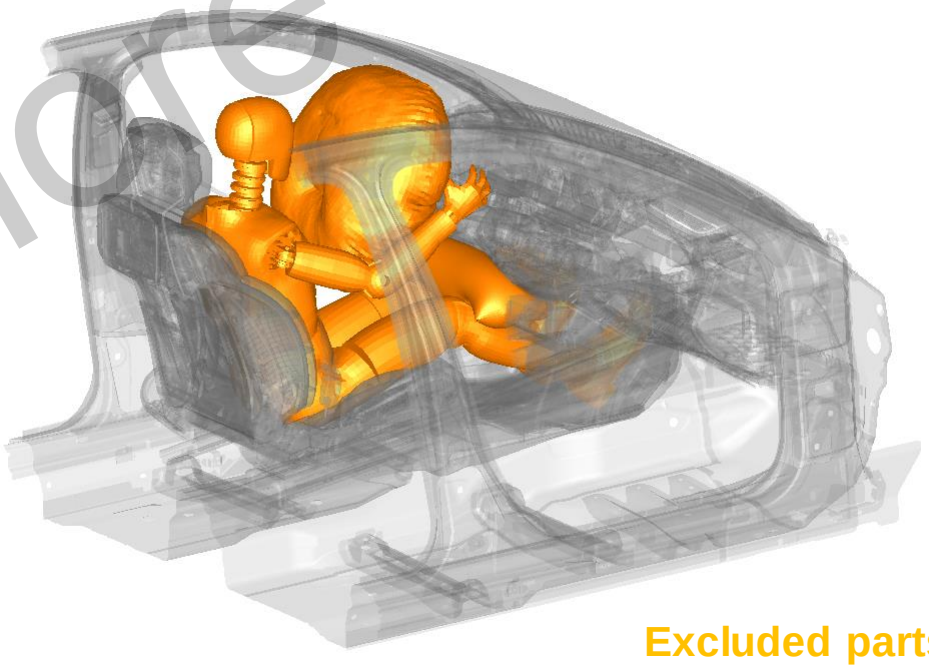
General recommendations:

- Generally use SOFT options for contact definitions
- Do not try to save elements for contact in order to reduce run time of the model

  The bucket sort algorithm will take care about that
- Work with the DTSTIFF flag to make models more independent form time step size
- Try to avoid large penetrations and x-edges
- Work with null shells coats for contact
- Do not double/triple contact definition(!!!!!!)
- For better scalability / run time performance / clarity use as less contacts as possible

No contact definitions needed:

- Airbag self contacts / airbag to housing contacts should be included in the airbag component model
- Internal dummy model contacts are included in each model
- These contact definition are part of the validation of each component and should not be touched

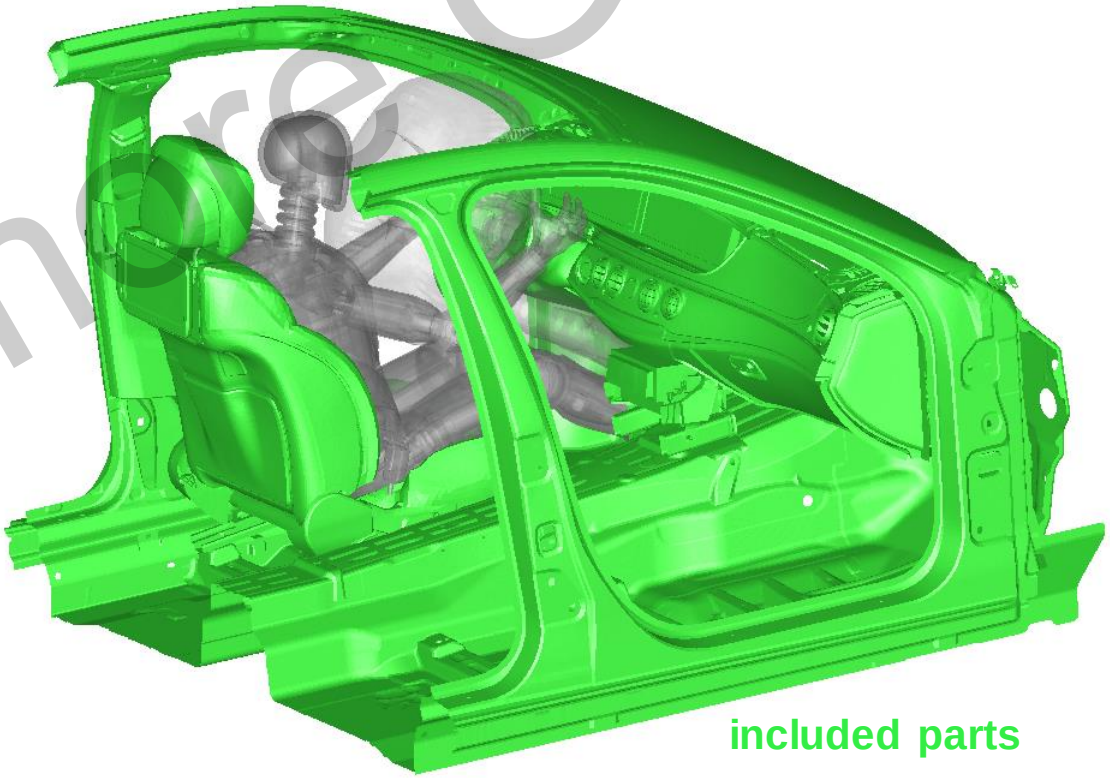


Excluded parts

Courtesy of Daimler

Global contact definition:

- The parts of any model component which should be considered for contact should be gathered in a *SET_PART
- Most elegant way: 1 *CONTACT_AUTOMATIC_SINGLE_SURFACE per occupant model



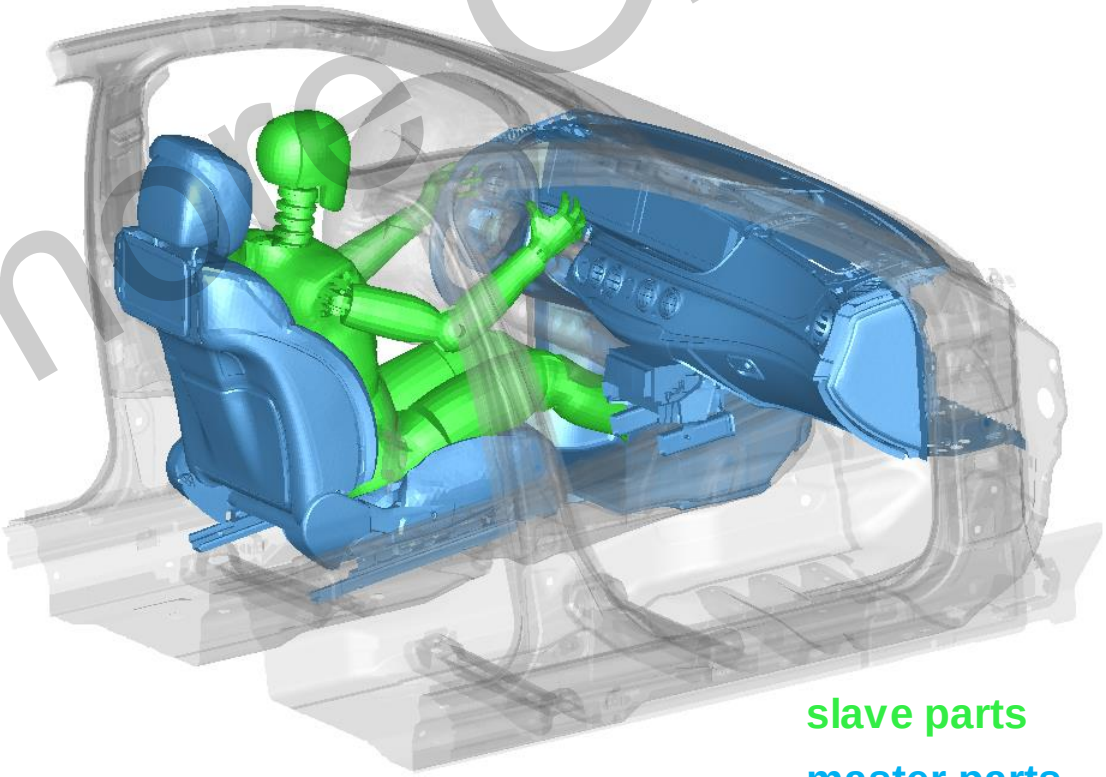
Courtesy of Daimler

Best practice: global contact definition in LS-DYNA:

```
*CONTACT_AUTOMATIC_SINGLE_SURFACE
$    SSID|     MSID|    SSTYP|    MSTYP|   SBOXID|   MBOXID|      SPR|      MPR|
         1                   2                   0         0         0         0
$      FS|       FD|       DC|       VC|      VDC|  PENCHK|       BT|       DT|
        -2        1
$     SFS|      SFM|      SST|      MST|     SFST|     SFMT|      FSF|      VSF|

$    SOFT|  SOFTSCL|   LCIDAB|   MAXPAR|   SBOPT|    DEPTH|    BSORT|   FRCFRQ|
         1
$  PENMAX|  STHKOPT|   SHLTHK|    SNLOG|     ISYM|    I2D3D|   SLDTHK|   SLDSTF|

$    IGAP|   IGNORE|   DPFRAC|   DTSTIF|                    FLANGL|  CID_RCF|
                   2                 7e-4
$   Q2TRI|   DTPCHK|    SFNBR|   FNLSCL|   DNLSCL|     TCSO|   TIEDID|   SHLEDG|

$  SHAREC|   CPARM8|   IPBACK|    SRNDE|
```

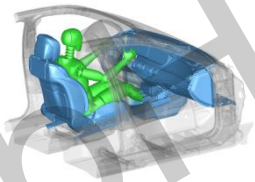
Standard Surface-to-Surface definition:

- *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE without edge checking
- Typical usage: dummy to interior
- Should work fine for 99%
- In rare cases edge checking for some components is needed



Courtesy of Daimler

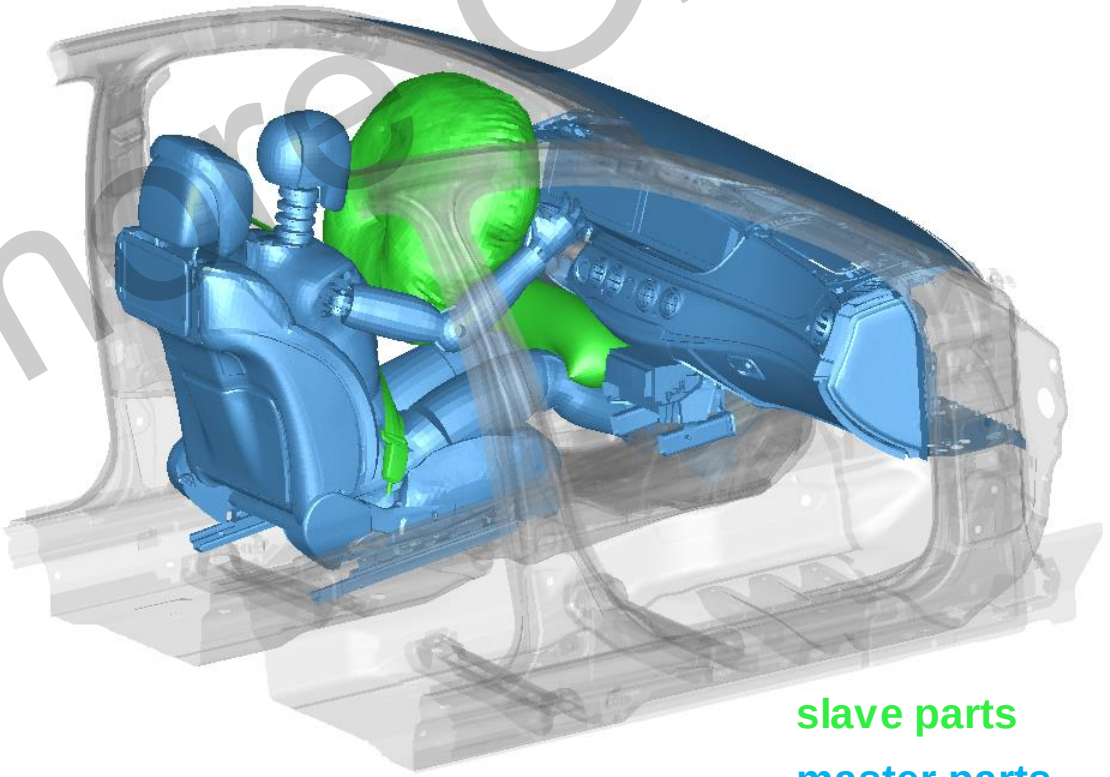
Best practice: standard surface-to-surface definition



```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE
$     SSID|     MSID|     SSTYP|     MSTYP|    SBOXID|    MBOXID|      SPR|      MPR|
         1         2          2          2          0          0         0         0
$       FS|       FD|       DC|       VC|      VDC|   PENCHK|       BT|       DT|
        -2         1
$      SFS|      SFM|      SST|      MST|     SFST|     SFMT|      FSF|      VSF|

$     SOFT|  SOFTSCL|   LCIDAB|   MAXPAR|    SBOPT|    DEPTH|    BSORT|   FRCFRQ|
         1
$   PENMAX|  STHKOPT|   SHLTHK|    SNLOG|     ISYM|    I2D3D|   SLDTHK|   SLDSTF|

$     IGAP|  IGNORE|   DPFRAC|   DTSTIF|                    FLANGL|  CID_RCF|
                 2                 7e-4
$    Q2TRI|   DTPCHK|    SFNBR|   FNLSCL|   DNLSCL|     TCSO|   TIEDID|   SHLEDG|

$   SHAREC|   CPARM8|   IPBACK|    SRNDE|
```

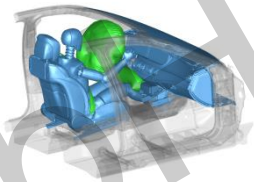
Surface-to-Surface definition with edge checking:

- *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE with SOFT=2
- Typical usage: bags and belts to interior/dummy
- Slightly higher computational costs than standard (SOFT=1) definition



Courtesy of Daimler

Best practice: surface-to-surface definition with edge checking



```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE
$     SSID|     MSID|     SSTYP|     MSTYP|    SBOXID|    MBOXID|      SPR|      MPR|
         1         2          2          2          0          0         0         0
$       FS|       FD|       DC|       VC|      VDC|   PENCHK|       BT|       DT|
        -2         1
$      SFS|      SFM|      SST|      MST|     SFST|     SFMT|      FSF|      VSF|

$     SOFT|  SOFTSCL|   LCIDAB|   MAXPAR|    SBOPT|    DEPTH|    BSORT|   FRCFRQ|
         2                                        3          5
$   PENMAX|  STHKOPT|   SHLTHK|    SNLOG|     ISYM|    I2D3D|   SLDTHK|   SLDSTF|

$     IGAP|  IGNORE|   DPFRAC|   DTSTIF|                     FLANGL|  CID_RCF|
                   2                 5e-4
$    Q2TRI|   DTPCHK|    SFNBR|   FNLSCL|   DNLSCL|     TCSO|   TIEDID|   SHLEDG|

$   SHAREC|   CPARM8|   IPBACK|    SRNDE|
```

Best practice: beam contact

- *CONTACT_AUTOMATIC_GENERAL
- High computational costs -> selective usage

```
*CONTACT_AUTOMATIC_GENERAL
$     SSID|     MSID|    SSTYP|    MSTYP|   SBOXID|   MBOXID|      SPR|      MPR|
         1                    2                   0         0         0         0
$      FS|       FD|       DC|       VC|      VDC|   PENCHK|       BT|       DT|
      0.15      0.15                           20.0
$     SFS|      SFM|      SST|      MST|     SFST|     SFMT|      FSF|      VSF|

$    SOFT|  SOFTSCL|   LCIDAB|   MAXPAR|    SBOPT|    DEPTH|    BSORT|   FRCFRQ|
         1
```

Best practice: tied contact

- *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET
- Suitable for most component to component fixations
- Robust and stable
- Penalty contact
- Can cause energy loss in glstat

```
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET
$     SSID|     MSID|    SSTYP|    MSTYP|   SBOXID|   MBOXID|      SPR|      MPR|
         1         2         2         2         0         0         0         0
$       FS|       FD|       DC|       VC|      VDC|   PENCHK|       BT|       DT|

$      SFS|      SFM|      SST|      MST|     SFST|     SFMT|      FSF|      VSF|
                         a-s-a-p   a-s-a-p
```

a-s-a-p: as small as possible

Best practice: *DEFINE_FRICTION

- Define separate friction values for each component to component contact when using a single contact interface
- A lot of contacts impede the scalability of your models!
- Invoke *DEFINE_FRICTION by setting FS=-2 and FD=DFID
- Use the *SET_PARTS of the contact interface and *PARAMETER for the friction values

```
*DEFINE_FRICTION
$     DFID|        FS|        FD|        DC|        VC|
          1       0.3       0.2         1        20
$   PID_I|    PID_J|        FS|        FD|        DC|        VC|    PTYPEI|    PTYPEJ|
     dummy      seat    &fs_d2s   &fd_d2s          1                   PSET       PSET
$   PID_I|    PID_J|        FS|        FD|        DC|        VC|    PTYPEI|    PTYPEJ|
     dummy  footwell    &fs_d2f   &fd_d2f          1                   PSET       PSET
$
```

Thank you very much for your attention!

Alexander Gromer

DYNAmore GmbH, Germany

alexander.gromer@dynamore.de