MSC.Dytran 2004

Release Guide
Chapter 7
Using MSC.Dytran

- Running MSC.Dytran, 37
- Postprocessing MSC.Dytran Results, 38
- Postprocessing MSC.Dytran Results of Windows on UNIX, 38
MSC.Dytran 2004 is the most significant and comprehensive version of MSC.Dytran released by MSC.Software. MSC.Dytran is a general-purpose, three-dimensional computer program for simulating the dynamic response of solids, structures and fluids. It combines structural and fluid mechanics technology to facilitate modeling, and uses explicit time integration to provide an efficient solution.

Many major enhancements have been incorporated into this release. A short description of each enhancements is given here, while a more detailed description of certain enhancements is available in later sections.

Chapter 1: Introduction

Chapter 2: MSC.Dytran LS-DYNA option

MSC.Dytran 2004 now includes the most advanced structural DMP solution, based on LS-DYNA. This capability allows the user to optionally execute LS-DYNA from inside MSC.Dytran. It is possible to run models on a cluster of CPUs in Distributed Memory Parallel (DMP) mode. The MSC.Dytran LS-DYNA 2004 release supports structural models only.

Chapter 3: Airbags (OOP) & General FSI

- Euler Element Resize Option
  This capability allows the user to re-define the element size of an adaptive Euler mesh at any time during the simulation.

- Multiple Adaptive Euler Domains for Type 9 and the Roe Solver
  Multiple Euler domains are automatically generated around a coupling surface, and each Euler domain automatically adapts itself when the coupling surface moves and deforms. The coupling surface can represent an airbag, an airbag compartment, a blast-resistant bunker, etc. This capability is available for the single material, hydrodynamic Euler solvers (Type 9 and Roe). Material in- and out-flow can be defined, as well as flow between the Euler domains, across porous or open areas in the coupling surfaces.

- Pol-pack based Porosity
  For simulations with multiple coupling surfaces a new method of computing transport between the Euler domains has been implemented. This method (the pol-pack method) is more accurate than the method that was already available in Dytran 2002 (the facet method). The pol-pack method remains accurate for large facets in the coupling surface that describe the flow-through area.

- BPFULL contact for folded airbags
  This contact logic correctly treats deployment of airbag models, where contact occurs between layers later during the simulation time. This is especially important for cases where objects are very close to the deploying airbag, and the contact between the layers needs to remain in tact under a large internal pressure.

  This new contact algorithm has been tested against 10 industrial airbag models, and has been verified during pre-release testing by several customers, receiving a very favorable response.

- Ignition Growth (IG) Equation of State for Type 9 and Type 11 Euler Solvers – One Material Only
  The Ignition Growth equation of state has been implemented in the Euler solvers, with the limitation that only one material can be used. It can work in combination with COUPLE and void. The advantage of the IG model over the conventional JWL equation of state is given by the fact that details of the onset and growth of an explosive detonation wave can
be accurately calculated. This includes aspects whether or not a compression wave will build up to a detonation wave, and whether such a detonation wave can extinguish at a later point in time. In conventional JWL modeling, assumptions have to be made about the detonation to actually occur and proceed with a constant detonation speed.

IG calculations are significant in modeling and understanding detailed material behavior in defense industry applications. Furthermore, the uncertainty on detonation build-up is important in calculations of 'sympathetic' detonations, i.e. safety issues where unintended impact loads on potentially explosive materials may actually develop detonation characteristics.

The IG material model has been verified for eight types of different explosive materials under a variety of experimental conditions reported in literature.

Chapter 4: Robustness, Speed, Ease of Use, Accuracy

- Many of the enhancements in this category were implemented in support of crash dummy modeling (Chapter 5) and to make several customer applications run robustly, with less CPU time, with less modeling effort, without the need for manual interventions, and with high accuracy.
- Springs with Hysteresis
  A spring with hysteresis can now be modeled by specifying a different loading and unloading curve for the CSPR and CELASx elements. This option allows accurate modeling of the chest deflection in a crash dummy.
- Multiple Acceleration Fields Applied to Parts of the Model
  Crash analysis typically requires two acceleration fields (crash-pulse, gravity) and the crash pulse is only applied to parts of the model (the dummy). The BODYFOR capability has been updated to allow multiple definitions for sets of gridpoints.
- Accelerometer output
  Output of acceleration, velocity, and displacement (x,y,z component) of a GRID relative to a moving coordinate frame can be requested on the new case control option ACC.
- CG definition of a MATRIG in a local coordinate system
  For positioning of a crash dummy, it is important that the CG of MATRIGs can be defined in a local coordinate system, that is attached to the MATRIG. The coordinate system is defined by gridpoints and positioning of the dummy automatically redefines the location and orientation of the coordinate system.
• Remove limit on number of contacts
  The limitation on the number of contacts (127) has been removed. There is no limit.

• Remove limit on number of gridpoint boundary conditions
  The limitation on the number of gridpoint boundary conditions (2047) has been removed. There is no limit.

• New internal numbering scheme for elements
  With previous versions, an array the length of the highest element number was used during input processing. This is no longer done.

• 2x2x2 Hexa Element
  A new hexa8 element based on a full integration scheme has been implemented. This element is useful in cases where the reduced integration scheme would lead to severe hourglassing.

• New Tet Element
  A new 4-noded tet element has been implemented. It is based on the standard linear finite element formulation (in contradiction with the previous one based on collapsed linear hexahedron). This new element is more robust and accurate and twice as fast as the previous element.

• New FOAM2 material model
  A new FOAM2 material model has been implemented that can accurately and robustly simulate a S-form material data, even when the material has large stiffness at the end of the compaction. Even when the slope of the stress-strain curve is greater than the given young modulus, the simulation remains stable. This is achieved by using the current state of the material to determine the time step. (In the previous FOAM2 model, the young modulus was used, which lead to an over-estimation of the time-step and instabilities.)

  This new capability is important to simulate the rebound phase of a foam-packaged computer during a drop-test.

• Fast Hughes-Liu Beam Elements
  The dynamic data storage and retrieval logic has been re-designed for the Hughes-Liu beam elements. This resulted in a 5-10 times speedup.

• This new capability is important to simulate the New HL beam cross-section
  New cross section of beam (hollow rectangular with round corners) has been implemented for Hughes-Liu beam. This option allows easy to use, accurate modeling of beams in applications like helicopter skid gear behavior under crash landing.
Chapter 1: Introduction

- **FLEXLM License check from Dytran-Explorer; Windows only**
  A feature to test the availability of a correct FLEXLM environment has been added to Dytran Explorer. This will assist in diagnosing licensing problems.

- **Cleanup of Installation**
  Certain obsolete tools have been removed, and no more links are generated in the user’s home-directory upon installation of MSC.Dytran. The following obsolete programs are removed from the installation directory: xdextr, xdytran, and mpeg_player.

- **Fixed Qualities**
  Forty-six qualities have been fixed. Among them:
  - Fixed mismatch between step summary energy and material summary energy.
  - The real PV curve of FOAM2 material has always slope that is greater than the $k$ for tension. Sometimes the time step stability is violated because it's based on $K$.
  - Fixed TOTAL-E in the printout.
  - Fixed a bug in the calculation of CG for the model when rigid element (MATRIGS - with only mass given) are put together with deformable elements.
  - Fixed a problem with the pressure load applied to TET4 element faces. The pressure load was applied in the wrong direction.
  - Fixed an error message given by LinElas (DMAT) for solids when created by the dytran preference. The error message was not needed by using a proper default.

Chapter 5: Crash Dummies for OOP

A validated Finite Element model of a 5% Hybrid III female crash dummy is now available. It allows the user to perform simulations without the need for a multibody linkage program like ATB or Madymo.

Chapter 6: System Information

This chapter gives you the Windows, UNIX, and LINUX software installation information. It also gives you memory requirements for all three platforms.

Chapter 7: Using MSC.Dytran

This chapter tells you how to use MSC.Dytran on all three platforms (Windows, UNIX, and LINUX). It also tells you how to postprocess results with MSC.Patran 2004 and how to postprocess MSC.Dytran Windows results on UNIX.
MSC.Dytran 2004 includes the most advanced structural DMP solution, based on LS-DYNA. This capability allows the user to optionally execute LS-DYNA from inside MSC.Dytran. The LS-DYNA option can be run on one processor or on a cluster of processors in DMP mode. The MSC.Dytran LS-DYNA 2004 release supports structural models only. No modification are needed to a regular MSC.Dytran input file to submit it with the LS-DYNA option on 1 or multiple CPUs.

On Windows, MSC.Dytran Explorer can be used to set up the environment to run in DMP mode. In order to run the DMP job with MSC.Dytran LS-DYNA option, first you need to make sure that “MPICH” is installed on all the machines that are part of the DMP cluster. MPICH is a public domain, portable implementation of MPI, the Standard for message-passing libraries. MPICH is made available on the MSC.Dytran
2004 installation CD. Please make sure that MPICH is installed on all machines that are clustered for the DMP job submission. Next, follow the steps below to set up your environment:

1. Click on the MSC.Dytran Explorer and select the input filename. Choose “DMP” as the Dytran “Executables” (Figure 2-1).

![MSC.Dytran Explorer Menu](image1)

**Figure 2-1** MSC.Dytran Explorer Menu

2. Click on the run/play button, ➤, to pop up the MSC.Dytran LS-DYNA window to configure your environment (Figure 2-1).

3. A. You can add new machines and their corresponding maximum processors by typing the machine name and number of processors in the blank field and clicking on the “Add Machine” button. You can add or delete as many machines as available in your cluster. The cluster you define is saved in the registry and is available for future job submissions (Figure 2-2).

B. Before submitting the job, you can make a sub-selection of machines to use for the simulation. A machine is selected by clicking the check box in front of its name. (Figure 2-2).
4. Once the machines are selected, you need to set up the “System Environmental Settings”. There are two options (Figure 2-2):

a. **No System shares needed** – In order to use this option, MSC.Dytran 2004 has to be installed on all the machines.

b. **Use System Shares** – This option allows all the machines to “share” the executable and the run directory on the user-defined machines. MSC.Dytran 2004 has to be installed only on the machine that has a “shared” executable:
   - The shared directory can be specified in the “Shared Execdir” (Figure 2-2).
   - The job input and output files can be shared and stored on the directory as listed in the “Shared Rundir” (Figure 2-2).

The location to a shared directory typically looks like: `\machinename\sharename`.

To use this option, it is required that the shared directories have proper access privileges.
5. To start the job, click on the \( \rightarrow \) in the MSC.Dytran LS-DYNA window. The first time you submit a job, you will be asked by MPICH to enter an account (Figure 2-3):

1. (domain\user):
2. (password):

This account will be used to access all the machines selected. The account information will be temporarily stored for the duration of your current login session. You are able to re-enter this information only after a log-off and re-login.

![Command Window](image)

**Figure 2-3** Command Window

After the simulation is finished, you can postprocess the binary result file (d3plot), using the DRA option in the MSC.Dytran preference of MSC.Patran 2004. A conversion tool of the d3plot file to regular MSC.Dytran ARC files is available by right-clicking on the file inside MSC.Dytran-Explorer.

A typical application for MSC.Dytran LS-DYNA is automotive crash analyses (Figure 2-4).
This model was run on a cluster of Windows machines; each one 1.7Ghz Pentium 4 CPU. The scaling is shown in Table 2-1.

**Table 2-1** Scaling of Automotive Crash Analyses Shown in Figure 2-4

<table>
<thead>
<tr>
<th>Number of CPUs 1.7 Ghz p4</th>
<th>MSC.Dytran LS-DYNA 2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 hrs. 52 min. 2 sec.</td>
</tr>
<tr>
<td>2</td>
<td>5 hrs. 35 min. 37 secs.</td>
</tr>
<tr>
<td>4</td>
<td>3 hrs. 3 min. 29 secs.</td>
</tr>
<tr>
<td>7</td>
<td>2 hrs. 22 min. 58 secs.</td>
</tr>
</tbody>
</table>
Chapter 3: Airbags (OOP) and General FSI

- Euler Element Resize Option
- Multiple Adaptive Euler Domains for Type 9 and the Roe Solver
- Pol-pack Based Porosity
- BPFULL Contact for Folded Airbags
- Ignition Growth (IG) Equation of State for Type 9 and Type 11 Euler Solvers (One Material Only)
Euler Element Resize Option

This capability allows the user to re-define the element size of an adaptive euler mesh at any time during the simulation.

This is done by referring to a TABLE from the MESH entry, to scale the element size up or down. The element size can be modified independently in the x-, y-, and z-directions:

```
MESH, 1, ADAPT, 0.01, 0.02, 0.03, +
+, +, +,
+, EULER, 200, +
+SCALE, 300
```

```
Pauler1,200,...
```

```
TABLED1,300,,,,,  ==> Time vs. Element Size Scale factor
```

Using this option allows optimization of accuracy and CPU time requirements. For example, during the initial stages of an airbag deployment, a fine Euler mesh is needed. Once the airbag has deployed partly, a coarser Euler mesh is sufficient.

During the time-step for which a resize is specified, the program will create a completely new Euler mesh, maps the solution of the old Euler mesh onto the new Euler mesh, and removes the old Euler mesh. After this mapping step, the solution will continue.

Multiple Adaptive Euler Domains for Type 9 and the Roe Solver

Multiple Euler domains are automatically generated around a coupling surface, and each Euler domain automatically adapts itself when the coupling surface moves and deforms. The coupling surface can represent an airbag, an airbag compartment, a blast-resistant bunker, etc. This capability is available for the single material, hydrodynamic Euler solvers (Type 9 and Roe). Material in- and out-flow can be defined, as well as flow between the Euler domains, across porous or open areas in the coupling surfaces.

An example of a side-curtain airbag is shown here, with courtesy of Autoliv (Figure 3-1) as well as an bunker blast example, with courtesy of the Berkeley University (Figure 3-2). All images in these Figures are created with CEI/Ensight.
Chapter 3: Airbags (OOP) and General FSI

MSC.Dytran 2004 Release Guide
Multiple Adaptive Euler Domains for Type 9 and the Roe Solver

Figure 3-1  Side-curtain Airbag

Figure 3-2  Bunker Blast Example
Pol-pack Based Porosity

For simulations with multiple coupling surfaces a new method of computing transport between the Euler domains has been implemented. This method (pol-pack) is more accurate than the method that was already available in MSC.Dytran 2002 (the facet method). The pol-pack method remains accurate for large facets in the coupling surface that describe the flow-through area.

The algorithm used is depicted schematically in Figure 3-3. The two Euler domains overlap, at which location a “Hole” Euler Mesh is automatically generated. The areas of two coupling surfaces that coincide within this “Hole” Euler Mesh can be designated as being porous, and allow gas to flow from one Euler domain into the other. The detailed theory is described in Ref 2. [“Simulation of a compartmented airbag deployment using an explicit, coupled Euler/Lagrange method with adaptive Euler domains”]

This new method calculating the flow through holes and open areas makes the applications listed in “Multiple Adaptive Euler Domains for Type 9 and the Roe Solver” above run more accurate and robust. It can also be applied to simulations of baffles in fuel tanks and storage containers (Figure 3-4)

By default this new pol-pack method is used. To activate the old facet method add PARAM, FLOW-METHOD, FACET to the input deck.
BPFULL Contact for Folded Airbags

This contact logic correctly treats deployment of airbag models where contact occurs between layers later during the simulation time. This is especially important for cases where objects are very close to the deploying airbag, and the contact between the layers needs to remain in tact under a large internal pressure.

This new contact algorithm has been tested against 10 industrial airbag models and has been verified during pre-release testing by several customers, receiving a very favorable response. Due to the sensitive nature of these models, no example images can be presented here.

This new algorithm is activated by specifying \texttt{SEARCH=BPFULL} on the \texttt{CONTACT} bulk data entry. It combines the functionality of the original search methods \texttt{FULL} and \texttt{BPLANE}. This new algorithm should not be used for all applications, since it has been written for airbag deployment simulations.

During the analysis, this contact algorithm might control the time-step to avoid missing contacts.

Ignition Growth (IG) Equation of State for Type 9 and Type 11 Euler Solvers (One Material Only)

The Ignition Growth equation of state has been implemented in the Euler solvers, with the limitation that only one material can be used. It can work in combination with \texttt{COUPLE} and \texttt{void}. The advantage of the IG model over the conventional JWL equation of state is given by the fact that details of the onset and growth of an explosive detonation wave can be accurately calculated. This includes aspects whether or not a compression wave will build up to a detonation wave, and whether such a detonation wave can extinguish at a later point in time. In conventional JWL modeling, assumptions have to be made about the detonation to actually occur and proceed with a constant detonation speed.

IG calculations are significant in modeling and understanding detailed material behavior in defense industry applications. Furthermore, the uncertainty on detonation build-up is important in calculations of ‘sympathetic’ detonations; i.e., safety issues where unintended impact loads on potentially explosive materials may actually develop detonation characteristics. The IG material model has been verified for eight types of different explosive materials under a variety of experimental conditions reported in literature (Figure 3-5).
Figure 3-5  IG Material Model
In this chapter, some details are given on the QA metrics of MSC.Dytran 2004, and how the enhancements affect the robustness, speed, ease of use, and accuracy. Please see Chapters 1-3 for a description of all the individual enhancements.
Robustness

The robustness of this release is quantified by a cross-platform quality metric. The QA for MSC.Dytran 2004 contains a set of 518 input decks. The binary result files (ARC and THS) are compared to a reference result, which is identical for each platform. By using the same reference result for each platform, a consistent cross-platform behavior is ensured. The quality metric for this release was reached at:

99+% success rate for every QA model (518) on every release machine (10)

The few models that did not pass the QA comparison ran properly till the end-time, but small differences in results occurred. These differences in results were within bounds and explainable by effects of different numerical round-offs when running the models cross-platform.

Speed

With this release a Distributed Memory Parallel (DMP) option becomes available for structural models. Please see Chapter 2 for more details on the MSC.Dytran LS-DYNA capability.

When not using the MSC.Dytran LS-DYNA option, virtually the same CPU time is used as with MSC.Dytran 2002 r2. We have tested this on all release platforms. There are two notable exceptions, where the CPU time was decreased significantly:

a. Hughes-Liu Beams

MSC.Dytran 2004 processes these elements 5-10 times faster. This was verified on a customer model for a helicopter skid-gear landing. Also see “Accuracy”.

b. Tet Elements

MSC.Dytran 2004 processes the new tet elements two times faster. This was verified on customer models for wheel-impact and computer drop-test. The computer drop-test is described in more detail in “Accuracy”.

Also, when not using the MSC.Dytran LS-DYNA option, it is possible to run models in Shared Memory Parallel (SMP) on multi-CPU hardware. The percentage of functionality that runs in SMP mode has increased significantly. Special attention was given to the scalability when using more than two CPUs. A much better scalability has been achieved up to four CPUs (Figure 4-1). We did not test the scalability characteristics on more than four CPUs. The SMP capability is now supported and was tested on all release platforms.
For the Fluid Structure Interaction (FSI) simulations, we have added new capabilities that can dramatically improve the CPU time with respect to previous releases. For instance, by using the multiple adaptive Euler domains with element resizing, a typical airbag model, with full CFD, can be simulated in 5-10 hours on a dual CPU desktop machine, compared to 30-40 hours using the previous release.

**Ease of Use**

The Dytran preference of MSC.Patran 2004 supports all the new functionality released with MSC.Dytran 2004.

With MSC.Patran 2004, the Direct Result Access (DRA) method is available for post-processing the results stored in the ARC files. The DRA method dramatically improves the speed of accessing the data and plotting/animating the results.

The effort for modeling Fluid Structure Interaction problems can be decreased significantly by using the multiple adaptive Euler domains. It completely eliminates the task of creating an Euler mesh, since it is generated fully automatic. There is no danger anymore that the structure in which the gas of fluid resides moves out of the Euler domain. Whenever the structure, modeled as a coupling surface, moves near the boundary of the adaptive Euler domain, additional Euler elements will be added. By removing Euler elements in regions where they are no longer functional, the memory requirements remain low.
Ease of use is also improved by the removal of several limitations. Most important is the limit on the number of contacts, which sometimes necessitated the creation and maintenance of more than one model to capture the proper load-paths through a structure. For a complete list of removed limitations, see Chapter 1.

Accuracy

Several enhancements were made that dramatically increased the accuracy of the solution.

1. Hughes-Liu Beams

The result of MSC.Dytran 2004 compares very well when applied to a customer example of a helicopter skid-gear landing simulation (Figure 4-2)

![Level Landing (12 ft/sec)](image)

**Figure 4-2** Helicopter Skid-gear Landing Simulation
2. Pol-pack based porosity
   See Pol-pack Based Porosity in Chapter 3.

3. BPFULL contact type
   See BPFULL Contact for Folded Airbags in Chapter 3.

4. Ignition Growth Equation of State
   See Ignition Growth (IG) Equation of State for Type 9 and Type 11 Euler Solvers (One Material Only) in Chapter 3.

5. FOAM2 material; 2X2 Hexa; New Tet Element
   See Chapter 1 for a description of the updates.

The combination of these three updates was successfully applied to a customer drop test model of a computer in its packaging (Figure 4-3). The simulation did not run till completion with MSC.Dytran 2002 r2, because the model became unstable during the rebound phase. This was resolved by updating the FOAM2 material model. Furthermore, because foam material is a relatively soft material, it is very prone to hourglassing, which disturbs the solution and makes it unstable. The new fully integrated Hexa element avoids these problems. Meshing with Hexa elements is sometimes difficult. In those cases, it is possible to use the new Tet element which gives a more accurate solution than the previous Tet element. In this example, the Tet element provided an acceptable solution for design purposes, since the predicted accelerations of the computer were very close to the experimental values.

Figure 4-3  Customer Drop Test
As part of the Example Problem Manual with MSC.Dytran 2004, a Finite Element model for the Hybrid III 5th%-tile small female dummy is introduced. This dummy has been especially calibrated and validated for Out-Of-Position (OOP) situations as described in regulation FMVSS 208, revision 2001 and beyond, set forward by the National Highway Traffic Safety Administration (NHTSA).

The dummy model comprises 28 rigid body segments (Figures 5-1 and 5-3), linked together with multi-axis joints. Each body segment has predefined inertia properties, and surface contour. The inertia properties are defined through the MATRIG option, while rigid shell elements define the surface contour. Between the individual body segments, “null” elements may be defined to obtain a closed surface for smooth contact behavior. The kinematic joints, connecting the body segments, are modeled with directional springs (CELAS1) and dampers (CDAMP1).
Figure 5-1  Small Female Dummy

The dummy model is offered with two neck models (Figure 5-2). A traditional 4-pivot neck model is offered that has proven robustness even at the highest off-design loading conditions, albeit with reduced accuracy. A full finite-element rubber neck model is offered for increased accuracy and versatility:

1. **Pivot Neck Model**: This model uses a series of rigid bodies connected with directional springs and dampers. There are solid elements between the rigid bodies, but they are only modeled to capture the inertia effects during the simulation. The stiffness is set to be very low.

2. **Rubber Neck Model**: The rigid parts of the neck are modeled with solid rubber elements. Contact is defined between the slits to avoid penetration during flexion of the neck. The spring and damper elements are completely deactivated when this model is chosen. The rubber elements have validated material properties.
Tetra elements are used to model the slits to improve the quality and of the robustness of the behavior of the neck during flexion. The rubber is modeled with a linear viscoelastic material model.

![Neck Models](image)

Figure 5-2  Neck Models

The structure of linked rigid bodies is generally known as a tree structure (Figure 5-3). Starting at the Lower Torso, which is at the root of the structure, the tree branches out to the Legs, and Arms and Head, through the Upper Torso. Several coordinate systems are associated with each rigid body to define its properties and the linkage with adjoining bodies. Every rigid body has a coordinate system that defines the body center of mass and the three principal inertia axes. Each body, except the Lower Torso which is at the root of the body structure, has a coordinate system which defines the location and orientation of the joint that connects it to its preceding, or Parent body. This coordinate system is referred to as the Joint Child Coordinate System. Body segments that have a connection to an outer branch of the tree structure, further have a coordinate system that locates the origin of that branch and its orientation relative to the parenting body. This coordinate system is referred to as the Joint Parent Coordinate System. Note that each body may have several children, but has only one parent. The relative position and orientation of the Joint Parent and Child coordinate systems determines the forces and moments that act between them. Depending on the body, two more types of coordinate systems may have been defined. For the Lower Torso, Upper Torso, and Head, a coordinate system is defined in the location and with the orientation of the accelerometers. Joint Parent coordinate systems in Femurs, Tibias and Neck double as coordinate frames for the load transducers that are present at these locations.
To help visualize the various coordinate systems, rod elements have been defined in their locations (Figure 5-4).
The structure of the MSC.Dytran input deck allows that the dummy model data is split into two files. The file, hyb305_case.mod, contains the CASE CONTROL data, comprising options for output of the dummy kinematics (into ARC files) and instrumentation responses (into THS files). The file, hyb305_bulk.mod, comprises the bulk data options defining the dummy model. The model defined in this file represents the dummy in so-called “reference position” (Figure 5-1). For a typical simulation analysis, the dummy is used in a position and with an orientation that differs from this reference position. A utility program is provided to change the position and orientation of the dummy as a whole and of the limbs individually, to accommodate this (Figure 5-5). The user provides the desired location of the H-point, orientation of the Lower Torso, and rotation angles of arm and leg segments in a position definition file. The utility program, hyb305.exe, then assembles a dummy model in the correct position and orientation from the hyb305_bulk.mod file and the user-provided position definition file. The resulting bulk data file can be included in a simulation model for analysis. Positioning or changing the orientation of dummy parts through other means may seriously affect the integrity of the dummy model and is discouraged.
Chapter 5: Crash Dummies for OOP

Reference Database:

Model Files:
- hyb305_case.mod <= Case Control
- hyb305_bulk.mod <= BULK DATA

Definition File:
hyb305.def

(-) H-Point Positioning
(-) Rotation around Joint
(-) Numbering Base (Renumbering)
(-) Selection of Neck Model
(Pivot or Rubber FEM)

Input File:
hyb305_case.dat <= Case Control

Create Input File From Reference Database:

% hyb305.exe

reading definition file: hyb305.def
reading model file     : hyb305_case.mod
reading model file     : hyb305_bulk.mod
Rotating around LeftKneeAngle     : -90.00000
Rotating around RightKneeAngle    : -90.00000
Rotating around LeftShoulderAngle : -90.00000      0.0000000E+00
Rotating around RightShoulderAngle: -90.00000      0.0000000E+00
writing input file     : hyb305_case.dat
writing input file     : hyb305_bulk.dat

-------------------------------------------------------------------------------
Summary of H-point and the CG locations of the parts
-------------------------------------------------------------------------------

<table>
<thead>
<tr>
<th>name</th>
<th>GRID</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. H-Point</td>
<td>1010000</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>1. Lower Torso</td>
<td>1010001</td>
<td>0.1000000</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>2. Lumbar Spine</td>
<td>1020001</td>
<td>-8.0809998E-04</td>
<td>0.0000000E+00</td>
<td>8.0715999E-02</td>
</tr>
<tr>
<td>3. Abdomen</td>
<td>1030001</td>
<td>-9.6593000E-02</td>
<td>0.0000000E+00</td>
<td>6.4117998E-02</td>
</tr>
<tr>
<td>4. Upper Torso</td>
<td>1040001</td>
<td>-8.0809998E-04</td>
<td>0.0000000E+00</td>
<td>0.1727160</td>
</tr>
<tr>
<td>5. Sternum</td>
<td>1050001</td>
<td>3.4299999E-02</td>
<td>0.0000000E+00</td>
<td>0.3310000</td>
</tr>
<tr>
<td>6. Neck Bracket</td>
<td>1060001</td>
<td>4.4814002E-02</td>
<td>0.0000000E+00</td>
<td>0.3865710</td>
</tr>
<tr>
<td>7. Neck</td>
<td>---- build from several disks ----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Head</td>
<td>1080001</td>
<td>8.9213997E-02</td>
<td>0.0000000E+00</td>
<td>0.5036000</td>
</tr>
<tr>
<td>9. Left Femur</td>
<td>1090001</td>
<td>0.0000000E+00</td>
<td>0.1779000</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>10. Right Femur</td>
<td>1100001</td>
<td>0.0000000E+00</td>
<td>2.2100000E-02</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>11. Left Knee</td>
<td>1110001</td>
<td>0.1442000</td>
<td>7.9609998E-02</td>
<td>-2.5010001E-02</td>
</tr>
<tr>
<td>12. Right Knee</td>
<td>1120001</td>
<td>0.1442000</td>
<td>-7.9609998E-02</td>
<td>-2.5010001E-02</td>
</tr>
<tr>
<td>13. Left Shin</td>
<td>1130001</td>
<td>0.3365000</td>
<td>0.1796100</td>
<td>-2.9850001E-02</td>
</tr>
<tr>
<td>14. Right Shin</td>
<td>1140001</td>
<td>0.3365000</td>
<td>2.0390000E-02</td>
<td>-2.9850001E-02</td>
</tr>
<tr>
<td>15. Left Tibia</td>
<td>1150001</td>
<td>0.4305000</td>
<td>7.9609998E-02</td>
<td>8.8149972E-02</td>
</tr>
<tr>
<td>16. Right Tibia</td>
<td>1160001</td>
<td>0.4305000</td>
<td>-7.9609998E-02</td>
<td>8.8149972E-02</td>
</tr>
<tr>
<td>17. Left Ankle</td>
<td>1170001</td>
<td>0.5945000</td>
<td>7.9609998E-02</td>
<td>6.6149928E-02</td>
</tr>
<tr>
<td>18. Right Ankle</td>
<td>1180001</td>
<td>0.5945000</td>
<td>-7.9609998E-02</td>
<td>6.6149928E-02</td>
</tr>
<tr>
<td>19. Left Foot</td>
<td>1190001</td>
<td>0.5651700</td>
<td>7.9609998E-02</td>
<td>-5.9087079E-02</td>
</tr>
<tr>
<td>20. Right Foot</td>
<td>1200001</td>
<td>0.5651700</td>
<td>-7.9609998E-02</td>
<td>-5.9087079E-02</td>
</tr>
<tr>
<td>21. Left Shoulder</td>
<td>1210001</td>
<td>5.4519001E-02</td>
<td>2.3000000E-02</td>
<td>0.3720840</td>
</tr>
<tr>
<td>22. Right Shoulder</td>
<td>1220001</td>
<td>5.4519001E-02</td>
<td>-2.3000000E-02</td>
<td>0.3720840</td>
</tr>
<tr>
<td>23. Left Upper Arm</td>
<td>1230001</td>
<td>-5.4520000E-02</td>
<td>0.2622000</td>
<td>0.3823000</td>
</tr>
<tr>
<td>24. Right Upper Arm</td>
<td>1240001</td>
<td>-5.4520000E-02</td>
<td>-0.2622000</td>
<td>0.3823000</td>
</tr>
<tr>
<td>25. Left Lower Arm</td>
<td>1250001</td>
<td>6.2679991E-02</td>
<td>0.1622000</td>
<td>0.3823000</td>
</tr>
<tr>
<td>26. Right Lower Arm</td>
<td>1260001</td>
<td>6.2679991E-02</td>
<td>-0.1622000</td>
<td>0.3823000</td>
</tr>
<tr>
<td>27. Left Hand</td>
<td>1270001</td>
<td>0.2752800</td>
<td>0.1622000</td>
<td>0.3822999</td>
</tr>
<tr>
<td>28. Right Hand</td>
<td>1280001</td>
<td>0.2752800</td>
<td>-0.1622000</td>
<td>0.3822999</td>
</tr>
</tbody>
</table>

-------------------------------------------------------------------------------

Figure 5-5 Dummy Positions
The calibration simulations performed focus on the dummy behavior during an OOP airbag impact. To this end only in the area of the neck and chest calibration tests have been performed. The head impact and knee impact are skipped at this point in time. The simulation performed follows the guidelines of Ref 1. [Technical Report - Development and Evaluation of the Hybrid III fifth Percentile Female Crash Test Dummy (H-III5F), NHTSA, August 1998]

For a complete description of these calibration tests, please refer to the Example Problem Manual. Here, we will describe the Neck Extension Calibration Test and the Thorax Pendulum Impact Test only.

**Neck Extension Calibration Test**

**Experiment**
In this test the neck and head of the dummy are attached to a pendulum. The structure is then lifted to a certain high and released. The pendulum will impact a foam material which will give it a certain deceleration profile. Due to the deceleration the head of the dummy will start show extension. In this test the moment and the head D-plane rotation are measured as a function of time.

**Simulation**
The neck portion of the dummy was extracted from the main hybrid305_bulk.dat file and was attached to the pendulum by merging the MATRIG definitions using the parameter MATRMERG. In order to save CPU time, the simulation will start just before the pendulum contacts the foam material. The contact with the foam is not modeled, but instead the pendulum is given a predefined rotational velocity profile. The rest of all the neck and head nodes are given an initial rotational and translational velocity by using the TIC3 input option. At the moment of impact, the radial velocity of the system is 3.5523 Rad/sec. The force in rotational spring 1080051 will be monitored. This resembles the neck moment. For the D-Plane, the location of nodes 1080020 and 1080023 are monitored. These nodes are part of the coordinate system that describes the location and orientation of the head accelerometer. From the relative location of these two nodes the head D-Plane rotation can be calculated.

**Results**
The results for the pivot neck model and the rubber neck model for every 20 mseconds followed by the time history graphics of the neck moment and head D-plane are given below (Figure 5-6).
Figure 5-6  Pivot Neck and Rubber Neck Results
Figure 5-6  Pivot Neck and Rubber Neck Results (continued)
Thorax Impact

Experiment
In this calibration test an impactor with a weight of 14 kg will contact the chest at a velocity of 6.47 m/s.

Simulation
First the dummy was positioned using the hyb305 positioner. The arms and legs were straightened. Than the floor and impactor were added to the test input file. A contact was defined between the impactor and the dummy chest. During the simulation, the force in the spring 1050031 was monitored. This is the impact force. Also the location of nodes 1040100 and 1050000 was monitored. The difference in x-position is a measure for the chest deflection.

Results
Below the results for the thorax impact are given for every 10 mseconds followed by the graphs for thorax displacement in time and thorax force deflection curve (Figure 5-7).
Figure 5-7  Thorax Impact
Figure 5-7  Thorax Impact (continued)
Software Installation

On the Windows platforms, MSC.Dytran 2004 easily installs from CD-ROM as it uses the standard Windows 2000 Installation Wizard. On Unix and Linux platforms, you use the MSC.Software standard installation script to install the software on your system. MSC.Dytran 2004 is the successor of MSC.Dytran 2002 r2.

MSC.Dytran uses the FLEXlm license manager as the licensing system for nodelock and network licensing. To run MSC.Dytran, you need an authorization code from MSC.Software Corporation. A new license is required for MSC.Dytran 2004, even if you already have a license for MSC.Dytran 2002r2.
On Windows and Linux computers, MSC.Dytran requires your computer to have an Ethernet card, even if your computer is not connected to a network. The FLEXlm licensing mechanism uses the Ethernet card to create the unique system identification encrypted in the license information file.

MSC.Dytran 2004 was built and tested on the following hardware with the listed software installed as given in Table 1.

### Table 1: Supported Hardware Configuration

<table>
<thead>
<tr>
<th>Platform</th>
<th>Operating System</th>
<th>Compiler Version</th>
<th>OpenMP Parallel Support</th>
<th>LS-DYNA &amp; LS-DYNA-DMP</th>
<th>USA</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium III</td>
<td>Windows NT 4.0-SP6A</td>
<td>Compaq Fortran 6.6B SMP: Intel 7.1(*)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>Intel Pentium III</td>
<td>Windows 2000-SP4</td>
<td>Compaq Fortran 6.6B SMP: Intel 7.1 (*)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>Intel Pentium III</td>
<td>Windows XP-SP1</td>
<td>Compaq Fortran 6.6B SMP: Intel 7.1(*)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>SGI (**) R10K/R12K</td>
<td>IRIX64 6.5.7m</td>
<td>MIPSpro 7.3.1.3m</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
<tr>
<td>HP-UX – PA RISC 2.0 (**)</td>
<td>HPUX 11.0</td>
<td>HP F90 V2.7</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
<tr>
<td>HP-UX Itanium2</td>
<td>HPUX B.11.22</td>
<td>HP F90 V2.7</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
<tr>
<td>Compaq Alpha</td>
<td>Tru64 Unix v5.0</td>
<td>DF 90 V5.5-2602</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
<tr>
<td>Sun Sparc Solaris</td>
<td>Solaris 5.8</td>
<td>Sun Work Shop 6u2 (FORTTRAN 95 6.2)</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
<tr>
<td>IBM RS/6000</td>
<td>AIX 4.3.3.0</td>
<td>XL Fortran 8.1</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
<tr>
<td>Intel Linux</td>
<td>Red Hat 7.3</td>
<td>Intel 7.1</td>
<td>Yes</td>
<td>No (***)</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
</tbody>
</table>

(*): For correct operation of the Intel Fortran compiler, MS DevStudio 6.6 or Compaq Visual Fortran 6.6B must be installed prior to installing the Intel compiler.

(**): With Dytran 2004, the following configurations are no longer supported:
- SGI: R4k and R5k
- HP: HP-UX 10.20
- SUN: Solaris 7

With Dytran 2004, the following configurations are supported by above builds:
- Windows: Intel Pentium IV (NT,W2k,XP)(tested)
- SGI: R8k (not tested)
- SUN: Solaris 5.9 (tested)
- IBM: power3, Power4 (tested)
- Intel Linux: MSC.Linux (tested)

In most cases MSC.Dytran 2004 will run on higher OS levels. It has been found that MSC.Dytran 2004 will not run on following configurations:
- Intel Linux: Redhat 9.0

(***) The LS-DYNA & LS-DYNA-DMP options on Unix and Linux will become available during Q2 of 2004.

## Memory Requirements

In general, the size of the memory required by MSC.Dytran depends on the size of the engineering problem you wish to solve. The default memory size is set to approximately 30 MB. This default size is appropriate for smaller sized problems.
You can change the preset default in the MSC.Dytran Explorer so that it fits your personal needs. In addition, you can define the minimum and maximum memory size and use the slider in the front panel to select the desired memory size. On Unix and Linux platforms you can use the command-line option (size="small/medium/large") or you can enter the MEMORY-SIZE definition in the input file.

MSC.Dytran traces the usage of memory and prints a summary at the end of the output file of each analysis. The memory size listed in the summary is exact. It reflects the memory required for storing the model in core memory after one integration step. Additional memory required during the analysis is automatically allocated and de-allocated.

When you change the memory setting for an analysis through the MSC.Dytran Explorer, the settings are stored to be used the next time that you run the analysis.

Under certain conditions, MSC.Dytran may stop and issue a message that it cannot allocate the required memory. Since the memory allocation in MSC.Dytran is dynamic, the system may require additional memory during an analysis. If the memory is available, it will be allocated and de-allocated when it is no longer needed. When your computer runs out of memory, the MSC.Dytran analysis may stop when it needs more memory to continue. You may solve this problem by closing applications on your computer that you do not need, or you can decrease the size of the core memory that MSC.Dytran allocates for the analysis if you are using substantially more than the analysis requires. You can find the information on the memory size requirements of the analysis in the memory summary at the end of the analysis. We recommend MSC.Dytran be used on a computer that has at least 256 MB of RAM.
Running MSC.Dytran

On Windows, submit an MSC.Dytran analysis by double-clicking the MSC.Dytran icon. The icon should be available on your desktop. Alternatively, you can use the “Start Menu” to locate MSC.Dytran under the “Programs Folder”. Once you picked either the icon, or the menu entry, the MSC.Dytran user environment appears on your screen.

The MSC.Dytran Explorer provides an on-line help system that contains information about the functionality of the MSC.Dytran Explorer. The MSC.Dytran Explorer provides some basic post-processing and animation tools.
On Unix and Linux platform you would use the command line interface like `dytran jid=xxx` or submit from MSC.Patran.

**Postprocessing MSC.Dytran Results**

MSC.Dytran results can be postprocessed with MSC.Patran. With MSC.Patran 2004, the Direct Result Access (DRA) method has been implemented for both regular Dytran output files as well as for the d3plot file which is generated by the LS-DYNA option.

In addition, on Windows, you can use the VisualVrml postprocessing and animation functionality and the Visual Time History functionality built into the MSC.Dytran Explorer. When using the LS-DYNA option, a conversion tool of the d3plot file to regular Dytran ARC files is available by right-clicking on the file inside Dytran-Explorer.

**Postprocessing MSC.Dytran Results of Windows on UNIX**

If you wish, you can postprocess the Windows analysis results on a UNIX computer. In this case, you need to convert the binary result files (`.ARC` and/or `.THS`) files to a UNIX format. You can perform this conversion by using the right-mouse button menu in the MSC.Dytran Explorer. Point your mouse at the file that you wish to convert, click the right mouse button, and select the `Convert to binary...` menu item. The converted files will have the `sb_` prefix. For Digital Alpha workstations, the native Windows result files can be used directly without conversion.

Alternatively, when running on Windows, you can select the option to output result files in UNIX format by default. To set this option, select the “Preferences” from the “Options” menu. Choose “Formats” and select “Convert output files automatically to UNIX-format”. If you select this option, the regular Windows result files and the converted UNIX-format files are written at the end of the analysis. You can recognize the UNIX-format files by the `ux` prefix.